Unsupervised Learning of Multiword Expressions

IOANNIS KORKONTZELOS

Ph.D. Thesis
This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

THE UNIVERSITY of York

Artificial Intelligence Group
Department of Computer Science
United Kingdom

20 September 2010

To my grandfather and aunts

Abstract

Multiword expressions are expressions consisting of two or more words that correspond to some conventional way of saying things (Manning & Schutze 1999). Due to the idiomatic nature of many of them and their high frequency of occurence in all sorts of text, they cause problems in many Natural Language Processing (NLP) applications and are frequently responsible for their shortcomings. Efficiently recognising multiword expressions and deciding the degree of their idiomaticity would be useful to all applications that require some degree of semantic processing, such as question-answering, summarisation, parsing, language modelling and language generation. In this thesis we investigate the issues of recognising multiword expressions, domain-specific or not, and of deciding whether they are idiomatic. Moreover, we inspect the extent to which multiword expressions can contribute to a basic NLP task such as shallow parsing and ways that the basic property of multiword expressions, idiomaticity, can be employed to define a novel task for Compositional Distributional Semantics (CDS). The results show that it is possible to recognise multiword expressions and decide their compositionality in an unsupervised manner, based on cooccurrence statistics and distributional semantics. Further, multiword expressions are beneficial for other fundamental applications of Natural Language Processing either by direct integration or as an evaluation tool.

In particular, *termhood-based* methods, which are based on nestedness information, are shown to outperform *unithood-based* methods, which measure the strength of association among the constituents of a multi-word candidate term. A simple heuristic was proved to perform better than more sophisticated methods. A new graph-based algorithm employing sense induction is proposed to address multiword expression compositionality and is shown to perform better than a standard vector space model. Its parameters were estimated by an unsupervised scheme based on graph connectivity. Multiword expressions are shown to contribute to shallow parsing. Moreover, they are used to define a new evaluation task for distributional semantic composition models.

Contents

List of Tables . . . 11
List of Figures . . . 13
Acknowledgement . . . 14
Declaration . . . 15

1 Introduction and Motivation **17**
1.1 Multiword Expressions . . . 17
1.2 Motivation . . . 21
1.3 Research Subjects and Analysis . . . 22
1.4 Research Objectives and Hypothesis . . . 25
1.5 Thesis Organisation . . . 26

2 Background and Field Review **28**
2.1 Introduction . . . 28
2.2 Survey on Multiword Expression Recognition . . . 30
2.2.1 Linguistic approaches . . . 32
2.2.1.1 Tokenisation . . . 32
2.2.1.2 Part of speech tagging . . . 32
2.2.1.3 Lemmatisation . . . 33
2.2.1.4 Parsing . . . 34
2.2.1.5 Part of speech patterns . . . 36
2.2.1.6 Context . . . 37
2.2.1.7 Gazetteers . . . 38
2.2.1.8 LEXTER (Bourigault 1992) . . . 39
2.2.1.9 Morpheme bootstrapping . . . 39
2.2.2 Statistical approaches . . . 40
2.2.2.1 Unithood-based approaches . . . 40
2.2.2.1.1 Occurrence and cooccurrence frequency counts 41
2.2.2.1.2 Mean and Variance . . . 41

2.2.2.1.3 Hypothesis Testing 42
2.2.2.1.3.1 The t test 43
2.2.2.1.3.2 The t test for differences 44
2.2.2.1.3.3 Pearson's chi-square test 45
2.2.2.1.3.4 Log-likelihood ratios test 46
2.2.2.1.4 Pointwise Mutual Information 49
2.2.2.1.5 Other Unithood-based statistical measures . 51
2.2.2.2 Termhood-based approaches 52
2.2.2.2.1 C-value 52
2.2.2.2.2 NC-value 56
2.2.2.2.3 Statistical barrier 58
2.2.2.2.4 Method of Shimohata et al. (1997) 60
2.2.3 Hybrid approaches 62
2.2.3.1 SNC-value 64
2.2.3.2 Combining extractors with Latent Semantic Analysis 65
2.2.3.3 Combining extractors with Adaptive Boosting 65
2.2.3.4 Canonical Forms 66
2.2.3.5 Distinguishing Subtypes of NVCs (Fazly & Stevenson 2007) 67
2.2.3.6 A Measure of Syntactic Flexibility of NVCs (Bannard 2007) 69
2.2.3.7 Semantics-based NVC Extraction (Van de Cruys & Moirón 2007) 70
2.2.3.8 Identifying NVCs in Token Context (Cook et al. 2007) 71
2.2.3.9 An NVC Database 73
2.2.4 Discussion 73
2.3 Survey on Distributional Similarity 75
2.3.1 Representing Context 76
2.3.2 Distributional Hypothesis 81
2.3.3 Measuring distributional similarity 82
2.3.4 Discussion 86
2.4 Survey on Multiword Expression Compositionality 88
2.4.1 Lin (1999) 90
2.4.2 Schone & Jurafsky (2001) 91
2.4.3 Bannard et al. (2003) 92
2.4.4 Baldwin et al. (2003) 93
2.4.5 McCarthy et al. (2003) 93
2.4.6 Venkatapathy & Joshi (2005) 94
2.4.7 Katz & Giesbrecht (2006) 95
2.4.8 Piao et al. (2006) 95
2.4.9 McCarthy et al. (2007) 96
2.4.10 Villavicencio et al. (2007) 96
2.4.11 Discussion 97

2.5 Survey on Distributional Semantics Composition 98
2.5.1 Modelling the Context of Word Sequences 99
2.5.2 A generic compositional distributional semantic model 102
2.5.2.1 Mitchell and Lapata Model 102
2.5.2.2 Erk and Pado Model 104
2.5.2.3 BEAGLE 107
2.5.3 Discussion 109
2.6 Summary 110

3 Analysing Automatic Term Recognition Methods 112
3.1 Introduction 113
3.2 Evaluation framework overview 116
3.3 Linguistic filters 117
3.4 Statistical Automatic Term Recognition Approaches 118
3.5 Evaluation 119
3.5.1 Experimental setting 119
3.5.2 Results 125
3.5.3 Further Experiments and Results 129
3.6 Summary 132

4 Resolving Compositionality 134
4.1 Introduction 135
4.2 Sense induction for resolving compositionality 137
4.2.1 Corpora collection and preprocessing 139
4.2.2 Sense Induction 140
4.2.2.1 Locating semantically important nouns 142
4.2.2.2 Graph creation 143
4.2.2.3 Graph clustering 144
4.2.3 Comparison of major induced senses 145
4.2.4 Determining compositionality 146
4.3 Test set of multiword expressions 147
4.4 Evaluation setting 149
4.5 Evaluation results 150
4.6 Unsupervised parameter tuning 151
4.6.1 Average Degree 152
4.6.2 Average Weighted Degree 153
4.6.3 Average Cluster Coefficient 154
4.6.4 Average Weighted Cluster Coefficient 155
4.6.5 Graph Entropy 155
4.6.6 Weighted Graph Entropy 156
4.6.7 Edge Density 156
4.6.8 Weighted Edge Density 157
4.7 Evaluation results of unsupervised parameter tuning 157
4.8 Evaluation on a larger dataset 160

4.9 Further evaluation of unsupervised parameter tuning 164
4.10 Summary 165

5 Multiword Expressions and Parsing 167
5.1 Introduction 168
5.2 Annotating multiword expressions 171
5.3 Evaluation 172
5.3.1 Shallow parsing change classes 175
5.3.1.1 Change class *P2LMw* 177
5.3.1.2 Change class *P2L* 179
5.3.1.3 Change class *L2PMw* 180
5.3.1.4 Change class *L2P* 181
5.3.1.5 Change class *PL2P* 182
5.3.1.6 Change class *P2PL* 183
5.3.1.7 Change class *PN* 183
5.3.1.8 Change class *PoS* 184
5.3.1.9 Change class *P2P* 185
5.3.1.10 Change class *MwA* 186
5.3.2 Shallow parsing complex change classes 188
5.4 Target multiword expressions and corpora collection 188
5.5 Experimental results and discussion 191
5.6 Chapter Summary 196

6 Distributional Semantics Composition 199
6.1 Introduction 200
6.2 Estimating Additive CDS Models from Data 204
6.2.1 Setting the linear equation system 206
6.2.2 Computing Moore-Penrose pseudo-inverse 208
6.3 Building positive and negative examples 209
6.3.1 Dataset containing multiword expressions 211
6.3.2 Dataset containing single words 212
6.4 Experiments 214
6.4.1 Experimental setting 217
6.4.2 Results on the dataset of multiword expressions 219
6.4.3 Results on the dataset of single words 222
6.5 Summary 225

7 Conclusion and Future Work 228
7.1 Thesis Summary 228
7.1.1 Literature Summary 228
7.1.2 Research Summary 230
7.2 Contributions 233
7.3 Future Work 235
7.3.1 Multiword expression Recognition 235

7.3.2 Compositionality analysis 236
7.3.3 Multiword expressions and shallow parsing 236
7.3.4 Compositional distributional semantics models 237

References **238**

Index **254**

Citation Index **254**

List of Tables

2.1 *GENIA* tagger accuracy. 33
2.2 Words that occur significantly more often with *powerful* (the first five words) and *strong* (the last five words) in the corpus used in (Manning & Schutze 1999). 45
2.3 Observed values table (*OT*). Bigram: "gene expression" 48
2.4 Expected values table (*ET*). Bigram: "gene expression" 48
2.5 Hypothesised models for trigrams 49
2.6 Various other unithood measures 50
2.7 Various other unithood measures. 52
2.8 Various association coefficients 53
2.9 Various association coefficients. 54
2.10 C-value example 55
2.11 Context words of the term candidates in table 2.10. The capital letters within square brackets denote which candidate term the preceding context word occurs with and the frequency of occurrence is within parenthesis. 57
2.12 Numbers of distinct words that precede ($R(N)$) or follow ($S(N)$) the tokens of the candidate term *basal cell carcinoma*. 60
2.13 P: Patterns for recognising canonical forms. sg stands for singular number, pl for plural, v for verb, n for nouns, pas for passive and *det* for determiner. 66
2.14 Categorisation of NVCs in Fazly & Stevenson (2007). 67
2.15 Idioms and canonical forms example. 72
2.16 Various distributional similarity measures 87
2.17 Example frequency vectors of the components of s = "close interaction" 104

3.1 *GENIA* and *PennBioIE* corpus statistics 120

3.2 Gold-standard (*GS*) term counts and candidate term counts per linguistic filter and term length in the *GENIA* corpus. 122
3.3 Gold-standard (*GS*) term counts and candidate term counts per linguistic filter and term length in the *PennBioIE* corpus. 122
3.4 Recall (*R*) and precision (*P*) percentages (%) per linguistic filter and length of candidate term in the *GENIA* corpus. 123
3.5 Recall (*R*) and precision (*P*) percentages (%) per linguistic filter and length of candidate term in the *PennBioIE* corpus. 123
3.6 Executed experiments 124

4.1 Test multiword expressions with compositionality annotation and information about whether their compositionality was successfully detected by the *1c1word* baseline (*B*), manual parameter selection (*M*), and *average cluster coefficient* (*A*). 148
4.2 Chosen parameter values. 149
4.3 Computations of graph connectivity measures and relevant quantities on the example graph of figure 4.4 153
4.4 Test multiword expressions with compositionality annotation 161

5.1 Cross validation datasets assessing the validity of both hypotheses together. 176
5.2 Summary of the basic change classes. ✓ or ✗ denote change classes that count positively or negatively towards improving shallow parsing. *?* denotes classes that are treated specially. 177
5.3 60 compositional multiword expressions randomly chosen from *WordNet*; 37 Noun - Noun sequences and 23 Adjective Noun sequences. The size of the respective corpus in sentences appears within parentheses. . . 189
5.4 56 non-compositional multiword expressions randomly chosen from *WordNet*; 26 Noun - Noun sequences and 30 Adjective Noun sequences. The size of the respective corpus in sentences appears within parentheses. . . 190
5.5 Summary of experimental results. “PoS” stands for parts of speech, “N N” for noun noun sequences and “J N” for adjective noun sequences. . . 193
5.6 Summary of experimental results - Baseline of random sequences. “PoS” stands for parts of speech, “N N” for noun-noun sequences and “J N” for adjective-noun sequences. 194

6.1 Example context vectors for the words of the definition: “*contact* ≡ *close interaction*” 205
6.2 Top 20 syntactic structures of *WordNet* definitions 214
6.3 Probability of confusing positive and negative instances of the multiword expressions dataset when composing with existing CDS models 220
6.4 Probability of confusing positive and negative instances of the multiword expressions dataset when composing with BAM and EAM 221
6.5 Probability of confusing positive and negative instances of single words datasets *NN* and *VN* when composing with existing CDS models 223

6.6 Probability of confusing positive and negative instances of single word datasets *NN* and *VN* for BAM and EAM. 225

List of Figures

1.1 Multiword expression classification (Sag et al. 2002). 18

2.1 Example parse tree. 34

3.1 Evaluation framework overview 116
3.2 Example *GENIA* sentence 121
3.3 NC-Value results on *GENIA* 2-grams. J&K filter. Precision, recall and F-Score. 126
3.4 Statistical methods on *GENIA* and *PennBioIE* 3-grams. Noun filter. F-Score. 127
3.5 Statistical methods on *GENIA* 3-grams. *J&K+Ns* filter. Precision, recall and F-Score. 128
3.6 *GENIA* sequences of any length, *Nouns* filter, various methods, Precision, Recall and F-Score. 131

4.1 System overview 139
4.2 "red carpet", sense induction example 142
4.3 Proposed system and *1c1word* accuracy. 150
4.4 An example undirected weighted graph 152
4.5 Unweighted graph connectivity measures. 158
4.6 Weighted graph connectivity measures. 158
4.7 Comparison of manual parameter tuning and the two best performing graph connectivity measures. 159
4.8 Comparison of baseline system, manual parameter tuning and the two best performing graph connectivity measures 162
4.9 Unweighted graph connectivity measures. 163
4.10 Weighted graph connectivity measures. 163

5.1 Evaluation process 173

5.2 Change class *P2LMw*. 178
5.3 Change class *P2L*. 179
5.4 Change class *L2PMw*. 180
5.5 Change class *L2P*. 181
5.6 Change class *PL2P*. 182
5.7 Change class *P2PL*. 183
5.8 Change class *PN*. 184
5.9 Change class *P2P*. 185
5.10 Change class *MwA*. 187
5.11 Change percentages per change class on average and per multiword expressions category. "N N" stands for noun-noun sequences and "J N" for adjective-noun sequences. 195

6.1 Probability of confusing positive and negative instances of the multiword expressions dataset when composing with existing CDS models 220
6.2 Probability of confusing positive and negative instances of the multiword expressions dataset for BAM and EAM and for various values for parameter α (where $\beta = 1 - \alpha$) 221
6.3 Probability of confusing positive and negative instances of *NN* (up) and *VN* (down) single words datasets when composing with existing CDS models and for various values for parameter α (where $\beta = 1 - \alpha$) . . . 224
6.4 Probability of confusing positive and negative instances of *NN* (up) and *VN* (down) single words datasets when composing with BAM and EAM and for various values for parameter α (where $\beta = 1 - \alpha$) 226

Acknowledgement

I am grateful to my supervisor, Dr. Suresh Manandhar, for his support, guidance and feedback throughout the duration of this degree. I would like to thank my internal examiner, Dr. James Cussens, and my external examiner, Dr. Eneko Agirre, for their insightful comments and feedback which improved this thesis. I would also like to thank Dr. Fabio Massimo Zanzotto for spending time to discuss about my plans at a point when I was trying to form my research questions and also for the cooperation that led to chapter 6.

I would never have achieved most things in my life, including my Ph.D. research, without the unconditional support and love of my parents Katerina and Nikos and my sister Aliki. They have always supported my education and have been a great motivation to accomplish this thesis.

My partner, Foteini Stavropoulou, was kind enough to support me during this course and especially during moments of bad results and disappointment, to believe in my progress and encourage me. I am very grateful for her contribution to the final result.

I would like to express my deepest gratitude to my friend and colleague Dr. Ioannis P. Klapaftis for contributing to my knowledge and understanding of Natural Language Processing. He was there to discuss my thoughts, answer my numerous questions and even cooperate successfully many times.

I would like to thank my friends in York and in Athens for accepting me and sharing parts of their life with me.

Finally, my research work would not have been possible without the scholarship that was awarded to me by the Department of Computer Science of the University of York. Moreover, the departmental Research Committee supported me financially so as to present my work in several conferences and workshops.

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

I hereby give consent for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

Some of the material contained in this thesis has appeared in the following published conference and workshop papers:

- Ioannis Korkontzelos, Ioannis Klapaftis and Suresh Manandhar. 2008. Reviewing and evaluating Automatic Term Recognition Techniques. *In proceedings of GoTAL 2008*, Gothenburg, Sweden.
- Ioannis Korkontzelos and Suresh Manandhar. 2009. Detecting Compositionality in Multiword Expressions. *In Proceedings of ACL-IJCNLP*, Singapore.

- Ioannis Korkontzelos, Ioannis Klapaftis and Suresh Manandhar. 2009. Graph Connectivity Measures for Unsupervised Parameter Tuning of Graph-Based Sense Induction Systems. *In Proceedings of the NAACL-2009 Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, Boulder, Colorado, USA.

- Ioannis Korkontzelos and Suresh Manandhar. 2010. Can Recognising Multiword Expressions Improve Shallow Parsing? *In Proceedings of NAACL-HLT 2010*, Los Angeles, California, USA.

- Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi and Suresh Manandhar, 2010. Estimating Linear Models for Compositional Distributional Semantics. *In Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China.

CHAPTER 1

Introduction and Motivation

Executive Summary

The notion of multiword expressions is introduced and its properties are discussed thoroughly. Its fundamental characteristics are formed into research directions and supported by a discussion on motivation. Then, the objectives of our research are outlined and the research hypothesis is formally stated. Finally, we present a chapter-by-chapter overview of the thesis.

1.1 Multiword Expressions

Multiword expressions are expressions consisting of two or more words that correspond to some conventional way of saying things (Manning & Schutze 1999). They are also known as collocations; in an attempt to emphasise the frequent cooccurrence of their components. Multiword expressions appear frequently in human language, in any kind of text or speech. They can be noun phrases such as *strong tea* and *weapons of mass destruction*, phrasal verbs such as *make up*, *break up* and *give in* and stock phrases such as *rich and powerful*. The large variation that multiword expressions exhibit is a main reason why there is no unified strict definition (Rayson et al. 2009). Some definitions such as the one of Manning & Schutze (1999) focus on the usage of multiword ex-

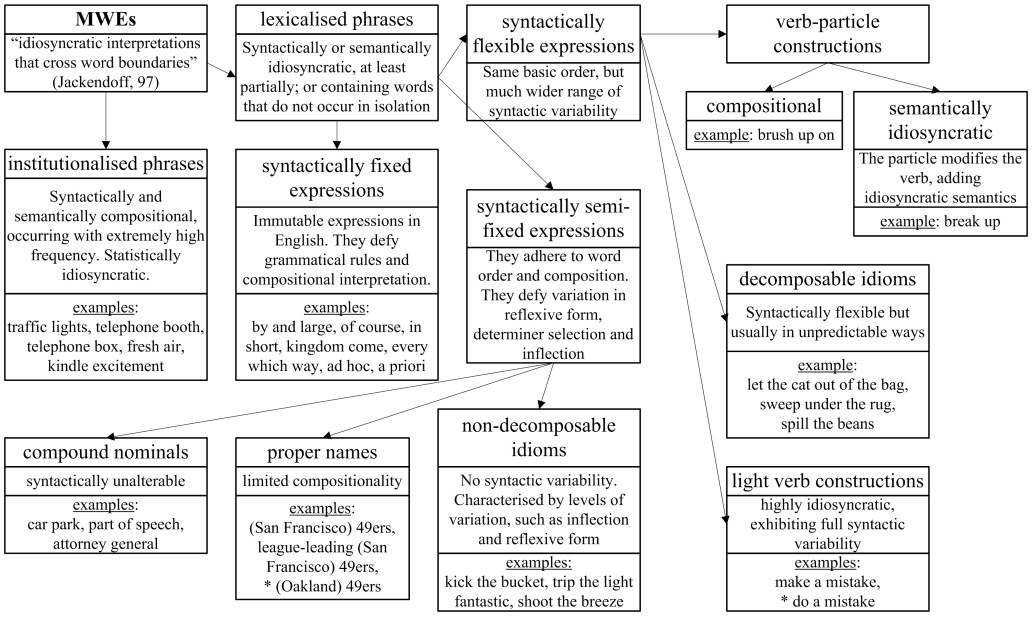

Figure 1.1: Multiword expression classification (Sag et al. 2002).

pressions. Other definitions focus on the frequency of occurrence: Baldwin et al. (2003) define multiword expressions as sequences of words that tend to cooccur more frequently than chance and are either decomposable into multiple simple words or idiosyncratic.

It is more convenient to approach the notion by spotting several characteristics of multiword expressions and in parallel inspecting the fine-grained classification presented in figure 1.1 (Sag et al. 2002). In figure 1.1, each multiword expression category is accompanied with a brief description of its basic properties and some examples.

> There is no unified phenomenon to describe but rather a complex of features that interact in various, often untidy, ways and represent a broad continuum between non-compositional (or idiomatic) and compositional groups of words (Moon 1998).

These features include various types and levels of *idiomaticity (lexico-syntactic, semantic, pragmatic and statistical)*, *institutionalisation*, *situatedness*, *identifiability*, *figuration* and *single-word paraphrasability* (Baldwin 2006).

Lexico-syntactic idiomaticity

Lexico-syntactic idiomaticity characterises fixed, immutable multiword expressions. They defy grammar rules because their components do not occur in grammatically correct positions. For example, *by and large* consists of *by*, which occurs usually as preposition and rarely as adverb, the coordinating conjunction *and* and the adjective *large*. Coordinating conjunctions always join two words of the same parts of speech or phrases of the same type. However, this rule is defied in the case of *by and large*. Other examples: *ad hoc*, *every which way*, *long time no see*, *be broke*, *be on the go*, *before long*, *Easy does it!*, *get going*, *How come?*, *Was my face red!*, *would ('d) just as soon*, *would ('d) rather*, *had ('d) better*

Semantic or pragmatic idiomaticity

Semantic or pragmatic idiomaticity characterises multiword expressions whose semantics or pragmatics significantly differ from the semantics or pragmatics of their components appearing separately. Examples: *under the weather*, *wet behind the ears*, *at the eleventh hour*, *beat around the bush*, *couch potato*, *elbow grease*, *feel blue*, *in the black*, *jump all over someone*, *keep one's nose to the grindstone*, *let sleeping dogs lie*, *pull someone's leg*.

Component words of idiomatic multiword expressions tend to cooccur with some specific words, among a large set of synonyms. These selections are called selectional preferences. Examples:

strong coffee, *??powerful coffee*	*stiff breeze*, *??rigid breeze*, *??firm breeze*
crash test, *??smash test*	*pub quiz*, *??bar quiz*, *??tavern quiz*
a quick study, *??a fast study*	*sweet dreams*, *??sugary dreams*, *??kind dreams*

Situatedness

Situated multiword expressions are associated with a fixed pragmatic point. In other words, these expressions are said at a specific time, during a specific period, at a specific place or by people that have a special property. Examples: *good morning*, *all aboard*, *Break a leg!*, *Cool it!*

Institutionalisation

Institutionalised multiword expressions are accepted as lexical items, through consistent use over time. Examples: *stiff breeze*, *??stiff wind*, *broad daylight*, *??narrow darkness*, *a piece of cake*, *bite off more than one can chew*, *can't make heads or tails of something*, *make a mountain out of a molehill*, *rain cats and dogs*, *change one's mind*.

Non-identifiability

The meaning of *non-identifiable multiword expressions* cannot be predicted from the surface form. The notion of non-identifiability is relevant to the notion of decomposability (see below). Example: *kick the bucket*, *fly off the handle*, *wet behind the ears*, *bent out of shape*, *blow one's top*, *by the skin of one's teeth*, *drag one's feet*.

Figuration

Figurative multiword expressions express some metaphor, metonymy, hyperbole, etc. Examples: *bull market*, *beat around the bush*, *hit the sack*, *in the red*, *jump to conclusions*, *lend someone a hand*, *leave well enough alone*.

Single-word paraphrasability

Some multiword expressions have the same or very similar semantics to some single words. Examples:

all right $\approx$ *fair*	*all right* $\approx$ *unharmed*
be on the road $\approx$ *travel*	*down in the dumps* $\approx$ *depressed*
get a move on $\approx$ *hurry*	*grab a bite* $\approx$ *eat*
hit the books $\approx$ *study*	*once in a while* $\approx$ *occasionally*
take it easy $\approx$ *relax*	*with bells on* $\approx$ *with additional ornament*

Translatability

Multiword expressions are usually not-translatable in other languages. ■

The above features, which interact with each other to define the degree of idiomaticity of a multiword expression (Nunberg et al. 1994), can be best summarised under the notions of *compositionality* and *decomposability*.

Compositionality

Compositionality is defined as the degree to which the meaning of a multiword expression can be predicted by combining the meanings of its components. The general notion can be split into syntactic and semantic compositionality. *Syntactic compositionality* is binary: Non-compositional multiword expressions are fixed and institutionalised (e.g. *by and large*). In contrast, *semantic compositionality* is continuous (Baldwin et al. 2003; Baldwin 2006).

Decomposability is defined as the degree to which the meaning of a multiword expression can be ascribed to the meanings of its constituents. It should be noted that although the notions of *decomposability* and *compositionality* are similar, they are not exactly the same. For example, the idiosyncratic multiword expression *"spill the beans"* is non-compositional but decomposable. Projecting the semantics of *"spill"* to *"reveal"* and the semantics of *"beans"* to *"secret"* leads to its literal interpretation. If A is a decomposable multiword expression consisting of i components, its similarity to all of its components, $sim(A, A_i)$, is expected to be high. In opposition to *compositionality*, *decomposability* can only refer to semantics. As a feature, it appears in a continuum. More examples of decomposable multiword expressions: *don't count your chickens until (before) they hatch (they've hatched)*, *get a kick out of something*, *get one's wires crossed*, *hard feelings*.

1.2 Motivation

Multiword expressions have attracted considerable attention from researchers both in terms of theory and practice. Initially, linguists described multiword expressions theoretically (Nunberg et al. 1994; Jackendoff 1997; Sag et al. 2002; Manning & Schutze 1999). Then, researchers started experimenting with this knowledge practically (Manning & Schutze 1999); however, identifying and treating multiword expressions properly has proven to be a pain in the neck for Natural Language Processing (NLP), due to lack of adequate resources such as manually annotated corpora in various languages. In recent years, there has been a growing awareness in the NLP community about problems related to multiword expressions (Sag et al. 2002). Several special interest workshops have been organised and discussed issues ranging from identifying multiword expressions and evaluation to their seamless inclusion in other NLP applications (Korhonen

et al. 2003; Fellbaum 2006; Tanaka et al. 2004; Moirón et al. 2006; Rayson et al. 2006; Grégoire et al. 2007, 2008; Anastasiou et al. 2009). Although there is substantial progress in multiword expression research, most real-world applications tend to ignore them or address them simply by listing.

Successful inclusion of multiword expressions means successful identification and treatment. For almost all NLP topics this is still not realised. In particular, this applies to NLP applications which require some degree of semantic processing, such as machine translation, question-answering (query segmentation and expansion), summarisation, lexicography, terminology, statistical parsing, language generation, language modelling in speech processing, word sense disambiguation and text simplification. The effect of successful handling of multiword expressions within NLP applications depends on the frequency of occurrence of multiword expressions. The improvement is expected to be more significant for non-compositional multiword expressions due to the difference in meaning between the multiword expression and its components.

1.3 Research Subjects and Analysis

As highlighted in the previous sections, identifying and treating multiword expressions are important fields of Natural Language Processing. In this thesis we focus on the following four main research directions:

- unsupervised recognition of multiword expressions
- unsupervised methods to decide compositionality of multiword expressions
- incorporating multiword expression knowledge into other NLP tasks
- use of multiword expressions for aiding research in other NLP tasks

Recognising multiword expressions is the fundamental task of the field. It could be potentially successfully approached in a supervised way, i.e. using a manually annotated training corpus to learn the characteristics (features) of multiword expressions as far as their structure and contextual environment is concerned. In succession, this knowledge would be used so as to locate multiword expressions that occur in another unannotated text.

However, there are several reasons to believe that supervised multiword expression recognition is of limited functionality and practical use. A major reason is that multiword

expressions vary largely in different styles and types of text. For example, multiword expressions of newswire articles are quite different to multiword expressions of biomedical articles. Moreover, multiword expressions of colloquial, everyday speech are much different to those occurring in some scientific text. Due to the idiomatic nature of multiword expressions it is debatable whether domain adaptation can be applied. As a result, to successfully recognise multiword expressions of different domains one would need large amounts of text annotated with multiword expressions for every different domain and text style.

As discussed in section 1.1, multiword expressions are rarely translatable to other languages. The morphological and contextual features of multiword expressions in a specific language, would be very different to the features of multiword expressions in another language. This limits the applicability of supervised multiword expression recognition even further. One would need different training examples not only for every different domain and text style, but also for every language of interest.

For these reasons, we focus on unsupervised multiword expression recognition. "Unsupervised" here means that the methods do not need any pre-annotated text for training. They are based on frequency statistics and other structural observations about multiword expressions. Some of these observations are not entirely language unspecific but transferable to other languages, meaning that they could be easily adapted after trivial modification. For example, many multiword expressions in English consist of an adjective followed by a noun. In contrast, an adjective might occur immediately after a noun in French. Given this fact, a system developed for English can easily be adapted to French.

Despite the arguments in favour of unsupervised term recognition, it should be noted that in some cases supervised term recognition is valuable. In languages and domains in which there are annotated data available and for applications in which language and domain transferability is not an issue supervised methods are more suitable than unsupervised ones, given that the former are usually shown to achieve more accurate results.

As discussed in section 1.1, the notion of compositionality can be seen as the characteristic of multiword expressions that is mostly responsible for the similarity or diversity of their meaning from the composition of the meanings of their components. In simple words, compositionality captures whether a multiword expressions should be considered as idiosyncratic or not. In the previous section, we discussed a number of NLP tasks which would be potentially improved by the inclusion of multiword expres-

sion knowledge. However, for compositional multiword expressions this might not be true; it is more likely that they would not affect the quality of the result either positively or negatively: For example, knowing that *computer science* is a multiword expression is highly likely not to affect how a parser would parse sentences containing this multiword expression. In contrast, knowing that *by and large* is a multiword expression would allow a parser to accept it as a unit within which grammar rules are defied. In addition, lexicographers are clearly interested in knowing which multiword expressions are non-compositional so as to include them in lexicons and dictionaries, while they would not be equally interested in compositional multiword expressions. For these reasons, it is an issue of great importance to do research towards systems able to decide whether a given multiword expressions is compositional or not.

Approaching the issue in a supervised way is of limited practical use for similar reasons to those of recognising multiword expressions. One would need huge amounts of human annotated data to cover all languages of interest and all domains and text styles in each of these languages. Manually annotating multiword expressions for compositionality requires expert or at least high-level knowledge and is therefore costly. Considering annotation in languages that are not widely-known highlights the limited applicability of supervised approaches to resolving the compositionality of multiword expressions. Instead, we choose to address the issue in unsupervised ways. We build on the definition of compositionality by comparing the meaning of the multiword expression to the meaning of its components. We mark as compositional multiword expressions the ones whose meaning is similar to the meaning of their components or as non-compositional in the opposite case.

Despite their disadvantages discussed above, supervised methods can be useful for a variety of other tasks (e.g. for adjusting parameters). Commonly, supervised methods are used for tasks where language and domain transferability are not important issues and annotated data are available.

In general, for every research subject it is a major issue to show experimental results supporting the contribution of this research to other fields, uses or applications. In the case of multiword expressions, it is important to show that our intuitions and expectations about how multiword expressions knowledge would contribute to other NLP tasks actually hold. The contributions should ideally be evaluated on tasks or applications that are in wide and everyday use. The more important the application to the community the

larger the significance of contributions.

The third research direction aims to inspect whether multiword expression recognition can contribute directly to other NLP tasks. Contrarily, the fourth research direction aims to inspect indirect contributions of recognising multiword expressions and deciding their compositionality to NLP research. Datasets annotated with special information such as compositionality are useful to define evaluation frameworks for other NLP tasks and assess the accuracy of different methods.

In section 1.1 we discussed the large variety of multiword expressions and supported the presentation with numerous examples. Due to the diversity of characteristics of multiword expressions relevant to either structure or semantics, it is extremely difficult to address all four research directions for all multiword expression classes. Restricting the multiword expressions of interest is useful because it widens the options of tools that can be used for multiword expression recognition and treatment. For example, regular expression filtering can be applied to the part of speech sequence of text to identify multiword expression candidates only if it is known beforehand that the multiword expressions of interest consist of some parts of speech appearing in some specific order. We choose to restrict the range of multiword expressions taking into account only expressions whose constituents form uninterrupted sequences (Shimohata et al. 1997). For example, the multiword expression *head of computer science department* is considered, while the light verb construction *give something a try* is not, because there is a noun phrase intervening between *give* and *a try*. Any further restrictions will be clearly stated on occasion if hypothesised.

1.4 Research Objectives and Hypothesis

The research objectives of this thesis can be summarised as follows:

- Our first objective is to investigate state-of-the-art unsupervised approaches in multiword expression recognition and assess them under a common evaluation framework. Further, we intend to analyse each method into its components in terms of distinct types of information that they take into account and assess these components separately. This will make clear what source of information or combination of sources best correlate with the probability that a candidate is an actual multiword expression.

- The second objective is to exploit the task of determining the compositionality of a given multiword expression. Our inspection aims to propose an entirely unsupervised method to address this task, evaluate its accuracy against gold-standard data and discuss its behaviour.

- Thirdly, we aim to investigate how multiword expression knowledge can be integrated into systems performing other fundamental NLP tasks. In particular, we intend to propose a method to incorporate multiword expressions into shallow parsing and assess its improvement or decline.

- The last objective is to use properties of multiword expressions such as compositionality so as to aid research in other NLP tasks. In particular, we explore the idea that multiword expressions can provide an evaluation platform for approaches to distributional semantics composition, a fairly new and interesting field of NLP research.

To summarise the introductory part we attempt to express the research hypothesis of this thesis as experimental procedure:

> *We hypothesise that the tasks of recognising multiword expressions and deciding their compositionality can be addressed in unsupervised manners, based on cooccurrence statistics and distributional semantics. Further, multiword expressions are beneficial for other fundamental applications of Natural Language Processing either by direct integration or as an evaluation tool.*

1.5 Thesis Organisation

The thesis is organised as follows:

Chapter 2 presents a detailed description of approaches to unsupervised multiword expression recognition and unsupervised approaches to resolving compositionality. It focuses on the limitations of current approaches to highlight the motivation for our research. The chapter also includes a small review on composition of distributional semantics. This part is necessary background to chapter 6, where the properties of multiword expressions serve as a platform to define an evaluation framework and help in

exploiting the task of composition of distributional semantics. A review on distributional semantics and similarity metrics is also included, because this field is used as a tool throughout our research.

Chapter 3 proposes an evaluation framework to assess different multiword expression recognition approaches. A number of approaches presented in chapter 2 as well as the components into which they are analysed are assessed under this common framework.

Chapter 4 presents an unsupervised approach to resolving compositionality of multiword expressions. It uses sense induction and distributional similarity. The evaluation is done on a set of compositional and non-compositional multiword expressions, which is in turn extracted from WordNet (Miller 1995) in a semi-supervised manner. The chapter also investigated unsupervised ways to perform parameter estimation of sense induction systems.

Chapter 5 proposes a framework to integrate multiword expression knowledge in another fundamental NLP task, shallow parsing. Evaluation is performed in an unsupervised manner, based on a classification of shallow parsing errors and on hypotheses that describe the characteristics of successful shallow parses against unsuccessful ones.

Chapter 6 takes advantage of multiword expressions and the property of compositionality to define an evaluation framework for composition of distributional semantics. State-of-the-art approaches to this task are evaluated under this framework. Experimental evidence spotlights the strengths and weaknesses of each approach. Moreover, a supervised setting for composing distributional semantics is proposed, achieving superior results.

Chapter 7 summarises the contributions of this thesis and presents the conclusions along with a discussion of open issues and future research directions.

CHAPTER 2

Background and Field Review

2.1 Introduction

This chapter aims at presenting past and recent approaches relevant to the research objectives of this thesis. The presentation focuses on inspecting the limitations of approaches in the literature and highlights issues that strengthen the motivation for our research and are addressed in the following chapters. This chapter is divided into four parts:

- Multiword expression recognition
- Distributional similarity
- Multiword expression compositionality
- Distributional semantics composition

Section 2.2 presents a detailed survey on methods for multiword expression recognition. We mainly focus on unsupervised methods aiming to identify multiword expressions whose constituents occur successively to form a sequence. However, for completeness we mention a few other important approaches.

We categorise the approaches in the literature in terms of the different types of linguistic information that they take into account as *linguistic*, *statistical*, and *hybrid*.

(a) Linguistic approaches use morphological, grammatical and syntactical knowledge to identify multiword expression candidates or preprocess text. (b) *Statistical* approaches analyse occurrence statistics of words or sequences. They are further classified into *termhood-based* or *unithood-based*. *Termhood-based* approaches attempt to measure the degree that a candidate multiword expression is a term; in other words, refers to a specific concept. *Unithood-based* approaches measure the attachment strength among the constituents of a candidate multiword expression. (c) *Hybrid* approaches efficiently combine linguistic and statistical components so as to take into account as much information as possible towards recognising multiword expression candidates.

In terms of evaluation, we discuss the extent to which direct comparison of different approaches is possible. We conclude that the approaches take various different sources of information into account that are not evaluated independently. Moreover, it is very difficult to compare among different approaches because they are evaluated using incompatible evaluation settings and resources.

In section 2.3 we present a small review on distributional semantics and distributional similarity measures. Distributional semantics provides the connection between the context of a target word and its meaning. The meanings of two target words can be compared by comparing the representations of their contexts. We analyse important issues of constructing context representations and review the most widely used measures of distributional similarity. Although distributional semantics is not in primary focus, we include this review because distributional semantics is used in most experimental settings and evaluation procedures throughout this thesis. This review provides the reader with the relevant background.

Section 2.4 focuses on approaches in the literature that attempt to decide compositionality of multiword expressions. Due to the fact that this field of research is not very explored these attempts are quite restricted. We focus on inspecting the limitations of these approaches in terms of types of multiword expressions that they apply to (see figure 1.1) and we classify them in classes of addressing the issue from different points of view. Also, we discuss the degree of supervision of these approaches.

In section 2.5 we present a survey on distributional semantics composition. This fairly new research field attempts to compose the distribution of context features of a sentence, phrase or syntactic unit, given the context distributions of its component words. We focus on the intuitions behind the approaches in the literature and conclude that most

approaches are weakly evaluated. We identify the lack of a framework able to provide positive and negative evaluation instances. This framework would provide a basis to estimate the accuracy of different methods and directly compare them.

2.2 Survey on Multiword Expression Recognition

Multiword expression recognition is a subfield of natural language processing research that has been explored at a significant level. The nature of the task allowed for methods and tools, that were developed earlier for other fields, to be applied for its purposes. *Collocation extraction*, *term recognition* and *keyphrase extraction* were explored earlier than *multiword expression recognition* and served as repositories of methods that could potentially be modified and applied for this task.

Collocations are groups of words that tend to occur together in text more often than by chance. Some examples are: *New York*, *vice president*, *stock exchange*. The notion of *collocation* is different to the notion of *multiword expression*. Sometimes, these notions are interchangeable in the literature, such as in Choueka (1988). They both refer to strongly related words, but this relation focuses on different aspects: the definition of *collocations* focuses at their increased cooccurrence frequency, while most of the definitions for *multiword expressions* focus on their varying levels of idiomaticity. As a result, the set of all potential collocations and the set of all multiword expressions are different, but certainly overlapping. For example, *New York* is a collocation and a multiword expression, because it refers to a different city than *York, UK*. In particular, excluding closed class word sequences, most collocations are multiword expressions, because they correspond to some conventional way of saying things (Manning & Schutze 1999). However, not all multiword expressions are collocations for all types of texts. For example, some idiomatic expressions do not occur very often in scientific articles, e.g. *crash test*. Due to the definition of collocations, plenty of statistical methods were used to assess the statistical significance of the cooccurrences of word group or sequence candidates, e.g. *hypothesis testing*. In subsection 2.2.2, we review the fundamental statistical approaches that are most widely used in multiword expression recognition.

Terms are words or sequences of words that verbally represent concepts of some specific domain of knowledge, usually scientific of technical (Kageura & Umino 1996). *Multiword terms* are a subset of multiword expressions because they consist of frequently cooccurring components which are most of the times non-compositional, non-

substitutable and non-modifiable. Some examples from the domain of biochemistry and molecular biology are: *cell cycle*, *kinetic assay* and *linkage map*. *Term extraction* approaches can be used for multiword expression recognition. We review several of these approaches in subsection 2.2.2.2.

Keyphrase extraction is the task of finding a set of words or phrases that represent the core-concepts of the document or summarise it. Multiword keyphrases are clearly collocations and also multiword expressions. Keyphrases can be either occurring in the document itself or not; in the latter case, the keyphrase should be closely related to other words in the document or a part of it. In the former case, *keyphrase extraction* methods can be applied to multiword expression recognition (e.g. Witten et al. (1999); Turney (2003)).

Term recognition and *keyphrase extraction* are based on the following basic principles (Kageura & Umino 1996; Ananiadou 2001):

- Tokens that appear together most of the time are likely to constitute a collocation.
- A frequent word or sequence of words occurring in a document is likely to be a term.
- A token which appears frequently in a domain is likely to be a term of this domain.
- A token which appears relatively more frequently in a specific class of documents is likely to be a term for this class of documents.
- A token which appears relatively more frequently in a specific domain is likely to be a term of this domain.
- A token whose occurrence is biased in some way to (a) domain(s) is likely to be a term.

In subsection 2.2.1 we review a handful of linguistic components that are used broadly in multiword expression recognition mainly as preprocessing components. In subsection 2.2.2 we present statistical approaches for this task. Finally, in subsection 2.2.3 we discuss some hybrid systems that combine selected linguistic and statistical components. The statistical components of some of these systems will be described in detail in subsection 2.2.2.

2.2.1 Linguistic approaches

Linguistic approaches for collocation extraction and term recognition include a handful of linguistic processing components. Most of these components correspond to basic Natural Language Processing tasks which were explored in the past and were solved almost perfectly or at least adequately accurately. They aim to remove textual noise and in general change the input raw text so as to make more substantial the frequency counts that will be counted latter on this text.

Preprocessing methods range from very simple ones, such as turning all characters of the text into lower-case, to sophisticated ones, such as resolving abbreviations and *NP syntax normalisation*, the process of projecting different ways of forming a noun phrase to one, e.g. e.g. *activity of enzymes* $\rightarrow$ *enzyme activity*. Below we introduce a number of preprocessing components useful for collocation extraction and manipulation. In particular, tokenisers, part of speech taggers, lemmatisers, parsers, and part of speech filters. We also include a discussion about the role of context types and gazetteers.

2.2.1.1 Tokenisation

Sentence splitters and tokenisers are components that input text and separate its sentences and words, respectively. Usually after tokenisation separate words are called tokens. Sentence splitters output a list of separated sentences, while tokenisers output a list of tokens for each input sentence.

Sentence splitting and tokenisation is the first step of corpus preparation or preprocessing. Although is seems trivial, in many cases it needs special attention. For example, splitting sentences at full stops would fail with sequences such as *'Dr. Who'*. Tokenisation is by itself an important means of improving the frequency counts computed on a corpus. Moreover, it is a prerequisite for other preprocessing components such as part of speech tagging and parsing.

2.2.1.2 Part of speech tagging

Part of speech tagging is the task of assigning one part of speech tag to each token of an input sentence. The part of speech tags are known beforehand. Part of speech tagging as a field of research started being exploited some years ago. A lot of supervised and unsupervised approaches have been proposed. The most successful ones are more

	WSJ corpus	*GENIA* corpus
A tagger trained on the *WSJ* corpus	97.05%	85.19%
A tagger trained on the *GENIA* corpus	78.57%	98.49%
GENIA tagger	96.94%	98.26%

Table 2.1: *GENIA* tagger accuracy.

than 95% accurate per word. However, the problem cannot be considered as solved, since per sentence accuracy remains poor. For example, the accuracy of *GENIA* tagger (Tsuruoka et al. 2005) is shown in table 2.1. The accuracy of *GENIA* tagger is presented in comparison with other taggers, trained on various corpora. *GENIA* tagger was trained on the *Wall Street Journal* corpus (*WSJ*), on the *GENIA* corpus and on the *PennBioIE* corpus. The former is a corpus of general-purpose text while the latter two are corpora of the biomedical domain.

Part of speech tagging is an important part of corpus preprocessing. Essentially, it provides classification of tokens into a small number of relatively large classes. Each class represents a part of speech, such as nouns, verbs, or a finer-grained subclass, such as nouns in plural and intransitive verbs. Based on this classification, sometimes researchers chose to discard some classes and work with the remaining tokens. Moreover, frequency of tokens of a specific class can be counted and used for various purposes of statical processing, for example as a back-off model for unknown words.

2.2.1.3 Lemmatisation

Lemmatisation is the task of analysing words so as to remove any inflectional prefixes or suffices and finally retrieve its basic form. For example, *'drivers drove cars'* would be lemmatised as *'driver drive car'*. Similarly to part of speech tagging, lemmatisation is a fundamental task of natural language processing that has been very well solved in the past.

Lemmatisation classifies tokens in far more fine-grained classes than part of speech tagging. It results in many small classes of extremely relevant words; that come from the same linguistic root. This is very useful in obtaining more reliable frequency counts by alleviating the detrimental effects of data sparseness.

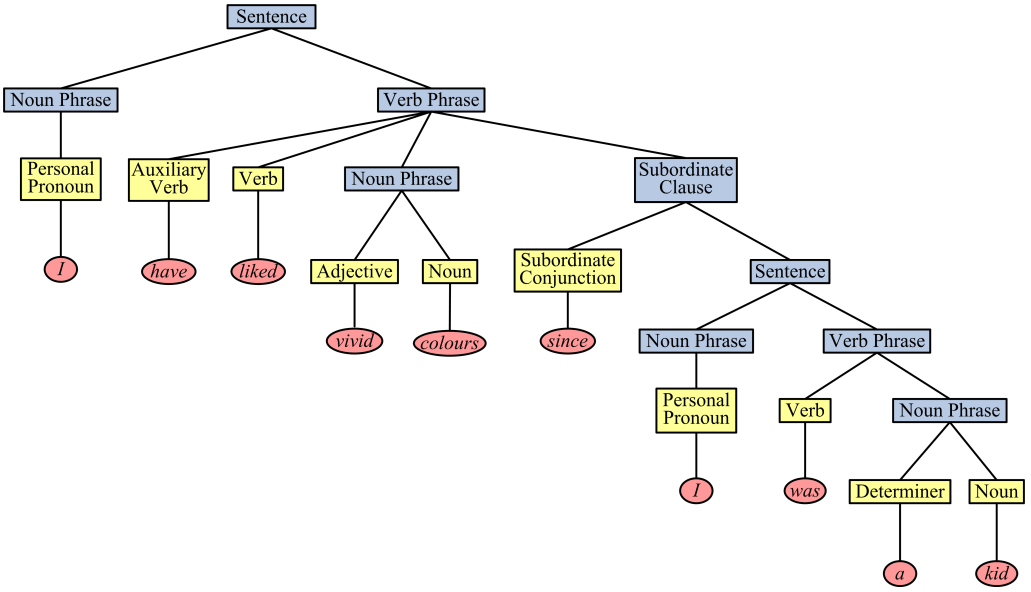

Figure 2.1: Example parse tree.

2.2.1.4 Parsing

Parsing or syntactic analysis is the process of analysing text to determine its grammatical structure with respect to a given grammar. Input text is tokenised and sometimes can be lemmatised as well. For each input sentence a parser outputs an analysis of the underlying structure of text. For example, the sentence *"I have liked vivid colours since I was a kid."* whose corresponding parts of speech sequence is *"Personal pronoun, auxiliary verb, verb, adjective, noun, subordinate conjunction, personal pronoun, verb, determiner, noun"* would be parsed as shown in figure 2.1.

There are several different kinds of parsers for natural language, exhibiting different levels of detail of produced syntactic analysis and accordingly different levels of computational complexity. Deep parsers fully analyse the input text and are usually computationally intense. Their output retains all information contained in the ideal parse tree, e.g. figure 2.1, and is usually shown in some encoding as the following:

(Sentence
(Noun Phrase (Personal Pronoun ***I***))
(Verb Phrase (Auxiliary Verb ***have***) (Verb ***liked***)

(Noun Phrase (Adjective ***vivid***) (Noun ***colours***))
(Subordinate Clause (Subordinate Conjunction ***since***)
(Sentence
(Noun Phrase (Personal Pronoun ***I***))
(Verb Phrase (Verb ***was***)
(Noun Phrase (Determiner ***a***) (Noun ***kid***))))))
(. .))

Another type of parsers, dependency parsers, aim to identify the relations between the tokens of each given sentence instead of retrieving the whole syntactic tree. Dependency parsers are computationally less intensive than deep parsers, but their output is less informative. For example, the previous example sentence would be ideally analysed as:

I [1]	- is subject of -	***liked [3]***
have [2]	- is auxiliary of -	***liked [3]***
vivid [4]	- is adjectival modifier of -	***colours [5]***
colours [5]	- is direct object of -	***liked [3]***
since [6]	- introduces an adverbial clause governed by -	***kid [10]***
I [7]	- is subject of -	***was [8]***
kid [10]	- is copula of -	***was [8]***
a [9]	- is determiner of -	***kid [10]***
kid [10]	- governs an adjectival clause governed by -	***liked [3]***

The numbers within square brackets next to each token represent the position offset of each constituent in the sentences to resolve possible conflicts.

A third type of parsers are shallow parsers also known as chunkers. They aim to slice each input sentence into the phrases that it contains. In other words, they are only resolving the shallow structure of the sentence, in contrast with deep parsers. Shallow parsers retain even less information about syntax than dependency parsers, but are generally much less computationally intensive. The shallow parsing output of the previous example would be:

Noun Phrase (Personal Pronoun ***I***)
Verb Phrase (Verb ***have liked***)
Noun Phrase (Adjective ***vivid***) (Noun ***colours***)

Subordinate Clause (Subordinate Conjunction ***since***)
Noun Phrase (Personal Pronoun ***I***)
Verb Phrase (Verb ***was***)
Noun Phrase (Determiner ***a***) (Noun ***kid***)

Parsers serve as components to multiword expression recognition and manipulation systems. The state-of-the-art accuracy of deep and dependency parsers is around 90% (Klein & Manning 2003). Parsing gives the opportunity to clean the context of a given word much further than part of speech tagging and lemmatisation. In particular it allows selecting from all context only words that are in some predefined relation with the target word. In a number of research works of the literature, experiments are based on the *selectional preferences* of a word. *Selectional preferences* are usually defined as the nouns that appear in subject or object relation with some verb, or as the verbs that occur in subject or object relation with a noun or a noun phrase. (Resnik 1993; Fazly et al. 2009; Mason 2004; Schulte im Walde 2003; McCarthy & Carroll 2003; Erk 2007; Wang et al. 2007; McCarthy et al. 2007).

2.2.1.5 Part of speech patterns

Part of speech patterns are regular expressions based on part of speech tags. They are applied on the part of speech sequence of text, thus part of speech tagging is a prerequisite. Part of speech patterns are used to identify subsequences of text that have some specific part of speech structure. In the field of recognising multiword expressions they can be used to identify candidates, given that the part of speech structure of desirable multiword expressions is previously known.

Part of speech patterns were used to identify pairs of nouns or noun sets that are in *is-a* relationship to each other (Hearst 1992, 1998). These patterns contain both parts of speech and words in their surface form. For example sequences that satisfy the pattern:

$$NP_0 \text{ such as } NP_1, NP_2 ... (\text{and}|\text{or}) NP_n \tag{2.1}$$

imply that:

$$\forall NP_i, i \in [1, n] : \text{ hyponym}(NP_i, NP_0) \tag{2.2}$$

Other part of speech patterns that recognise hyponyms of NP_0 are:

$$NP_1 \text{ is a (kind of) } NP_0 \tag{2.3}$$

$$\text{such}NP_0 \text{ as } NP, *(\text{and}|\text{or})NP \tag{2.4}$$

$$NP(,NP)*, (\text{and}|\text{or}) \text{ other } NP_0 \tag{2.5}$$

$$NP_0, \text{ including } (NP,)*(\text{and}|\text{or})NP \tag{2.6}$$

$$NP_0, \text{ especially } (NP,)*(\text{and}|\text{or})NP \tag{2.7}$$

Part of speech patterns were also used in Justeson & Katz (1995), in an attempt to identify domain-specific term candidates. Term candidates satisfy the following pattern:

$$(\ (A|N)^+ \mid (A|N)^*\ (NP)?\ (A|N)^*\ )\ N \tag{2.8}$$

where N stands for *Noun*, A for *Adjective* and P for *Preposition*. The pattern accepts sequences of adjectives and nouns that end with a noun. Optionally, the pattern accepts pairs of the sequence described above joined with a preposition. Justeson & Katz (1995) report that in the corpus they assembled from various technical terminology dictionaries 97% of the multiword noun phrases consist of nouns and adjective only, and more than 99% consist of nouns, adjectives, and the preposition *"of"*.

2.2.1.6 Context

In this subsection, we discuss the issue of choosing the aspects of context for multiword expression recognition and manipulation. Exceptionally we present this discussion here, although it does not refer to any linguistic component but a classification of different context types. Séaghdha & Copestake (2007) identify three distinct types of context of multiword expressions:

- *Type context*: The contexts in which instances of the multiword expression appear; e.g. all sentences in the corpus that contain the target multiword expression *"sweet dreams"*.

- *Word context*: The contexts in which instances of each constituent of the multiword expression appears; e.g. all sentences containing *"sweet"* or *"dreams"*.

- *Relation context*: The contexts in which both constituents of the multiword ex-

pression appear; e.g. all sentences containing both *"sweet"* and *"dreams"* but not necessarily in succession.

- *Token context*: The context in which the particular instance of the multiword expression token was found.

Each type of context can be used in multiword expression recognition and manipulation for different purposes. *Type context* is used where all instances of a multiword expression are treated in the same manner. In contrast, in the case that different instances need to be classified into several different classes *token context* should be used. For example, Fazly et al. (2009) argue that idiomatic expressions do not always appear as idiomatic; but instead are idiomatic in some contexts only. They support a per-instance view to the issue of compositionality of idiomatic expressions. Their argument strongly depends on the fact that idiomaticity is not binary but appears in various levels. Consequently, they measure the idiomaticity of an instance by quantifying how different the *token context* distribution of the instance is from the corresponding *type context*.

The choice of context type is very important for all methods that use distributional similarity, i.e. they describe a class of instances by the frequencies of occurrence of other words in their context and then compare these descriptions to extract meaningful results. Context types define the classes of instances that are unified to compute their context distribution jointly. In chapter 2.3 we present distributional similarity measures of the literature in detail.

2.2.1.7 Gazetteers

Gazetteers are repositories of multiword expressions. To go back to the first approaches in recognising multiword expressions, previously populated lists were extensively used to recognise instances in the input text. This technique was extensively used in many tasks relevant to recognising instances or sequences of tokens with some specific properties.

However, using gazetteers creates a number of problems: they are expensive because they require large amount of human effort to create and they are difficult to maintain because of the addition of new items. Gazetteers containing multiword expressions suffer from an extra drawback; due to the syntactic flexibility of several multiword expression classes, the gazetteers should not contain just the surface form of the expression but in-

stead an encoding of it so as to be matched with parsing output. Consequently, gazetteers are dependent on the output representation of a specific parser.

For these reasons, there was significant research effort to reduce the size of necessary gazetteers using bootstrapping and extracting rules from few available instances (Mikheev, Moens & Grover Mikheev et al.). Some systems use only a few seeds to extract from the web instances similar to the seeds and thus automatically create gazetteers (Nadeau et al. 2006; Etzioni et al. 2004, 2005; Banko et al. 2007). Researchers have also proposed encodings of multiword expressions that capture their syntactic flexibility and are compactly represented (Villavicencio et al. 2004).

2.2.1.8 LEXTER (Bourigault 1992)

Bourigault (1992) presented an early, pure linguistic approach for recognising domain specific multiword expressions. The approach works in two steps: (a) the first step locates candidate multiword expressions by identifying boundary words, i.e. words that are known to mark the beginning or the end of a multiword expression; and (b) the second step parses each candidate to analyse it into subsequences.

In the first stage, the proposed system, LEXTER uses negative knowledge about the desired multiword expressions, i.e. parts of speech that never occur in a valid multiword expression. It applies part of speech patterns similar to those discussed in section 2.2.1.5, but accepts as candidates all sequences on which the patterns do not apply. Such patterns are regular expressions made of conjugated verbs, pronouns, conjunctions, and part of speech sequences such as “preposition-determiner”, etc.

In the second stage, LEXTER parses candidate multiword expressions to analyse them in subgroups which are possibly terms due to their grammatical structure and their position in the maximal-length candidate. LEXTER uses its own parsing module, made up of hand-written parsing rules.

2.2.1.9 Morpheme bootstrapping

Heid (1999) proposed a multiword expression recognition approach, that uses boosting on domain-specific morphemes to overcome data sparsity in German. They combine linguistic components with a statistical approach; relative frequency comparison, against a reference corpus. The linguistic components perform tokenising, part of speech tagging, lemmatisation and pattern matching with regular expressions.

The linguistic bootstrapping approach consists of 3 steps. Step (1) uses relative frequency to identify domain-specific morphemes; prefixed and suffixes. Step (2) identifies single word candidates, by applying regular expressions that contain these morphemes. Finally step (3) extracts multiword expressions by applying regular expressions that describe part of speech and lemma sequences. Then the single term candidates of step (2) are used to filter out irrelevant sequences.

2.2.2 Statistical approaches

Statistical approaches for recognising and manipulating multiword expressions consist of applications of various statistical tools. All these tools input frequency counts of words, tokens, N-grams, cooccurrences of words, etc. and features that capture the context of instances for various context types (subsection 2.2.1.7). Statistical approaches process the frequency counts and context distributions in many diverse ways and output judges that characterise candidate multiword expressions as such or output scores that quantify useful features of multiword expressions, such as decomposability and compositionality.

Kageura & Umino (1996) define two important concepts relevant to *term recognition*. The first one, *unithood*, refers to the degree of strength of syntactic combinations or collocations. The second, *termhood*, refers to the degree that a candidate term is related to a domain-specific concept. For example, in an eye-pathology corpus, *"soft contact lens"* is a valid term, which has both high *termhood* and *unithood*. However, its frequently occurring substring *"soft contact"* will have a high *unithood* and a low *termhood*, since it does not refer to a key domain concept. In the following subsections, we present the most important statistical approaches in the literature classifying them as *unithood-based* (section 2.2.2.1) or *termhood-based* (section 2.2.2.2) (Kageura & Umino 1996).

2.2.2.1 Unithood-based approaches

Unithood-based methods attempt to identify whether the constituents of a multiword candidate term form a collocation rather than cooccurring by chance (Kageura & Umino 1996).

2.2.2.1.1 Occurrence and cooccurrence frequency counts

The simplest method of finding collocations in a text corpus is by counting the frequency of N-grams occurring in the corpus. It is useful for fixed collocations, only, meaning that among the collocation words there cannot be other words that are not parts of the collocation.

N-gram frequency is not very helpful when applied on unprocessed text, because it cannot take into account word variation such as number or gender. For example, *pub quiz* and *pub quizzes* would be counted as different collocations. Usually, N-gram frequency is applied in combination with part of speech tagging, stemming and probably some part of speech filter (see subsection 2.2.1).

In the literature, it is common to count frequencies of more than one word or N-gram sequence and output system decisions which depend on statistics and computations based on these counts. In these cases, we refer to these frequencies as cooccurrence counts. Cooccurrence counts are usually employed to assess how strong the connection between two words or concepts of language are. For example, suppose that *jeans* occurs more frequently than *suit* in some corpus. However, if *businessman* cooccurs with *suit* in more sentences than it does with *jeans*, then one can conclude that the connection between *businessman* and *suit* is stronger.

Another statistical tool useful in collocation extraction is relative frequency. Relative frequencies can be used to find collocations which are characteristic of a corpus, when comparing to other corpora. If f_i, c_i and n_i are the frequency of occurrence of a given n-gram in corpus i, its count of occurrence and the total tokens of corpus i, respectively, the relative frequency ratio is defined as:

$$r = \frac{f_1}{f_2} = \frac{\frac{c_1}{n_1}}{\frac{c_2}{n_2}} = \frac{c_1 n_2}{c_2 n_1} \tag{2.9}$$

2.2.2.1.2 Mean and Variance

The mean and variance technique can be employed to capture collocations consisting of words in a more flexible relationship to each other. It is able to capture collocations which allow for other words to occur among their component words. For example: *knock on **somebody's** door*. This technique can be specified for different N-gram lengths. Below, we present the computation for the bigram case:

Suppose that a collocation candidate, consisting of two tokens, occurs n times in some text corpus. Let d_i be the signed distance, or offset in other words, between the collocation tokens in their i^{th} occurrence. The *mean* is defined as the average of these offsets:

$$\overline{d} = \frac{\sum_{i=1}^{n} d_i}{n} \tag{2.10}$$

The *variance* measures how much the individual offsets deviate from the mean. Its square root is the *standard deviation* of the offsets:

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n} \left(d_i - \overline{d}\right)^2}{n-1}} \tag{2.11}$$

Mean and variance can be used to discover collocations by looking for pairs with high frequency and low deviation. The lower the deviation is, the more often the pair of words occurs at about the same distance. The computational efficiency of this technique clearly depends on the size of the collocational window, within which mean and variance is computed. Computations can be very demanding for large window sizes.

2.2.2.1.3 Hypothesis Testing

Collocations are defined as sequences whose component words cooccur more frequently than by chance. *Hypothesis testing* provides the statistical framework for comparing the frequency of occurrence of an event with the frequency of it by chance. In other words, various hypothesis testing methods assess whether or not something is a chance event.

The basic framework is the following: We first form the *null hypothesis* (H_0). Then we compute the probability, p, of the event if H_0 was true and we reject H_0 if p is too low (typically beneath a *significance level* of $p < 0.05, 0.01, 0.005$ *or* 0.001).

For collocation and term extraction the *null hypothesis* is independence; defined as the case that there is no association between the words, beyond occurrences by chance. The hypothesis can be written for any given sequence length. Below, we focus on the bigram case. Let w_1 and w_2 be the component words of a collocation candidate. The

independence hypothesis is:

$$P(w_1 w_2) = P(w_1) P(w_2) \tag{2.12}$$

There are several different hypothesis testing methods. In this review we include the most widely used ones and discuss their advantages and disadvantages.

2.2.2.1.3.1 The t test

This is a statistical test widely used for collocation extraction purposes. It is a function of the difference between observed and expected means, scaled by the variance. The test indicates the probability of getting a sample with the observed t test value (or one more extreme), assuming that the sample is drawn from a distribution with mean μ.

If $\overline{x}$ is the sample mean, s^2 is the sample variance, N is the sample size and μ is the mean of the distribution, we compute the following statistic, whose values correspond to confidence levels:

$$t = \frac{\overline{x} - \mu}{\sqrt{\frac{s^2}{N}}} \tag{2.13}$$

The basic disadvantage of the *t* test is that it assumes that probabilities are normally distributed, which is never true, since they are bounded to be in the interval $[0, 1]$. However, probabilities can be approximately normally distributed which means that using a t test may be reasonable in some circumstances. For example, consider the following probability computation for words w_1 and w_2 which occur 15828 and 4675 times, respectively, in a corpus of 14307668 words:

$$P(w_1) = \frac{15828}{14307668}, \quad P(w_2) = \frac{4675}{14307668} \tag{2.14}$$

The independence hypothesis computation gives:

$$H_0 : \; P(w_1 w_2) = P(w_1) P(w_2) \approx 3.615 \times 10^{-7}$$

The process of randomly generating bigrams and assigning 1 if the bigram is $w_1 w_2$ or 0

otherwise is a **Bernoulli trial** with:

$$p = \mu = 3.615 \times 10^{-7}, \quad \sigma^2 = p\,(1-p) \approx p \tag{2.15}$$

$$\overline{x} = \frac{8}{14307668} \approx 5.591 \times 10^{-7}, \quad t \approx 0.999932 \tag{2.16}$$

The approximation about σ^2 holds for small values of p. The value of *t* is not larger than 2.576, the critical value for $\alpha = 0.05$, so the null hypothesis cannot be rejected.

2.2.2.1.3.2 The t test for differences

The *t* test can also be used to find words whose cooccurrence patterns best distinguish between words. For example it can be used to find words that best differentiate the meaning of *strong*, w_1, and *powerful*, w_2. For each candidate word, v, we count the cooccurrences of it with w_1 and w_2, to create two independent normal populations. The *null hypothesis* is now that the average difference is 0 ($\mu = 0$), thus the numerator becomes:

$$\overline{x} - \mu = \overline{x} = \frac{1}{N}\sum_{i=1}^{N}(x_{1_i} - x_{2_i}) = \overline{x}_1 - \overline{x}_2 \tag{2.17}$$

For the denominator, we add the variances of the two populations, since the variance of the difference of two independent random variables is the sum of their individual variances.

$$t = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2.18}$$

Using the approximation: $s^2 = p(1-p) \approx p$:

$$\overline{x}_1 = s_1^2 = P(w_1 v), \quad \overline{x}_2 = s_2^2 = P(w_2 v) \tag{2.19}$$

Assuming that $C(x)$ is the count of occurrences of x in the corpus and N its size, t becomes:

$$t = \frac{P(w_1 v) - P(w_2 v)}{\sqrt{\frac{P(w_1 v) + P(w_2 v)}{N}}} = \frac{\frac{C(w_1 v)}{N} - \frac{C(w_2 v)}{N}}{\sqrt{\frac{C(w_1 v) + C(w_2 v)}{N^2}}} = \frac{C(w_1 v) - C(w_2 v)}{\sqrt{C(w_1 v) + C(w_2 v)}} \tag{2.20}$$

w	t	c_w	$c_{\text{strong w}}$	$c_{\text{powerful w}}$
computers	3.1622	933	0	10
computer	2.8284	2,337	0	8
symbol	2.4494	289	0	6
machines	2.4494	588	0	6
Germany	2.2360	2,266	0	5
⋮	⋮	⋮	⋮	⋮
support	7.0710	3,685	50	0
enough	6.3257	3,616	58	7
safety	4.6904	986	22	0
sales	4.5825	3,741	21	0
opposition	4.0249	1,093	19	1
⋮	⋮	⋮	⋮	⋮

Table 2.2: Words that occur significantly more often with *powerful* (the first five words) and *strong* (the last five words) in the corpus used in (Manning & Schutze 1999).

For example, table 2.2 shows words that occur the most often with *powerful* or *strong* in the corpus used in Manning & Schutze (1999). c_x denotes the occurrence frequency of x, while C_{xy} denotes the cooccurrence frequency of x and y together. Column t shows the computed values of statistic t of equation 2.20.

2.2.2.1.3.3 Pearson's chi-square test

This is an alternative test for dependence which does not assume normally distributed probabilities. In essence, it compares observed values with the expected ones for independence. If the difference between observed and expected frequencies is large, the null hypothesis of independence can be rejected. If O_{ij} and E_{ij} are the observed and expected values, relevant to the cell (i, j) of the table of frequencies, the quantity X^2 is defined as:

$$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{2.21}$$

It can be shown that the quantity X^2 is asymptotically χ^2 distributed. χ^2 values correspond to confidence levels. Pearson's chi-square test can be applied to any table of frequencies. For example, the following contingency table shows cooccurrence and non-

cooccurrence counts, in a text corpus of $N = 14,307,668$ tokens, to reflect the dependence of occurrences of w_1, *new*, and w_2, *companies*:

Observed values	$w_1 =$ new	$w_1 \neq$ new
$w_2 =$ companies	8 (new companies)	4,667 (e.g. old companies)
$w_2 \neq$ companies	15,820 (e.g. new machines)	14,287,173 (e.g. old machines)

Expected values are computed from the marginal probabilities:

$$E_{ij} = \frac{1}{N}\sum_{j=1}^{N} O_{ij} \times \frac{1}{N}\sum_{i=1}^{N} O_{ij} \times N = \frac{1}{N} \times \sum_{j=1}^{N} O_{ij} \times \sum_{i=1}^{N} O_{ij} \tag{2.22}$$

For example:

$$E_{11} = \frac{1}{N} \times (8 + 4,667) \times (8 + 15,820) \approx 5.2 \tag{2.23}$$

The expected values contingency table is:

Expected values	w_1	$\neg w_1$
w_2	5.2	4669.8
$\neg w_2$	15822.8	14287170.2

Thus, $X^2 \approx 1.55$, not greater than the critical value, 3.841, for a probability level of 5% and one degree of freedom for a 2×2 table, and the null hypothesis cannot be rejected.

In general, for collocation and term extraction, the differences between t and the χ^2 test do not seem to be large. However, χ^2 test can be applied to a wider range of problems, because it is also appropriate for large probabilities, for which the normality assumption of t test fails.

2.2.2.1.3.4 Log-likelihood ratios test

Log-likelihood ratios test (Brown et al. 1988; Dunning 1993) proposes a different approach to hypothesis testing, which seems to perform better than the χ^2 statistic when applied on sparse data. It also has the advantage that it is more interpretable than the χ^2 test; it shows how much more likely a hypothesis is than the other.

For *collocation and term extraction*, the usual hypotheses are the dependence and independence ones. For a bigram consisting of tokens w_1 and w_2, they are:

Hypothesis 1 (H_1)	Independence	$P(w_2\|w_1) = p = P(w_2\|\neg w_1)$
Hypothesis 2 (H_2)	Dependence	$P(w_2\|w_1) = p_1 \neq p_2 = P(w_2\|\neg w_1)$

where C_x is the count of occurrences of w_x in a corpus of size N and the probabilities can be estimated as:

$$p = \frac{C_2}{N}, \quad p_1 = \frac{C_{12}}{C_1}, \quad p_2 = \frac{C_2 - C_{12}}{N - C_1} \tag{2.24}$$

The process of randomly generating bigrams and assigning 1 if the bigram is $w_1 w_2$ or 0 otherwise is a *Bernoulli trial*, following the *binomial distribution*. The *binomial distribution* is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability x. The probability of getting exactly k successes is given by the probability mass function:

$$b\left(k;\, n, x\right) = \binom{n}{k} x^k \left(1 - x\right)^{(n-k)} \tag{2.25}$$

where:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{2.26}$$

The likelihoods of getting exactly the observed counts for w_1, w_2 and $w_1 w_2$, for hypotheses 1 and 2 are respectively:

$$L\left(H_1\right) = b\left(C_{12};\, C_1, p\right)\; b\left(C_2 - C_{12};\, N - C_1, p\right) \tag{2.27}$$

$$L\left(H_2\right) = b\left(C_{12};\, C_1, p_1\right)\; b\left(C_2 - C_{12};\, N - C_1, p_2\right) \tag{2.28}$$

The logarithm of the likelihood ratio λ is:

$$\begin{aligned} \log \lambda &= \log \frac{H_1}{H_2} \\ &= \log \frac{b\left(C_{12};\, C_1, p\right)\; b\left(C_2 - C_{12};\, N - C_1, p\right)}{b\left(C_{12};\, C_1, p_1\right)\; b\left(C_2 - C_{12};\, N - C_1, p_2\right)} \\ &= \log L\left(C_{12};\, C_1, p\right) + \log L\left(C_2 - C_{12};\, N - C_1, p\right) \\ &\quad - \log L\left(C_{12};\, C_1, p_1\right) - \log L\left(C_2 - C_{12};\, N - C_1, p_2\right) \end{aligned} \tag{2.29}$$

where:

$$L\left(k;\, n, x\right) = x^k \left(1 - x\right)^{(n-k)} \tag{2.30}$$

OT	w_1	$\neg w_1$
w_2	$n_{11} = 563$	$n_{12} = 702$
$\neg w_2$	$n_{21} = 1,085$	$n_{22} = 57,553$

Table 2.3: Observed values table (*OT*). Bigram: "gene expression"

ET	w_1	$\neg w_1$
w_2	$m_{11} = 35.44$	$m_{12} = 1,229.56$
$\neg w_2$	$m_{21} = 1,612.56$	$m_{22} = 55,940.44$

Table 2.4: Expected values table (*ET*). Bigram: "gene expression"

The log likelihood ratio λ can be used for hypothesis testing, since the quantity $-2\log\lambda$ is asymptotically χ^2 distributed. A high log-likelihood means that observed and expected values diverge significantly, indicating that the bigram constituents, w_1 and w_2, do not cooccur by chance. Contrarily, a log-likelihood close to 0 indicates that the bigram constituents, w_1 and w_2, cooccur by chance.

Equivalently, the log-likelihood ratio can be computed from contingency tables of observed and expected values, discussed in the previous section. For example, tables 2.3 and 2.4 are the tables of observed and expected values for the bigram "gene expression" occurring in *GENIA*.

Expected values are computed from the marginal probabilities of observed values. Assuming that n_{ij} is the i, j cell of the table of observed values, and m_{ij} is the i, j cell of the table of expected values, expected values can be computed as follows:

$$m_{ij} = \frac{\sum_{\forall k} n_{ik} \times \sum_{\forall k} n_{kj}}{\sum_{i,j} n_{ij}} \tag{2.31}$$

Finally, the log-likelihood ratio (λ) is computed as follows:

$$-2\log\lambda = 2 \times \sum_{i,j} n_{ij} \times \log\left(\frac{n_{ij}}{m_{ij}}\right) \tag{2.32}$$

For N-grams, where $N > 2$, there are more than one hypothesised models to compute expected values. For example, table 2.5 shows the different hypothesised models

Model 1	$\frac{P(w_1w_2w_3)}{P(w_1)\ P(w_2)\ P(w_3)}$
Model 2	$\frac{P(w_1w_2w_3)}{P(w_1w_2)\ P(w_3)}$
Model 3	$\frac{P(w_1w_2w_3)}{P(w_1)\ P(w_2w_3)}$
Model 4	$\frac{P(w_1w_2w_3)}{P(w_1w_3)\ P(w_2)}$

Table 2.5: Hypothesised models for trigrams

for trigrams. We use the extended log-likelihood ratios algorithm (McInnes 2004), in order to calculate log-likelihood ratios for each hypothesised model. For each model a different table of expected values is computed, while the observed values table remains the same for all. Then, the log-likelihood ratio corresponding to each model is calculated, as described in equation 2.32. The model with the lowest log-likelihood ratio is chosen. This model best represents the N-gram, since when a model is a good fit the observed values are close to the expected ones.

2.2.2.1.4 Pointwise Mutual Information

Pointwise Mutual Information (*PMI*) (Church & Hanks 1990) is a measure motivated by information theory and can be used for *collocation and term extraction*. If x' and y' are events, in our case the occurrence of particular tokens, *PMI* is defined as:

$$PMI\left(x', y'\right) = \log_2 \frac{P\left(x'y'\right)}{P\left(x'\right)\ P\left(y'\right)} = \log_2 \frac{P\left(x'|y'\right)}{P\left(x'\right)} = \log_2 \frac{P\left(y'|x'\right)}{P\left(y'\right)} \tag{2.33}$$

PMI can be extended to accommodate more than two tokens (McInnes 2004). The measure computes the amount of information increase we have about the occurrence of y' given that x' has occurred. For example, $PMI(w_1, w_2) = 20$ can have the following interpretations:

- Information about w_1 occurring at position i increases by 20 bits, knowing that w_2 occurs in position $i + 1$.
- Uncertainty is reduced by 20 bits.

#	Name	Formula
1.	Joint probability	$P(xy)$
2.	Conditional probability	$P(y\|x)$
3.	Reverse conditional probability	$P(x\|y)$
4.	Symmetric conditional probability	$\frac{P(xy)^2}{P(x*)P(*y)}$
5.	Selectional Association	$\frac{P(x\|y) \times PMI(x,y)}{P(*\|y) \times PMI(*y)}$
6.	Mutual dependency (MD)	$\log \frac{P(xy)^2}{P(x*)P(*y)}$
7.	Log frequency biased MD	$\log \frac{P(xy)^2}{P(x*)P(*y)} + \log P(xy)$
8.	Mutual dependency (MD)	$\log \frac{P(xy)^2}{P(x*)P(*y)}$
9.	Dice formula	$\frac{2f(xy)}{f(x)+f(y)}$
10.	Normalised expectation	$\frac{2f(xy)}{f(x*)f(*y)}$
11.	Mutual expectation	$\frac{2f(xy)}{f(x*)f(*y)} \cdot P(xy)$
12.	Salience	$\log \frac{P(xy)^2}{P(x*)P(*y)} \cdot \log f(xy)$

Table 2.6: Various other unithood measures

- We are much more certain that w_2 is the current word, if w_1 is the next one.

Unfortunately, decrease in uncertainty does not always correspond to an interesting relation between events x' and y'. Moreover, *PMI* does not perform well, when frequencies are low. None of the measures we have seen sofar performs very well for low frequency events, but there is evidence that data sparseness is a particularly difficult problem for *PMI*. Daille et al. (1994) have shown experimentally that the log likelihood ratio performs better than pointwise mutual information in the task of term recognition.

For N-grams of $N > 2$, there are more than one hypothesised models to compare against the joint distribution of N-gram constituents. The process is similar to the process followed in log-likelihood ratios. For each model different pointwise mutual information values are calculated, and the one with the lowest pointwise mutual inform-

ation value, i.e. the model which best represents the observed counts, is chosen. For example, the pointwise mutual information formula for the i^{th} 3-gram model of table 2.5 is $\log(\text{Model}_i)$.

Evert & Krenn (2001) have evaluated a handful of statistical measures in the task of extracting collocational support verb constructions in German. In particular, they evaluated pointwise mutual information, Dice coefficient, Pearson's chi-square, t test, log-likelihood and cooccurrence frequency. The authors report that t test achieves the best results, however none of the measures performed significantly better than cooccurrence frequency.

2.2.2.1.5 Other Unithood-based statistical measures

Except from the measures presented in the previous paragraphs, more unithood-based statistical measures can be used. Most of these are imported from other research fields and are included here for reasons of completeness. Pecina & Schlesinger (2006) present the largest collections of such measures in the literature. However, only a selection of these measures is evaluated or combined using various methods (e.g. support vector machines, neural networks, linear discriminant analysis etc.).

Let x and y be two words occurring in a corpus of N bigrams. $\bar{w}$ stands for any word except w; $f(w)$ for the frequency of word w and $*$ for any word. As defined in paragraph 2.2.2.1.3.3, the contingency table of observed frequencies for a bigram xy is:

O	x	$\bar{x}$	Sums
y	$a = f(xy)$	$b = f(x\bar{y})$	$f(x*)$
$\bar{y}$	$c = f(\bar{x}y)$	$d = f(\bar{x}\bar{y})$	$f(\bar{x}*)$
Sums	$f(*y)$	$f(*\bar{y})$	N

The table cells are sometimes referred to as f_{ij}. The corresponding table of expected values $\hat{f}(xy) = f(x*)f(*y)/N$ is:

E	x	$\bar{x}$
y	$\hat{f}(xy)$	$\hat{f}(x\bar{y})$
$\bar{y}$	$\hat{f}(\bar{x}y)$	$\hat{f}(\bar{x}\bar{y})$

#	Name	Formula
13.	Fisher's exact test	$\frac{f(x*)!f(\bar{x}*)!f(*y)!f(*\bar{y})!}{N!f(xy)!f(\bar{x}y)!f(x\bar{y})!f(\bar{x}\bar{y})!}$
14.	z score	$\frac{f(xy) - \hat{f}(xy)}{\sqrt{\hat{f}(xy)(1 - (\hat{f}(xy)/N))}}$
15.	Poison significance measure	$\frac{\hat{f}(xy) - f(xy)\log\hat{f}(xy) + \log f(xy)!}{\log N}$
16.	Squared log likelihood ratio	$-2\sum_{i,j}\frac{\log f_{ij}^2}{\hat{f}_{ij}}$

Table 2.7: Various other unithood measures.

Tables 2.6 and 2.7 presents a variety of unithood measures that include: estimation of joint and conditional bigram probabilities (1-4), mutual information and derived measures (5-11), statistical tests of independence (13-15), likelihood measures (16). Tables 2.8 and 2.9 presents various other heuristic association measures and coefficients.

2.2.2.2 Termhood-based approaches

Termhood-based methods focus on measuring how likely a candidate represents a specific concept, of general text or of some specific domain. These methods take into account nestedness information (Kageura & Umino 1996), i.e. they process frequencies of candidate multiword expressions and the frequencies of their substrings. In this review we present in detail C-value and NC-value (Maynard & Ananiadou 2000a; Frantzi et al. 2000), statistical barrier (Nakagawa 2000; Nakagawa & Mori 2002) and the method of Shimohata et al. (1997).

2.2.2.2.1 C-value

C-value (Maynard & Ananiadou 2000a; Frantzi et al. 2000) is a statistical measure of termhood for multiword expression recognition. The computation is based on information about occurrence of candidate terms as parts of other longer term candidates, i.e. nestedness information. The C-value measure comes together with a computationally efficient algorithm, which scores candidate multi-token terms according to the measure. It takes into account:

Name	Formula	Name	Formula
Hamann	$\frac{(a+d)-(b+c)}{a+b+c+d}$	Sokal-Michiner	$\frac{a+d}{a+b+c+d}$
Rogers-Tanimoto	$\frac{a+d}{a+2b+2c+d}$	Russel-Rao	$\frac{a}{a+b+c+d}$
S cost	$\log\left(1+\frac{\min(b,c)}{a+1}\right)^{-\frac{1}{2}}$	Jaccard	$\frac{a}{a+b+c}$
Baroni-Urbani	$\frac{a+\sqrt{ad}}{a+b+c+\sqrt{ad}}$	1st Kulczynsky	$\frac{a}{b+c}$
2nd Kulczynski	$\frac{1}{2}\left(\frac{a}{a+b}+\frac{a}{a+c}\right)$	Odds ratio	$\frac{ad}{bc}$
U cost	$\log\left(1+\frac{\min(b,c)+a}{\max(b,c)+a}\right)$	Yulle's Q	$\frac{ad-bc}{ad+bc}$
Confidence	$\max\left[P(y\|x),P(x\|y)\right]$	T combined cost	$\sqrt{U\times S\times R}$
Braun-Blanquet	$\frac{a}{\max(a+b,a+c)}$	2nd Sokal-Sneath	$\frac{a}{a+2(b+c)}$
Michael	$\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$	Mountford	$\frac{2a}{2bc+ab+ac}$
Simpson	$\frac{a}{\min(a+b,a+c)}$	3rd Sokal-Sneath	$\frac{b+c}{a+d}$
Driver-Kroeber	$\frac{a}{\sqrt{(a+b)(a+c)}}$	Yulle's ω	$\frac{\sqrt{ad}-\sqrt{bc}}{\sqrt{ad}+\sqrt{bc}}$
Piatersky-Shapiro	$P(xy)-P(x*)P(*y)$	Klosgen	$\sqrt{P(xy)}\cdot AV$
4th Sokal-Sneath	$\frac{1}{4}\left(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{d+b}+\frac{d}{d+c}\right)$		
5th Sokal-Sneath	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$		

Table 2.8: Various association coefficients

- the total frequency of occurrence of the candidate term in the corpus
- the frequency of the candidate term as part of longer candidate terms
- the number of these longer candidate terms
- the length of the candidate term (in number of tokens)

For example, consider the first two columns of table 2.10. The candidate string *basal cell carcinoma* appears 13 times in the corpus, but 9 of them are as part of a longer string. Thus, the frequency of candidate terms that appear as substrings of other

Name	Formula
Pearson	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$
Fager	$\frac{a}{\sqrt{(a+b)(a+c)}} - \frac{1}{2}\max(b,c)$
Unigram subtuples	$\log\frac{ad}{bc} - 3.29\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}$
R cost	$\log\left(1+\frac{a}{a+b}\right)\cdot\log\left(1+\frac{a}{a+c}\right)$
Phi	$\frac{P(xy)-P(x*)P(*y)}{P(x*)P(*y)(1-P(x*))(1-P(*y))}$
Kappa	$\frac{P(xy)+P(\bar{x}\bar{y})-P(x*)P(*y)-P(\bar{x}*)P(*\bar{y})}{1-P(x*)P(*y)-P(\bar{x}*)P(*\bar{y})}$
J measure	$\max[P(xy)\log\frac{P(y\|x)}{P(*y)}+P(x\bar{y})\log\frac{P(\bar{y}\|x)}{P(*\bar{y})},$ $P(xy)\log\frac{P(x\|y)}{P(x*)}+P(\bar{x}y)\log\frac{P(\bar{x}\|y)}{P(\bar{x}*)}]$
Gini index	$\max[P(x*)(P(y\|x)^2+P(\bar{y}\|x)^2)-P(*y)^2+$ $P(\bar{x}*)(P(y\|\bar{x})^2+P(\bar{y}\|\bar{x})^2)-P(*\bar{y})^2,$ $P(*y)(P(x\|y)^2+P(\bar{x}\|y)^2)-P(x*)^2+$ $P(*\bar{y})(P(x\|\bar{y})^2+P(\bar{x}\|\bar{y})^2)-P(\bar{x}*)^2]$
Laplace	$\max\left[\frac{NP(xy)+1}{NP(x*)+2},\frac{NP(xy)+1}{NP(*y)+2}\right]$
Conviction	$\max\left[\frac{P(x*)P(*y)}{P(x\bar{y})},\frac{P(\bar{x}*)P(*y)}{P(\bar{x}y)}\right]$
Certainty factor	$\max\left[\frac{P(y\|x)-P(*y)}{1-P(*y)},\frac{P(x\|y)-P(x*)}{1-P(x*)}\right]$
Added value (AV)	$max[P(y\|x)-P(*y),P(x\|y)-P(x*)]$
Collective strength	$\frac{P(xy)+P(\bar{x}\bar{y})}{P(x*)P(y)+P(\bar{x}*)P(*y)}\cdot\frac{1-P(x*)P(*y)-P(\bar{x}*)P(*y)}{1-P(xy)-P(\bar{x}\bar{y})}$

Table 2.9: Various association coefficients.

candidates should be decreased. However, the nested frequency of a given candidate term is not a reliable measure of its nestedness, since it does not take into account the number of different candidate terms, in which it might appear as nested. For example, consider the following terms in the domain of real time systems: *"real time clock"*, *"real time systems"*, *"real time group"* and *"real time expert system"*. The fact that they all

candidate string α	$f(\alpha)$	$\|\alpha\|$	$C\text{-}value(\alpha)$
W: *adenoid cystic basal cell carcinoma*	3	5	$3\ \log_2 5 \approx 6.97$
X: *cystic basal cell carcinoma*	1	4	$1\ \log_2 4 = 2$
Y: *ulcerated basal cell carcinoma*	5	4	$5\ \log_2 4 = 10$
Z: *basal cell carcinoma*	13	3	$10\ \log_2 3 \approx 15.85$

Table 2.10: C-value example

contain *"real time"* as substring, increases its possibility to be a term.

Consequently, the nestedness (NST) of a candidate term (ct) is defined as the fraction of its nested frequency over the number of distinct candidate terms, in which it appears as nested. Let T_{ct} be the set of candidate terms, in which the target candidate term (ct) appears as nested, $P(T_{ct})$ be the cardinality of T_{ct} and $f(x)$ denote the frequency of occurrence of x. Then, the nestedness (NST) of a candidate term (ct) is:

$$NST(ct) = \frac{1}{P(T_{ct})} \times \sum_{b \in T_{ct}} f(b) \tag{2.34}$$

Towards computing a *termhood* value, Frantzi et al. (2000) subtract the nestedness of the candidate term from its frequency of occurrence. The length of the candidate term in tokens, $|ct|$, is also taken into account. The longer the candidate term is, the more likely it is a valid term. The C-value score is defined as:

$$C\text{-}value(ct) = \begin{cases} \log_2(|ct|) \times [f(ct) - NST(ct)], & \text{if } ct \text{ is nested} \\ \log_2(|ct|) \times f(ct), & \text{otherwise} \end{cases} \tag{2.35}$$

In case that the candidate term (ct) appears as nested, C-Value is defined by the upper branch of equation 2.35. In the opposite case, i.e. that the candidate term never appears as nested, it is assigned a value based on its length and frequency of occurrence (lower branch of equation 2.35).

To return to the example of table 2.10, the C-values of the first three candidate terms can be computed using the first branch of formula 2.35. For the last one, the computation

is the following:

$$\begin{aligned} C\text{-}value(\text{basal cell carcinoma}) &= \log_2 3\ (13 - \frac{1}{3}(3+1+5)) \\ &= 10\ \log_2 3 \approx 15.85 \end{aligned} \tag{2.36}$$

The C-value algorithm starts with part of speech tagging the corpus and applying a predefined part of speech filter. Then, it extracts candidate terms that satisfy the part of speech filter, a stop-list and a frequency threshold, which depends on the working domain and on the type of terms that are accepted by the part of speech filter.

Subsequently, it computes the C-value of these candidate terms, starting from the longest terms. After computing the C-value of each term, it computes the C-value of its substrings. Finally, it filters out string with C-values lower than a predefined threshold and sorts candidates in decreasing C-value order.

2.2.2.2.2 NC-value

The NC-value measure (Maynard & Ananiadou 2000a; Frantzi et al. 2000) is an extension of C-value so as to take advantage of contextual information, by assigning weights to context words: *nouns*, *verbs* and *adjectives*. The NC-value algorithm takes as input the candidate term list together with the C-value of each candidate and outputs a re-ranked version of the same list. The final ordering is expected to include actual terms in higher positions and thus improve *term recognition* results. The NC-value measure takes into account:

- the number of terms a context word appears in
- its frequency as a context word
- its total frequency in the corpus
- its length

Initially, the NC-value algorithm creates a set of context words, Cw, by applying a predefined window on the left and right context of the n highest ranked term candidates, according to the C-value algorithm. The parameter n is defined beforehand, depending on the desired computational complexity of the resulting algorithm. Then, it filters out

Part of speech	Context words
Adjectives	*pathologic* [W(2), Z(3)], *clinical* [W(2), Z(2)], *cutaneous* [X(1), Z(2)], *micronodular* [Z(3)], *epithelial* [W(1), Z(1)], *extraprostatic* [W(1), Z(1)], *immunohistochemical* [W(1), Z(1)], *metastatic* [W(1), Z(1)], *nodular* [Y(1), Z(1)], *odontogenic* [W(1), Z(1)], *surgical* [W(1), Z(1)]
Nouns	*diagnosis* [W(2), Y(1), Z(2)], *neoplasm* [W(2), Z(2)], *prostate* [W(2), Z(2)], *report* [W(2), Z(2)], *tumor* [W(2), Z(2)], *cancer* [Z(3)], *condition* [X(1), Z(2)], *basal-cell* [Z(2)], *extension* [W(1), Z(1)], *patient* [W(1), Z(1)], *skin* [Z(2)], *specimen* [W(1), Z(1)], *treatment* [W(1), Z(1)], *biopsy* [Z(1)], *radiotherapy* [Y(1)]
Verbs	*include* [W(1), Z(2)], *base* [Y(1), Z(1)], *characterize* [X(1), Z(1)], *roll* [Y(2)], *warrant* [W(1), Z(1)]

Table 2.11: Context words of the term candidates in table 2.10. The capital letters within square brackets denote which candidate term the preceding context word occurs with and the frequency of occurrence is within parenthesis.

Cw words other than nouns, verbs and adjectives and uses the following measure, which assigns high weights to words that tend to appear with terms more frequently:

$$weight(w) = \frac{t(w)}{n} \tag{2.37}$$

where $w \in Cw$, $t(w)$ is the number of terms the word w appears with, and n is the total number of terms considered. In the case that a context word (Cw) of a candidate term was not encountered a context word of the top n candidate terms of the C-Value output list, it is assigned a zero weight.

For example, table 2.11 shows the context words for the term candidates in table 2.10 hypothesizing that $n = 4$, i.e. the context of all four candidate terms is taken into account[1]. Context words that occur with one, two or three candidates are assigned the weights: $\frac{1}{4}$, $\frac{2}{4}$ or $\frac{3}{4}$, respectively.

[1]For brevity, context word in table 2.11 are a subset of the context words of the term candidate occurences, collected from the web.

Subsequently, for each candidate term, $ct = w_1, w_2, \cdots, w_k$, the algorithm computes $C(ct)$, the set of distinct context words of α, which appear in Cw. The *NC-value* corresponding to each candidate term, ct, is then computed as the linear interpolation of C-Value and the context factor, as follows:

$$NC\text{-}value(ct) = 0.8 \times C\text{-}value(ct) + 0.2 \times \sum_{b \in C_{ct}} f_{ct}(b)\ weight(b) \tag{2.38}$$

where b is a word in C_{ct} and $f_{ct}(b)$ is the frequency of b as a context word of α. NC-value outputs the list of the same candidate terms as C-Value, sorted in decreasing order of NC-value score.

For example, the *NC-value* of candidate term Y, *ulcerated basal cell carcinoma* in table 2.10, can be computed taking into account its context words: *nodular*, *diagnosis*, *radiotherapy*, *base* and *roll* (table 2.11):

$$\begin{aligned}
NC\text{-}value(\mathrm{Y}) &= 0.8 \times C\text{-}value(\mathrm{Y}) + 0.2 \times \sum_{b \in C_{\mathbf{Y}}} f_{\mathbf{Y}}(b)\ weight(b) \\
&= 0.8 \times 10 + 0.2\{f_{\mathbf{Y}}(\textit{nodular}) \times weight(\textit{nodular}) + \\
&\qquad f_{\mathbf{Y}}(\textit{diagnosis}) \times weight(\textit{diagnosis}) + \\
&\qquad f_{\mathbf{Y}}(\textit{radiotherapy}) \times weight(\textit{radiotherapy}) + \\
&\qquad f_{\mathbf{Y}}(\textit{base}) \times weight(\textit{base}) + \\
&\qquad f_{\mathbf{Y}}(\textit{roll}) \times weight(\textit{roll})\} \\
&= 8 + 0.2\{1 \times \frac{2}{4} + 1 \times \frac{3}{4} + 1 \times \frac{1}{4} + 1 \times \frac{2}{4} + 2 \times \frac{1}{4}\} \\
&= 8 + 0.2 \times 2.5 \\
&= 8.5
\end{aligned}$$

Likewise, the *NC-value* of terms W, X and Z in table 2.10 is computed as 6.875, 1.75 and 15.03, respectively.

2.2.2.2.3 Statistical barrier

Statistical Barrier (SB) (Nakagawa 2000; Nakagawa & Mori 2002) similarly to C-value and NC-value assumes that successful multiword candidates that have complex structure are made of existing simpler terms. Thus, they first measure the *termhood* of single

words, and then use it to measure the *termhood* of complex terms. The basic intuition is that if a single word, N, expresses a key concept of a domain that a document treats, then the author must be using N, not only frequently, but also in various ways. Thus, there will be a number of valid terms containing N.

The basic intuition behind the method is that if a single word expresses a concept of a domain, then it occurs not only frequently, but also in various ways. It is expected that a number of valid terms contain this single word. This potential relationship between single words and multiword candidate terms is exploited to automatically recognise terms.

In particular, after part of speech tagging a given corpus, Nakagawa (2000) extracts a list of single words. Let $R(N)$ and $S(N)$ be two functions that calculate the number of distinct words that adjoin N or N adjoins, respectively. Then, for each candidate term, $ct = N_1, N_2, \ldots, N_k$ a score based on the geometric mean is calculated:

$$GM(ct) = \sqrt[2k]{\prod_{i=1}^{k}[(R(N_i)+1) \times (S(N_i)+1)]} \tag{2.39}$$

For example, suppose the candidate term "*basal cell carcinoma*" (Z) and the statistics shown in table 2.12. In some corpus, there are 20 distinct words occurring before "*basal*" and 10 distinct words occorring after it. Thus, for this example, formula 2.39 can be computed as:

$$\begin{aligned}
GM(Z) &= \sqrt[2\times 3]{\prod_{i=1}^{3}[(R(N_i)+1) \times (S(N_i)+1)]} \\
&= \sqrt[6]{(20+1)\times(10+1)+(8+1)\times(3+1)+(4+1)\times(12+1)} \\
&= \sqrt[6]{21\times 11+9\times 4+5\times 13} \\
&= \sqrt[6]{231+36+65} \\
&= \sqrt[6]{332} \\
&\approx 2.6315
\end{aligned}$$

Nakagawa (2000) notes that the frequency of independent occurrences of candidate terms have a significant impact on the term recognition process. Independent occurrences are the ones, where the candidate term ct, is not nested to any other candidate

candidate token N	$R(N)$	$S(N)$
basal	20	10
cell	8	3
carcinoma	4	12

Table 2.12: Numbers of distinct words that precede ($R(N)$) or follow ($S(N)$) the tokens of the candidate term *basal cell carcinoma*.

term. To incorporate this, GM is multiplied by the *marginal frequency*, $MF(ct)$, i.e. the number of independent occurrences of ct:

$$SB(ct) = GM(ct) \times MF(ct) \tag{2.40}$$

For example, supposing that the candidate term "*basal cell carcinoma*" (Z) occurs independently 10 times, its statistical barrier score is:

$$SB(Z) = GM(Z) \times MF(Z) = 2.6315 \times 10 = 26.315$$

Statistical barrier outputs a list of candidates sorted in decreasing order of SB score.

2.2.2.2.4 Method of Shimohata et al. (1997)

Shimohata et al. (1997) propose another termhood-based method for recognising multiword expressions, especially domain specific ones. The method consists of two steps. Firstly, it identifies sequences that are highly possible to be either multiword expressions or components of multiword expressions. In succession, it takes into account nestedness information to create multiword expressions by joining output candidates of the first step that cooccur frequently, or by deleting candidates that are subsumed by others.

In the first step of the method, out of all N-grams of the input text, only the meaningful ones are kept by applying an entropy threshold. For every N-gram the method calculates the distribution of adjacent words preceding and following it. This is based on the idea that adjacent words will be widely distributed if the N-gram is meaningful, and they will be restricted if the N-gram is a substring of a meaningful string. The probability

$p(w_i)$ of each possible adjacent word w_i of an N-gram x is:

$$p(w_i) = \frac{f(w_i, x)}{f(x)} \tag{2.41}$$

where $f(x)$ is the frequency of N-gram x and $f(w_i, x)$ is the frequency that w_i precedes or follows x. The entropy of $H(x)$ of N-gram x is defined as:

$$H(x) = -\sum_{i=1}^{n} p(w_i) \log p(w_i) \tag{2.42}$$

after calculating the entropy of both sides of the N-gram, the lowest is adopted as the entropy of the N-gram and is thresholded by a tunable parameter T_{entropy}:

$$\min\left(H_{\text{left}}(x), H_{\text{right}}(x)\right) > T_{\text{entropy}} \tag{2.43}$$

The maximum value of $H(x)$ is $\log n$ and is taken if $f(w_i) = 1,\ \forall i$. The minimum value of $H(x)$ and is taken if x occurs once ($f(x) = 1$). All N-grams that do not satisfy the previous inequality are discarded, in an attempt to keep meaningful units such as compound words, prepositional phrases, and idiomatic expressions.

In the second stage, the method joins frequently cooccurring N-grams, based on the idea that there is some N-gram that is often used to introduce a multiword expression. This introducing N-gram is called key. The method works as follows:

1. For every key k from the N-grams $w_i, i \in [1, n]$ retrieve sentences of the corpus containing k.

2. Keep w_i if the frequency of cooccurrence of k and w_i exceeds a predefined threshold T_{freq}.

3. Examine every two N-grams x and y pairwise:

   - Join x and y if:
   $$\frac{freq(x, y)}{freq(x)} \geq T_{\text{ratio}} \tag{2.44}$$
   - Discard x if y subsumes it and:
   $$\frac{freq(y)}{freq(x)} \geq T_{\text{ratio}} \tag{2.45}$$

4. Arrange accepted N-grams according to their order in the corpus.

Shimohata et al. (1997) illustrate with an example how the second stage works. "Refer to" is identified as key for the underlined N-grams of the following instances:

Refer to the appropriate manual for instructions on ...
Refer to the manual for specific instructions.
Refer to the installation manual for specific instructions for ...
Refer to the manual for specific instructions on ...

Assuming $T_{\text{freq}} = 2$, the strings which cooccur with the key, k, more than twice are extracted in the second step:

x	$freq(k, x)$
the	4
manual	4
for specific instructions	3
on	2

Based on the intuition that longer N-grams are more significant if they are frequent enough, the third step joins such N-grams. This process is repeated until no N-gram satisfies the inequalities. Assuming $T_{\text{ratio}} = 0.75$, "manual" and "for specific instructions" are joined together and subsequently removed from the list.

z	$freq(k, x)$
the	4
manual for specific instructions	3
on	2

The fourth step arranges the N-grams of the list to construct the final multiword expression: "Refer to the ... manual for specific instructions on ...", where "..." indicate gaps that can be filled with any word.

2.2.3 Hybrid approaches

Hybrid approaches consist of linguistic and statistical components and possibly of other components such as supervised and unsupervised classifiers (e.g. hidden Markov models, support vector machines, decision trees, naive Bayes classifiers).

In practice, most of the statistical approaches of section 2.2.2 use various levels of linguistic preprocessing. Tokenisation (subsection 2.2.1.1) is used by all statistical methods and its performance clearly affects their results since statistical approaches are based on frequency counts. Cooccurrence frequency counts (paragraph 2.2.2.1.1) can be computed on tokenised text, but without any restriction to the length or the parts of speech of the desired sequences the computation can become impractically demanding for large corpora. Pointwise Mutual Information (paragraph 2.2.2.1.4) suffers from extremely frequent function words, and is reported to perform better using a stoplist.

C-Value (paragraph 2.2.2.2.1), NC-Value (paragraph 2.2.2.2.2) and statistical barrier (paragraph 2.2.2.2.3) methods take as input multiword expression candidates identified by part of speech patterns (subsection 2.2.1.5) which in turn hypothesise part of speech tagging (subsection 2.2.1.2). The method of Shimohata et al. (1997) (paragraph 2.2.2.2.4) does not need any linguistic preprocessing other than tokenisation, but probably lemmatisation (subsection 2.2.1.3) would improve its results.

In this subsection, we will present a number of more complicated approaches in the literature, which we call hybrid. Some of them identify multiword expression by context and semantic information, e.g. SNC-value (Maynard & Ananiadou 2000b), while others, e.g. Bannard (2007), by quantifying various properties of multiword expressions such as syntactic flexibility. The tasks for which these methods were developed range from identifying multiword expressions (Van de Cruys & Moirón 2007) to structuring data for databases (Schulte im Walde 2003).

Hybrid methods for identifying various types of multiword expressions also appear as components of bigger systems. For example, Feldman et al. (1998) propose a system to structure domain-specific text documents for databases. Their system among others contains a term generation stage where sequences of tagged lemmas are selected as potential term candidates. For this purpose they employ part of speech patterns and association measures, such as cooccurrence frequency, pointwise mutual information and log-likelihood test. Furthermore, Subramaniam et al. (2003) attempt to perform information extraction from biomedical articles. Their term extraction component applies pattern rules based on stopwords and keywords at the shallow parse of input texts. Then terms are recognised using machine readable dictionaries.

2.2.3.1 SNC-value

TRUCKS (Maynard & Ananiadou 2000b,a) is a multiword expression identification system that uses C-value and NC-value algorithms (discussed in paragraphs 2.2.2.2.1 and 2.2.2.2.2) as its first two layers and adds a third called Importance Weight (IW). Importance Weight incorporates three types of contextual information: (a) syntactic (b) terminological and (c) semantic knowledge.

Syntactic knowledge is based on identifying boundary words, i.e. words which occur immediately before or after a candidate term. Boundary words were also incorporated in LEXTER (section 2.2.1.7). Based on the idea that particular syntactic categories are used to delimit candidate terms, boundary words are weighted differently, according to their category. Experiments computing the relative likelihood of each syntactic category occurring in the context of a multiword expression or not were conducted to compute the syntactic weights, $syn(d)$, where d is a context term. Results indicated that verbs should be weighted as important (1.2), followed by preposition (1.1) nouns (9) and finally adjectives (0.7).

Terminological knowledge concerns the terminological status of context words, i.e. whether or not the context contains domain-specific multiword expressions that have been recognised already. As context terms are considered candidate terms that were highly ranked by NC-value, in the top third of the list. For each candidate, a context term (CT) weight is computed based on its frequency of cooccurrence with other context terms:

$$CT(\alpha) = \sum_{d \in T_\alpha} f_\alpha(d) \tag{2.46}$$

where α is a candidate multiword expression, T_α is the set of context terms of α, d is a word from T_α, $f_\alpha(d)$ is the frequency of d as a context term of α.

Semantic knowledge about context terms is obtained using the UMLS Semantic Network, provided by U.S. National Library of Medicine (NLM). The similarity between a candidate and its context terms is computed based on the distance between them in the Semantic Network hierarchy. The computation depends on the vertical position and horizontal distance between the two term nodes n_1 and n_2. The closer they are, the larger the similarity value. Let $depth(n_1)$ and $depth(n_2)$ be the distances of nodes n_1 and n_2 from the root respectively; and c the number of common ancestors. The similarity of

two terms t_1 and t_2 is defined as:

$$sim(t_1, t_2) = \frac{2 \times c}{depth(n_1) + depth(n_2)} \tag{2.47}$$

The importance weight incorporates the syntactic, terminological and semantic weights:

$$IW(\alpha) = \sum_{b \in C_\alpha} syn(b) + \sum_{d \in T_\alpha} f_\alpha(d) sim(\alpha, d) \tag{2.48}$$

where α is the candidate term, C_α is the set of context words of α, $syn(b)$ is the syntactic weight of b, T_α is the set of context terms of α, $f_\alpha(d)$ is the frequency of d as a context term of α, and $sim(\alpha, d)$ is the similarity weight of d as a context term of α. The final SNC-Value ranking is computed by adding the Importance Weight with the NC-Value:

$$SNC\text{-}Value(\alpha) = NC\text{-}Value(\alpha) + IW(\alpha) \tag{2.49}$$

where α is the candidate term, $NC\text{-}Value(\alpha)$ is the $NC\text{-}Value$ of α and $IW(\alpha)$ is the importance weight of α.

2.2.3.2 Combining extractors with Latent Semantic Analysis

Schone & Jurafsky (2001) evaluated a variety of unithood-based collocation extraction approaches: frequency, pointwise mutual information (PMI), selectional association, symmetric conditional probability, Dice formula, log-likelihood ratios, Pearson's chi-square, z score and t test. They showed that information-like approaches, particularly z score, symmetric conditional probability, and chi-square perform better than the others but in general results are very low. They also proposed two new approaches, based on *Latent Semantic Analysis (LSA)*, which combine the three most successful approaches in an attempt to take advantage of non-compositionality and non-substitutability. They conclude that LSA improved results, but very little compared to the effort required to obtain the two LSA models.

2.2.3.3 Combining extractors with Adaptive Boosting

Vivaldi et al. (2001) used four extractors for domain specific terms: (a) a semantic-based, (b) a Greek and Latin form analyser, (c) a context-based, and (d) a collocational extractor. The semantic-based extractor exploits the idea that terms are made of other

v	det:*NULL*	n_{sg}	v	det:*NULL*	n_{pl}
v	det:*a/an*	n_{sg}			
v	det:*the*	n_{sg}	v	det:*the*	n_{pl}
v	det:*DEM*	n_{sg}	v	det:*DEM*	n_{pl}
v	det:*POSS*	n_{sg}	v	det:*POSS*	n_{pl}
v	det:*OTHER*	$[n_{sg,pl}]$	det:*ANY*	$[n_{sg,pl}]$	be v_{pas}

Table 2.13: P: Patterns for recognising canonical forms. sg stands for singular number, pl for plural, v for verb, n for nouns, pas for passive and *det* for determiner.

domain-specific terms. The extractor uses EuroWordNet to determine whether the term candidate itself and its components belong to their chosen domain (medical) or not.

The Greek and Latin form analyser, splits candidates to their Greek and Latin components, if they are made of such. Then it obtains the meanings of the components from lexica and scores candidates, accordingly. The context based extractor uses SNC-value (Maynard & Ananiadou 2000b,a) (section 2.2.3.1) with very small modifications. Finally, the collocational extractor uses unithood-based approaches, such as log-likelihood ratios, mutual information, and cubed mutual information, MI^3.

The extractors were evaluated separately, and then combined using simple voting schemes or Adaptive Boosting. Results showed that using AdaBoost in the meta-learning step, the ensemble constructed surpasses the performance of all individual extractors and simple voting schemes, obtaining significantly better recall.

2.2.3.4 Canonical Forms

Fazly & Stevenson (2006) hypothesise that semantic idiosyncrasy is reflected in lexical and/or syntactic behaviour. They also claim that lexical and syntactic flexibility can be explained in terms of decomposability, which in turn is inversely related to idiomaticity. For example, the highly idiomatic expression *"shoot the breeze"* is non-decomposable, while the less idiomatic expression *"spill the beans"* is analysable with *"spill"* corresponding to "reveal" and *"beans"* corresponding to "secret(s)".

Fazly & Stevenson (2006) propose a set of patterns that can be used to recognise different syntactic forms of a given noun-verb construction (NVC); i.e. different canonical forms of an NVC. They are proposed as a product of theoretic linguistic knowledge and experimental results and are shown in table 2.13. The patterns mix parts of speech and

Class	Example	I	T	IN	IN	NC
LIT	They shot a bird.	-	+	-	-	-
ABS	They make a living singing.	+	- -	+	+	++
LVC	They gave the lasagna a try.	++	-	+	++	+
IDM	they shot the breeze.	+++	- - -	+	++	++

Table 2.14: Categorisation of NVCs in Fazly & Stevenson (2007).

specifications for various properties of the component words, such as number.

Fazly & Stevenson (2006) claim that idiomatic *NVCs* can be represented in a lexicon using their one or more canonical forms. Using the predefined set of syntactical patterns of table 2.13 they count the frequency, $f(v, n, pt_k)$, of each *NVC*, $\langle v, n \rangle$, occurring in each one, $pt_k \in P$. Then, they calculate the z-score for each $\langle v, n \rangle$ and each pattern pt_k:

$$z_k(v, n) = \frac{f(v, n, pt_k) - \overline{f}}{s} \tag{2.50}$$

where $\overline{f}$ is the mean of frequencies and s the standard deviation.

2.2.3.5 Distinguishing Subtypes of NVCs (Fazly & Stevenson 2007)

Fazly & Stevenson (2007) propose statistical measures that can identify several linguistic features of multiword expressions in English. They focus on multiword expressions that consist of a verb and a noun, i.e. noun-verb combinations (NVCs). NVCs are classified into four classes: *literal constructions (LITs)*, *abstract constructions (ABSs)*, *light Verb constructions (LVCs)*, and *idiomatic expressions (IDMs)*. The classification is based on the levels of the following properties that characterise multiword expressions and were discussed in section 1.1: idiosyncrasy (I), translation (T), institutionalisation (IN), lexico-syntactic fixedness (LF), and non-compositionality (NC). The proposed statistical measures are used as features to train a decision tree induction system *C5.0*.

Table 2.14 presents four categories of NVCs: literal constructions (LITs), abstract constructions (ABSs), light Verb constructions (LVCs), and idiomatic expressions (IDMs). LVCs are multiword expressions because usually their overall meaning diverges from the combined meanings of the constituents. In ABSs the meaning of the verbs is metaphorical and the basic physical semantics are extended. For each class, table 2.14 shows an example and an indication of the level of the above properties, ranging from "- - -",

which means that the property never occurs, to "+++", which means that the property occurs always.

Fazly & Stevenson (2007) argue that the above properties are not sufficient to determine the class of a given NVC. Semantic properties of phrasal constituents, selectional preferences of verbs and semantic category of nouns are specified in the parser output and can be used as classification features.

Supposing that $f(v,n)$ is the raw frequency of the verb-object pair (v,n), and $*$ represents a summation over all verbs or nouns that occur in the candidate multiword expressions list, the proposed measure of *institutionalisation* is computed as the pointwise mutual information (PMI) of (v,n):

$$PMI(v,n) = \log \frac{P(v,n)}{P(v)\ P(n)} \approx \frac{f(*,*)\ f(v,n)}{f(v,*)\ f(*,n)} \tag{2.51}$$

Separate measures are proposed for three kinds of *fixedness*: *lexical*, *syntactical* and *overall*. *Lexical fixedness* is computed as:

$$Fixedness_{lex}(v,n) = \frac{PMI(v,n) - \overline{PMI}}{std} \tag{2.52}$$

where $\overline{PMI}$ is the mean and std the standard deviation over the PMI of the target and all its variants. *Syntactical fixedness* is defined as:

$$\begin{aligned} Fixedness_{syn}(v,n) &= D(P(pt|v,n)\|P(pt)) \\ &= \sum_{pt_k \in \mathcal{P}} P(pt_k|v,n) \log \frac{P(pt_k|v,n)}{P(pt_k)} \end{aligned} \tag{2.53}$$

where $\mathcal{P}$ is the set of patterns of table 2.13, $P(pt_k|v,n)$ represents the syntactic behaviour of the target and $P(pt_k)$ represents the typical syntactic behaviour over all verb-object pairs. The dominant pattern can be determined by the following equation:

$$Pattern_{dom}(v,n) = \underset{pt_k \in \mathcal{P}}{\text{argmax}}\ f(v,n,pt_k) \tag{2.54}$$

Overall fixedness is a linear combination of *lexical* and *syntactical fixedness*:

$$Fixedness_{overall}(v,n) = \alpha Fixedness_{syn}(v,n) + (1-\alpha) Fixedness_{lex}(v,n) \tag{2.55}$$

$Fixedness_{adj}$ quantifies the degree of fixedness of the target (v, n) combination with respect to adjectival modification of the noun constituent:

$$Fixedness_{adj}(v, n) = D(P(a|v, n)\|P(a)) \tag{2.56}$$

where a is a pattern which marks the presence or absence of an adjectival modifier preceding the noun.

Fazly & Stevenson (2007) include in their feature set the verb itself and the semantic category of the noun according to WordNet; the ancestor of its first sense. Compositionality can be measured by comparing the context of each multiword expression candidate, $t = \langle v, n \rangle$, to the context of its constituents, n and v (see section 2.3). The context of the target is a vector holding the frequencies of nouns cooccurring within a window ± 5. Based on the cosine similarity metric (see equation 2.72), they use as features the measures: $\cos(t, v)$, $\cos(t, n)$ and $\cos(t, rv)$, where rv is a verb morphologically related to n, extracted from WordNet.

2.2.3.6 A Measure of Syntactic Flexibility of NVCs (Bannard 2007)

Bannard (2007) presents a measure of syntactic fixedness through which English NVCs can be identified in corpora. The measure captures variation of multiword expression candidates and is proposed as a tool for lexicographers. It is applied on the output of a syntactic parser, RASP (Briscoe et al. 2006). The system is evaluated using dictionary-published multiword expressions.

Three types of non-morphological variation are identified:

- Variation by addition or dropping of a determiner, e.g. *run the show*, *run their show*
- Internal phrase modification, e.g. *break the ice*, *break the diplomatic ice*
- Passivisation, e.g. *call the shots*, *the shots were called*

Each kind of variation is associated with one component word. Passivisation is associated with the verb. Internal modification and determiner variation are associated with the noun in object position.

Syntactic fixedness is interpreted as the extent to which the probability of variation of an *NVC*, (v, n), deviates from the probability of variation of its components, v and

n. Assuming that x denotes syntactic variation, syntactic fixedness is measured as the conditional pointwise mutual information of the component y given the word z:

$$\begin{aligned} PMI(x;y|z) &= H(x|z) - H(x|y,z) \\ &= -\log_2 p(x|z) - [-\log_2 p(x|y,z)] = log_2 \frac{p(x|y,z)}{p(x|z)} \end{aligned} \quad (2.57)$$

In the case of passivisation $z \equiv noun$ and $y \equiv verb$, whereas in the cases of determiner variation and internal modification $z \equiv verb$ and $y \equiv noun$. Overall syntactic variation is computed as the sum of variations for the above categories:

$$\begin{aligned} SynVar(W) &= \sum_{n}^{i} PMI(VerbVar_i;\ Obj|Verb) + \\ &\quad + \sum_{n}^{i} PMI(ObjVar_i;\ Verb|Obj) \end{aligned} \quad (2.58)$$

2.2.3.7 Semantics-based NVC Extraction (Van de Cruys & Moirón 2007)

Van de Cruys & Moirón (2007) propose a fully unsupervised method for large-scale NVC extraction. They hypothesise that non-compositionality can be quantified by measuring non-substitutability. For example, consider *break the* {*vase, cup, dish*} versus *break the* {*ice, ?snow, ?hail*}. Their approach is evaluated on automatically extracted data from Dutch Newswire corpora, using lexical resources. It uses dependency parsing, clustering of semantically related nouns and statistical measures, to quantify selectional preferences.

Noun clustering is realised by distributional similarity measures (see section 2.3), in the following way: dependency triples are extracted and for each noun a feature vector with the frequencies of the dependency relations, in which it participates, is created. Then, frequency values are replaced by PMI scores. Finally, clusters are created using a simple K-means clustering algorithm with cosine similarity.

The *Kullback-Leibler divergence* between the prior probability of a noun $p(n)$ and the probability of the noun given a verb $p(n|v)$ is:

$$S_v = \sum_n p(n|v)\ \log \frac{p(n|v)}{p(n)} \quad (2.59)$$

S_v is used as a normalisation constant in most of the statistical measures, proposed by Van de Cruys & Moirón (2007). The formula:

$$A_{v \to n} = \frac{p(n|v)\ \log \frac{p(n|v)}{p(n)}}{S_v} \tag{2.60}$$

measures the unique preference of a verb to a noun. The ratio of verb preference for a particular noun, compared to other nouns of the same semantic cluster, C, is given by:

$$R_{v \to n} = \frac{A_{v \to n}}{\sum_{n' \in C} A_{v \to n'}} \tag{2.61}$$

The *Kullback-Leibler divergence* between the prior probability of the verbs $p(v)$ and the probability of the verbs given a noun $p(v|n)$ is:

$$S_n = \sum_v p(v|n)\ \log \frac{p(v|n)}{p(v)} \tag{2.62}$$

The unique preference of a noun to a verb is:

$$A_{n \to v} = \frac{p(v|n)\ \log \frac{p(v|n)}{p(v)}}{S_n} \tag{2.63}$$

The ratio of noun preference for a particular verb, compared to other nouns of the same semantic cluster, C, is given by:

$$R_{v \to n} = \frac{A_{n \to v}}{\sum_{n' \in C} A_{n' \to v}} \tag{2.64}$$

2.2.3.8 Identifying NVCs in Token Context (Cook et al. 2007)

Cook et al. (2007) address the task of identifying idiomatic NVCs on a per-instance basis. Therefore the type of context they choose is token context (see 2.2.1.6). Their approach uses informative prior knowledge about the overall syntactic behaviour of an NVC (e.g. type context), syntactic fixedness measures and contextual information.

The basic idea is that idiomatic NVCs tend to be fixed with respect to their canonical forms (CF), the syntactic configurations in which they occur. Table 2.15 shows two *NVCs*

Idiom	**Canonical forms**
(pull, weight)	pull one's weight
(hold, fire)	hold fire, hold one's fire

Table 2.15: Idioms and canonical forms example.

together with the canonical forms in which they mainly appear when used idiomatically.

The preferred patterns can vary across different idiom types, and can involve a number of syntactic properties: the voice of the verb, the determiner introducing the noun, the number of the noun, etc.

Cook et al. (2007) assume that in most cases, idiomatic usages of a *NVC* tend to occur in a small number of canonical forms for that idiom, while literal usages of it are less syntactically restricted. Usages in syntactic patterns other than the canonical forms are hypothesised to be literal.

Cook et al. (2007) use the automatically determined canonical forms in Fazly & Stevenson (2006) (subsection 2.2.3.4) to compute the measures proposed in three different settings: CForm, Diff$_{I\text{-}CF,L\text{-}NCF}$ and Diff$_{I\text{-}CF,L\text{-}Comp}$. CForm uses knowledge of canonical forms only, and classifies an *NVC* instance as idiomatic if it satisfies one of the canonical forms, or as literal otherwise. The latter methods assume the distributional hypothesis and are based on the computation of the following cooccurrence vectors:

- $\vec{v}_e$: frequencies of the words cooccurring with an *NVC* e.
- $\vec{v}_t$: frequencies of the words cooccurring with a single token t, part of some *NVCs*.
- $\vec{v}_{I\text{-}CF}$: cooccurrence vector for the uses of a *NVC* in its *CFs*, thus idiomatic.
- $\vec{v}_{L\text{-}NCF}$: cooccurrence vector for the uses of a *NVC* in its *non-CFs*, thus literal.
- $\vec{v}_{L\text{-}Comp}$: the result of summing and normalising vectors $\vec{v}_t$, of the *NVC*'s components.

Diff$_{I\text{-}CF,L\text{-}NCF}$ compares $cos(\vec{v}_t, \vec{v}_{I\text{-}CF})$ with $cos(\vec{v}_t, \vec{v}_{L\text{-}NCF})$. In words, it quantifies the distance between the similarity of the *NVC* with its components, in literal and idiomatic cases. *Diff*$_{I\text{-}CF,L\text{-}Comp}$ compares $cos(\vec{v}_t, \vec{v}_{I\text{-}CF})$ with $cos(\vec{v}_t, \vec{v}_{L\text{-}Comp})$, assuming that the literal uses of an *NVC* can be inferred by the uses of its components.

Experiments showed that *CForm* performs best, comparably to a supervised baseline approach.

2.2.3.9 An NVC Database

Schulte im Walde (2003) proposes a method to create an NVC database in German. The method efficiently stores verbs according to their selectional preferences. For nouns, the database stores their verbal selectional preferences and the most frequently cooccurring adjectival and genitive noun phrase modifiers.

NVCs are located using a statistical grammar model, based on the framework of head-lexicalised context free grammars. It computes the frequencies for any two lexical items l_1 and l_2 cooccurring within a relationship r: $f(l_1, r, l_2)$. The collocation strength is based on the probabilistic cooccurrence counts and is determined by the lexical association measure log-likelihood. The system is trained on a German newspaper corpus.

2.2.4 Discussion

In this section we presented in detail many widely used linguistic and statistical components for the task of multiword expression recognition. In parallel, we presented a large variety of complete systems that address the task, taking into account various sources of information. Linguistic systems consider the linguistic properties of text to decide which sequences should be candidate multiword expressions. Statistical approaches quantify the properties of the candidates and their context and score each candidate separately. They are classified as unithood-based, if they assess the attachment strength of the constituents of the candidates; or as termhood-based if they assess the degree that a candidate multiword expression refers to a specific concept. Hybrid systems efficiently mix linguistic and statistic components and sometimes other machine learning tools such as Latent Semantic Analysis, AdaBoost, and various classifiers.

To summarise the main disadvantages of this field, one should consider how all these methods and components are evaluated. Although many information sources have been explored, recognition approaches are evaluated separately, or in small groups, using different corpora and evaluation settings. Thus, it is impossible to directly compare these methods. Below, we present evidence about this diversity in evaluation settings and corpora, in order of presentation in the previous parts of this section. The first paragraph below discusses evaluation details of linguistic and unithood-based approaches. The

second and third paragraphs refer to termhood-based and hybrid approaches, respectively.

Justeson & Katz (1995) evaluated their part of speech patterns on large text collections on a variety of domains: metallurgy, space engineering, and nuclear energy. Séaghdha & Copestake (2007) extracted 1443 compounds nouns from the 90 million word written part of the British National Corpus (BNC), using the output of the RASP parser (Briscoe et al. 2006). Manning & Schutze (1999) evaluated a handful of basic unithood-based approaches (i.e. frequency counts, mean and variance, hypothesis testing and PMI) on a 14 million words corpus from New York Times newswire: from August to November 1990. Dunning (1993) evaluated log-likelihood ratios test on a 31777-word sample of text obtained from the Union Bank of Switzerland, describing market conditions for 1986 and 1987, while Church & Hanks (1990) evaluated the same test on the 1988 AP corpus, an unbalanced sample of American newswire of 44344077 words. Daille et al. (1994) extracted terminology from a 240000 words French corpus, divided into 9541 sentences. 2400 term candidates that appear at least three times in the text were scored using log likelihood ratio and pointwise mutual information. LEXTER (Bourigault 1992) was evaluated on a corpus of 1200000 words in French. It consists of 1700 short papers describing short term projects of the Research Development Division of Electricité de France.

Frantzi et al. (2000); Maynard & Ananiadou (2000a,b) evaluated C-value, NC-value and SNC-value on a corpus of 800000 eye pathology reports, which had been tagged by the Brill part of speech tagger (Brill 1992). From each record, the diagnosis and disease description fields were kept, resulting in a corpus of 810719 words. Statistical Barrier Nakagawa (2000); Nakagawa & Mori (2002) was evaluated on 8834 manually annotated terms contained in 1870 abstracts from the NACSIS Academic Conference Database. Shimohata et al. (1997) tested their method on a computer manual in English comprising 120240 sentences and 1311522 words. Pecina & Schlesinger (2006) evaluate many collocation measures and combinations of measures on data from the morphologically and syntactically annotated Prague Dependency Treebank 2.0 containing about 1.5 million words. 12232 dependency bigrams consisted the collocation candidates.

Schone & Jurafsky (2001) used a randomly-selected corpus consisting of a 6.7 million word subset of the TREC databases (DARPA, 1993-1997). Vivaldi et al. (2001) trained their system on a Spanish corpus taken from the IULA LSP corpus, that was

collected at the Institute for Applied Linguistics of the Universitat Pompeu Fabra. It consists of abstracts of medical reports on asthma that amount to 100000 words, manually annotated. A shorter corpus of 10000 words was used for testing. Fazly & Stevenson (2006, 2007); Fazly et al. (2009); Cook et al. (2007) evaluated the proposed measures on verb-noun pairs from the British National Corpus (BNC). The BNC was parsed using the Collins parser (Collins 1999). Syntactic dependencies were extracted using TGrep2 (Rohde 2004). Verb-noun pairs that fulfil frequency threshold constraints were kept and information about their lemmas were enriched using WordNet (Miller 1995). Bannard (2007) uses for evaluation 979156 unique verb-noun pairs of the BNC as identified by RASP (Briscoe et al. 2006). Van de Cruys & Moirón (2007) tested their NVC extraction method on the 5000 most frequent candidates extracted from the Twente Nieuws Corpus Ordelman (2002), a large corpus of Dutch newspaper texts (500 million words), which has been automatically parsed by the Dutch dependency parser Alpino (van Noord 2006).

It is evident that there is need for an evaluation framework able to evaluate a variety of different methods under common settings and corpora. A limitation is that it is not possible to evaluate under the same setting approaches that aim to recognise different classes of multiword expressions. However, it is possible to define an evaluation framework for a large number of these approaches, far more than the evaluations of the literature. Results will be able to assess which are the best performing approaches and which are the strengths and weaknesses of each one. Moreover, it is often the case that the components which consist a multiword expression recognition system are not evaluated separately. As a result, one is unable to assess the contribution of each component to the final result.

2.3 Survey on Distributional Similarity

In this section, we briefly review the field of distributional representation of context, according to the vector space model (Salton et al. 1975). We introduce the distributional hypothesis which connects context representations with the notion of meaning and discuss how context representations can be compared. Finally, we present a variety of distributional similarity measures in the literature.

2.3.1 Representing Context

The context of all or some instances of a target word occurring in a corpus can be efficiently represented as a vector (Salton et al. 1975). There is a variety of ways for this representation, based on two main choices: (a) How context characteristics are represented as features. (b) What source of information relevant to the cooccurrence of the target word with each feature will be represented by the vector values.

To decide about the former issue, one should consider what distinguishes good context features from bad ones which in turn depends upon the characteristics of the target words. For example, if the target words are nouns, features can be the verbs whose objects are the target nouns. Let the target words be the nouns: house, apartment and room; and the features be the following verbs: buy, rent, sell and book. The following table shows vectors of binary values.

	buy/obj	rent/obj	sell/obj	book/obj
house	1	1	1	0
apartment	1	1	1	0
room	0	1	0	1

A unary feature value means that the corresponding target noun appears as object of the corresponding verb at least once in the corpus. A zero feature value designates the opposite case. Parsing the corpus is necessary to induce this kind of vectors (section 2.2.1.4).

If the target words were verbs, prepositions or determiners, the above features would not be at all informative; because words of these classes are never objects of verbs and all feature values would be zero. Instead of predefining grammatical features for different part of speech classes of target words, one could encode as features any possible relations in which the target word participates together with a context word. This choice would create a very informative feature space and as well very precise to the extent that accurate parsers are available.

Based on the idea of selectional preferences (Resnik 1993), i.e. the fact that words tent to have preferences for other words that they occur with in specific syntactic relations, a number of syntactic vector spaces have been introduced (Pado & Lapata 2007; Baroni & Lenci 2009). Their common basis is that they represent the context of target word occurences by encoding all or a selection of the syntactic relations that include

the target word. The construction of a syntactic vector space usually uses a dependency parser (discussed in section 2.2.1.4). For example the dependency parse of the sentence *"I have liked vivid colours since I was a kid."* is:

I (PRP) [1]	- is **subj**ect of -	***liked (VBN) [3]***
have (VBP) [2]	- is **aux**iliary of -	***liked (VBN) [3]***
vivid (JJ) [4]	- is **a**djectival **mod**ifier of -	***colours (NNS) [5]***
colours (NNS) [5]	- is **d**irect **obj**ect of -	***liked (VBN) [3]***
since (IN) [6]	- starts an **adv**erbial **cl**ause governed by -	***kid (NN) [10]***
I (PRP) [7]	- is **subj**ect of -	***was (VBD) [8]***
kid (NN) [10]	- is **cop**ula of -	***was (VBD) [8]***
a (DT) [9]	- is **det**erminer of -	***kid (NN) [10]***
kid (NN) [10]	- governs an **adj**ectival **cl**ause governed by -	***liked (VBN) [3]***

The numbers within square brackets next to each token represent its position offset to resolve possible conflicts and part of speech tags appear within parentheses. The strings of the middle column in bold designate an alias for each relation. Based on the above dependency parse, target words *"liked"* and *"kid"* can be represented as follows:

	I/subj	*have*/aux	*colours*/dobj	*kid*/adjcl	advcl/*kid*	cop/*was*	*a*/det	adjcl/*liked*
liked	1	1	1	1	0	0	0	0
kid	0	0	0	0	1	1	1	1

To reduce data sparcity, relations can be defined over parts of speech instead of the actual lemmas or surface representations, resulting in the following alternative vector representation:

	PRP/subj	VBP/aux	NNS/dobj	NN/adjcl	advcl/*NN*	cop/VBD	DT/det	adjcl/*VBN*
liked	1	1	1	1	0	0	0	0
kid	0	0	0	0	1	1	1	1

Sometimes, syntactic vector spaces encode higher order of syntactic relations, i.e. syntactic relation chains or paths starting from the target word or ending at it. For example, the second order syntactic relations of *"liked"* in the previous sentence are:

"*vivid*/amod/*colours*/dobj" and "*since*/advcl/*kid*/adjcl". Words in these relations can potentially be omitted or replaced by their parts of speech.

The major disadvantage of syntactic vector spaces is that the feature space is extremely sparse, consisting of a vast number of features. Due to this sparsity, huge amounts of data would are needed to compute practically useful vectors.

For this reason, the bag-of-words approach is used in most of the literature. According to this approach, the context words within a window of $\pm n$ words are chosen as features. Instead of a window, usually all words in the same sentence or paragraph with the target word are represented as features. Sometimes, part of speech information are employed to filter out of the feature set uninformative words. Closed part of speech classes, i.e. prepositions, determiners, conjunctions and pronouns, are considered to be the least informative, because they contain function words with no semantic content. As far as the open classes are concerned, nouns are considered to be of higher discriminative ability than verbs and adjectives (Agirre et al. 2006). The following table presents vectors for target nouns: house, apartment and room and context noun features (the feature set is small for demonstration purposes):

	home	flat	rent	bed
house	1	1	1	0
apartment	1	1	1	0
room	0	1	1	1

Note one of the features, "rent", can be either a noun or a verb. Sometimes, this is an important source of noise, especially in cases that the meaning of the verb is significantly different to the meaning of the noun, e.g. the noun "park" and the verb "to park". To overcome this problem, part of speech information can be incorporated in the features as follows, given that the corpus is part of speech tagged: home/noun, flat/noun, rent/noun and bed/noun.

The second, equally important decision for representing context in vectors is about the source of information that are represented by the vector values. Binary feature vectors, which were discussed previously, are among the least informative representations. It is indicative that in the previous table, almost all values are unary. Instead of binary feature values, it is common to use as feature values the counts of cooccurrence of the corresponding target word and feature word. Extending the previous example,

the following table shows cooccurrence counts feature vectors for target words: house, apartment and room.

	home	flat	rent	bed
house	50	60	5	0
apartment	60	200	7	0
room	0	40	100	9

The target word "house" cooccurs 50 times with "home" within some predefined window. The remaining values can be interpreted in the same vein.

Cooccurrence counts feature values are more informative than binary values, because the indicated the level of relevance of the corresponding feature to the current word. However, cooccurrence counts feature vectors are not directly comparable to each other, especially when the words under comparison appear with significant different frequencies. To overcome this deficiency, counts can be normalised by the counts of target word occurrences. The resulting vectors of frequencies of cooccurrence are directly comparable to each other. Given that "house" occurs 150 times in the corpus, "apartment" 300 times and "room" 150 times, the corresponding frequency vectors are:

	home	flat	rent	bed
house	0.333	0.400	0.033	0.000
apartment	0.200	0.667	0.023	0.000
room	0.000	0.267	0.667	0.060

The table reveals that "home" is more common in the context of "house" than in the context of "apartment", although the counts in the previous tables would suggest the opposite.

Further than frequencies, tf-idf can be employed to distinguish the most discriminative features for each target word. Originally, tf-idf provides a way to weight features so as to reflect how important a word is to a document in a corpus. tf stands for term frequency and idf for inverse document frequency. It should be noted that term is used in this context in a different sense than the subclass of multiword expressions. Here, term refers to a word occurring in some document. Term frequency, tf, of a word, w,

occurring $n_{w,d}$ times in a document, d, is defined as:

$$tf(w,d) = \frac{n_{w,d}}{\sum_{i \in d} n_{i,d}} \tag{2.65}$$

The denominator is the sum of all word counts occurring in document d.

Inverse document frequency, idf, of word w in a corpus, c, containing $|c|$ documents is defined as:

$$idf(w) = \log \frac{|c|}{|\{d : w \in d\}|} \tag{2.66}$$

The numerator is the number of documents in the corpus and the denominator is the number of documents which contain the word w. $tf\text{-}idf$ is defined as the product of tf and idf:

$$tf\text{-}idf(w,d) = tf(w,d) \times idf(w) \tag{2.67}$$

$tf\text{-}idf$ can be modified to reflect the importance of a context word (feature) for a target word. Term frequency, tf, is modified as the cooccurrence count $n_{w,cw}$ of target word, w, and context word, cw. Inverse document frequency, idf, now measures the importance of a context word cw for the set of target words W.

$$tf(w,cw) = n_{w,cw} \tag{2.68}$$

$$idf(cw) = \log \frac{|W|}{|\{w \in W : cw \text{ context of } w\}|} \tag{2.69}$$

Extending the previous example, the importance of context word "buy" for target word "house" is:

$$\begin{aligned} tf\text{-}idf(\text{house}, \text{buy}) &= tf(\text{house}, \text{buy}) \times idf(\text{buy}) \\ &= n_{\text{house,buy}} \times \log \frac{|W|}{|\{w \in W : \text{buy is context of } w\}|} \\ &= 50 \times \log(\frac{3}{2}) \\ &\approx 8.80 \end{aligned}$$

It is worth noting that if a context word cooccurs with all target words then idf is zero for this context word. Thus, only discriminating features are non-zero. The full $tf\text{-}idf$ table of the previous example is the following:

tf-idf vectors	home	flat	rent	bed
house	8.80	0.00	0.00	0.00
apartment	10.57	0.00	0.00	0.00
room	0.00	0.00	0.00	4.29

The table reveals that "bed" is a characteristic feature for "room", while rent is not important for any of the target words, since it cooccurs with all of them.

In the literature, instead of cooccurrence counts, cooccurrence frequencies and $tf\text{-}idf$ feature values, other ways of measuring the relation of each context word to the target word have been used. It is common to compare the target word with each feature word using pointwise mutual information, t-test and log-likelihood values. Potentially, any unithood-based multiword expression extraction approach (section 2.2.2) could be used to compute feature values.

2.3.2 Distributional Hypothesis

In the previous section we discussed ways to present the context of target words in vectors. The distributional hypothesis, introduced in Harris (1954), allows using context vectors as a representation of semantics (meaning):

> The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.

The underlying idea that "a word is characterised by the company it keeps", or in other words that "similar words share similar contexts" was popularised by Firth (1957).

Although it is not obvious from the distributional hypothesis, it is widely hypothesised in the literature that the contexts of semantically different words are dissimilar. Thus, comparing the context vectors of two words can judge if the words are semantically similar. Lexical distributional semantics has been largely used to model word meaning in many fields as computational linguistics (McCarthy & Carroll 2003; Manning et al. 2008), linguistics (Harris 1964), corpus linguistics (Firth 1957), and cognitive research (Miller & Charles 1991). Recently, the distributional hypothesis has been operationally defined in many ways in the fields of physicology, computational linguistics, and information retrieval (e.g. Li et al. (2000); Pado & Lapata (2007); Deerwester et al. (1990)).

The underlying assumption is that contextual distributional similarity correlates with semantic similarity (Miller & Charles 1991). There are cases that this assumption does not hold entirely; mainly in the case that two words are semantically similar but appear in different contexts.

This happens sometimes with pairs of words in hyponymy relation. For example, texts about motorbikes rarely contain the word vehicle, although motorbike is a vehicle. Moreover, the distributional hypothesis does not hold when a word, w_1, is similar to some other word, w_2, only in some sense of word w_1 other than its major sense. Major sense is the most frequently used sense of a word. For example, the major sense of "chip" is "computer chip". Thus, although the "potato chip" sense is semantically similar to "snack", we would expect the contexts of "chip" and "snack" to be semantically dissimilar.

Distributional similarity can be simply extended to cover multiword expressions (Schone & Jurafsky 2001; Baldwin et al. 2003; McCarthy et al. 2003; McCarthy 2006):

> If a multiword expression is compositional, it occurs in similar context with its components. Otherwise, its context will be significantly different to the context of its components.

2.3.3 Measuring distributional similarity

The distributional hypothesis, discussed in the previous section, poses an important issue about the way that two given distributional vectors can be compared. In the relevant literature, several distributional similarity measures are proposed (e.g. Lee (1999); Dagan et al. (1999)). In this subsection, we will discuss a number of distributional similarity measures of the literature, ranging from fundamental ones such as Euclidean distance and cosine similarity to more complicated ones, such as the measure of Lin (1998a).

One of the simplest distributional similarity measures is Hamming distance (or L_1 norm). The measure computes the sum of absolute differences of the values of two n-dimensional vectors. It is usually applied to compare vectors of the same dimension and of binary values. Assuming two vectors $\vec{v}$ and $\vec{u}$ of dimension n, Hamming distance, $L_1(\vec{v}, \vec{u})$, is computed as:

$$L_1(\vec{v}, \vec{u}) = \sum_{i \in [1,n]} |v(i) - u(i)| \tag{2.70}$$

where $v(i)$ and $u(i)$ are the i^{th} dimensional values of vectors v and u, respectively.

Euclidean distance (or L_2 norm) of two n-dimensional vectors $\vec{v}$ and $\vec{u}$ measures the length of the line between points $\vec{v}$ and $\vec{u}$ in the n-dimensional space:

$$L_2(\vec{v}, \vec{u}) = \sqrt{\sum_{i \in [1,n]} (v(i) - u(i))^2} \tag{2.71}$$

where $v(i)$ and $u(i)$ are the i^{th} dimensional values of vectors v and u, respectively. Hamming and Euclidean distance are usually applied to compare vectors of the same dimension and of binary or real values.

One of the most usual distributional similarity measures is cosine similarity (Salton & McGill 1986). The measure is inspired from geometry and calculates the angle between two n-dimensional vectors in the n-dimensional space. Cosine similarity supports real feature values and vectors of the same dimension. Assuming two vectors $\vec{v}$ and $\vec{u}$ of dimension n, their cosine similarity $cos(\vec{v}, \vec{u})$ is computed as:

$$cos(\vec{v}, \vec{u}) = \frac{\sum_{i \in [1,n]} v(i) \times u(i)}{\sqrt{\sum_{i \in [1,n]} v(i)^2 \times \sum_{i \in [1,n]} u(i)^2}} \tag{2.72}$$

where $v(i)$ and $u(i)$ are the i^{th} dimensional values of vectors v and u, respectively. The geometric properties of cosine guarantee that the computed value lies in $[-1, 1]$.

Jaccard similarity coefficient, J, is a combinatorial measure that computes the similarity of two sets as the cardinality of their intersection over the cardinality of their union (Salton & McGill 1986). It is a common measure for computing the distributional similarity of binary vectors of any dimensions. The intersection of two vectors $\vec{v}$ and $\vec{u}$ of dimensions n and m, respectively, is defined as the number of their common non-zero features, and their union as the number of their distinct non-zero features:

$$J(\vec{v}, \vec{u}) = \frac{|\{i \in [1,k] : v(i) \neq 0 \wedge u(i) \neq 0\}|}{|\{i \in [1,k] : v(i) \neq 0 \vee u(i) \neq 0\}|} \tag{2.73}$$

$v(i)$ and $u(i)$ are the i^{th} dimensional values of vectors v and u, respectively, and $k = \max(n, m)$. Jaccard similarity coefficient always lies in $[0, 1]$.

Another usual combinatorial measure is Dice similarity coefficient, $dice$. Dice sim-

ilarity coefficient is computed as double the intersection cardinality over the sum of cardinality of two sets (Lin 1998a,b). Similarly to Jaccard similarity coefficient, this measure accounts for binary feature vectors, i.e. the existence or inexistence of a feature. Given two vectors $\vec{v}$ and $\vec{u}$ of dimensions n and m, respectively, and $k = \max(n, m)$, the measure is defined as:

$$dice(\vec{v}, \vec{u}) = \frac{2 \times |\{i \in [1, k] : v(i) \neq 0 \vee u(i) \neq 0\}|}{|\{i \in [1, k] : v(i) \neq 0\}| \times |\{i \in [1, k] : u(i) \neq 0\}|} \tag{2.74}$$

Dice similarity coefficient always lies in $[0, 1]$. van Rijsbergen (1979) showed that Jaccard's and Dice similarity coefficients are monotonic in each other.

Kullback-Leibler divergence, KL, is a measure of similarity from information theory (Cover & Thomas 1991). It is also known as information divergence, information gain or relative entropy. It is a non-symmetric measure of the difference between two probability distributions, P and Q. Kullback-Leibler divergence measures the expected number of extra bits required to code samples from P when using a code based on Q, rather than using a code based on P. Typically, P represents the "true" distribution of data (i.e observations) or a precise calculated theoretical distribution. Q represents a model, a description, or an approximation of P.

Kullback-Leibler divergence can be applied to compute vector similarity, by using vectors as discrete distributions. Given two vectors $\vec{v}$ and $\vec{u}$ of the same dimension n, the measure is defined as:

$$KL(\vec{v} \| \vec{u}) = \sum_{i \in [1,n]} v(i) \log \frac{v(i)}{u(i)} \tag{2.75}$$

$v(i)$ and $u(i)$ are the i^{th} dimensional values of vectors v and u, respectively. Kullback-Leibler divergence is not symmetric, $KL(\vec{v} \| \vec{u}) \neq KL(\vec{u} \| \vec{v})$, and always non-negative, $KL(\vec{v} \| \vec{u}) \geq 0$.

A major deficiency of Kullback-Leibler divergence is that it can only be applied if $v(i) > 0 \wedge u(i) > 0, \forall i \in [1, n]$, due to the logarithm part of the formula. Taking into account that features i for which $v(i) = 0 \wedge u(i) = 0$ should not contribute to the sum, a new measure is proposed to cover features for which $v(i) > 0 \vee u(i) > 0, i \in [1, n]$. This measure is Jensen-Shannon divergence, JS, and is symmetric and normalised (Rao 1982; Lin 2002):

$$JS(\vec{v}, \vec{u}) = \frac{1}{2}[KL(\vec{v} \| \frac{1}{2}(\vec{v} + \vec{u})) + KL(\vec{u} \| \frac{1}{2}(\vec{v} + \vec{u}))] \tag{2.76}$$

Lee (1999); Dagan et al. (1999) proposed α-skew divergence (s_α), a generalisation of Kullback-Leibler divergence that uses a trainable parameter α:

$$s_\alpha(\vec{v}, \vec{u}) = KL(\vec{v} \| \alpha \cdot \vec{v} + (1 - \alpha) \cdot \vec{u}) \tag{2.77}$$

α can be thought as the degree of confidence in the distribution described by $\vec{u}$ or, equivalently, $(1 - a)$ can be thought of as controlling the amount by which $\vec{u}$ is smoothed by $\vec{v}$.

Confusion probability measures the degree to which a word w_1 can be substituted into the contexts in which w_2 appears. It has been originally used in language modelling, to smooth word cooccurrence probabilities (Essen & Steinbiss 1992). Let $\vec{v}$ and $\vec{u}$ be two n-dimensional vectors representing the context of words w_1 and w_2, respectively, and $P(w)$ be the frequency of word w. Then confusion probability, $conf$, is defined as:

$$conf(\vec{v}, \vec{u}, P(w_1)) = \sum_{i \in [1,n]} v(i) \times u(i) \times \frac{P(w_1)}{P(i)} \tag{2.78}$$

Confusion probability is not symmetric: $conf(\vec{v}, \vec{u}, P(w_1)) \neq conf(\vec{u}, \vec{v}, P(w_2))$.

Kendall's τ looks for correlation between the feature values of two vectors. It is a non-parametric measure that only considers the order of features for each vector sorted in decreasing order of the corresponding feature values. Thus, this measure is meaningful when applied on real valued feature vectors. Kendall's τ has been used for evaluating a system that aims to devise clusters of synonym adjectives (Hatzivassiloglou 1994). Assuming that $\vec{v}$ and $\vec{u}$ are two n-dimensional vectors, it is defined as:

$$\tau(\vec{v}, \vec{u}) = \sum_{i,j \in [1,n]} \frac{sign\left[(v(i) - v(j)) - (u(i) - u(j))\right]}{2\binom{|n|}{2}} \tag{2.79}$$

The intuition behind Kendall's τ is: if sorting the n features by feature values in $\vec{v}$ yields exactly the same ordering as that which results from sorting them according to $\vec{u}$, then $\tau(\vec{v}, \vec{u}) = 1$; if it yields exactly the opposite ordering then $\tau(q, r) = -1$.

Different vector representations (e.g. pointwise mutual information, t-test and log-likelihood values) combined with the distributional similarity measures presented above or other similar, lead to more complicated measures of the literature. For example, Lin (1998a) applies Jaccard similarity coefficient on the positive valued features of PMI

vectors, sim_{J+PMI}. Given vectors $\vec{v}$ and $\vec{u}$ of n and m features representing words w_1 and w_2, respectively, and $k = \max(n, m)$:

$$sim_{J+PMI}(\vec{v}, \vec{u}) = \frac{|\{i \in [1,k] : PMI(w_1, i) > 0 \wedge PMI(w_2, i) > 0\}|}{|\{i \in [1,k] : PMI(w_1, i) > 0 \vee PMI(w_2, i) > 0\}|} \quad (2.80)$$

Similarly, Lin (1998a) proposes another measure based on positive valued PMI vectors:

$$sim_{Lin}(\vec{v}, \vec{u}) = \frac{\sum\limits_{\substack{PMI(w_1,i) \geq 0 \wedge PMI(w_2,i) \geq 0, \\ i \in [1, \min(n,m)]}} PMI(w_1, i) + PMI(w_2, i)}{\sum\limits_{\substack{PMI(w_1,i) > 0, \\ i \in [1,n]}} PMI(w_1, i) + \sum\limits_{\substack{PMI(w_2,i) > 0, \\ i \in [1,m]}} PMI(w_2, i)} \quad (2.81)$$

Further work in distributional similarity exploits transitive measures. For example, Karov & Edelman (1998) propose a measure that judges "taste" as similar to "eat", even if they never occur together, given that "taste" cooccurs with "banana", "eat" cooccurs with "apple" and "banana" cooccurs with "apple". Pecina & Schlesinger (2006) presents a big collection of distributional similarity measures. The ones that were not discussed above are presented, for reasons of completeness, in table 2.16.

2.3.4 Discussion

Several research works evaluated different distributional similarity measures on various tasks, attempting to find if there is a superior one. Lee (1999); Dagan et al. (1999) evaluated many widely used distributional similarity measures on the task of choosing the correct one between two nouns to be object of a verb in some sentence. Their goal was to improve probability estimation of unseen cooccurrences. From the existing measures, Jaccard similarity coefficient achieved the best performance. Except for evaluating existing distributional similarity measures, α-skew divergence was proposed and achieved superior results. However, there seems to be no universally best distributional similarity measure; instead the choice of a measure primarily depends on the application.

For that reason, Weeds et al. (2004); Weeds & Weir (2005) propose a theoretical framework which generally describes distributional similarity measures, using a number

Name	Formula
Cross entropy	$-\sum_{i\in[1,\min(n,m)]} v(i)\log(u(i))$
Reverse cross entropy	$-\sum_{i\in[1,\min(n,m)]} u(i)\log(v(i))$
Reverse confusion probability	$\sum_{i\in[1,\min(n,m)]} v(i)\times u(i)\times\frac{P(w_2)}{P(i)}$
Cosine of PMI	$\frac{\sum_{i\in[1,\min(n,m)]} PMI(w_1,i)\times PMI(w_2,i)}{\sqrt{\sum_{i\in[1,n]} PMI(w_1,i)^2}\times\sqrt{\sum_{i\in[1,m]} PMI(w_2,i)^2}}$
Reverse KL divergence	$\sum_{i\in[1,\min(n,m)]} u(i)\log\frac{u(i)}{v(i)}$
Reverse skew divergence	$KL(\vec{u}\|\alpha\cdot\vec{u}+(1-\alpha)\cdot\vec{v})$

Table 2.16: Various distributional similarity measures

of options for feature values and parameters which control the evaluation metrics in an information retrieval precision-recall basis. Various distributional similarity measures of the literature are described when specific combinations of feature values and parameter setting are chosen. Thus, the framework allows the user to know the internal statistical and linguistic properties of each measure and thus, predict, before any experimental evaluation, the appropriate measure for a given application.

The authors analysed the neighbour sets returned by seven distributional similarity measures. They classified measures in three classes: (a) measures that select high frequent neighbours regardless of the frequency of the target noun (e.g. cosine, Jensen-Shannon divergence, AMCRM[2]-Recall (Weeds et al. 2004)); (b) measures that select low frequency neighbours regardless of the frequency of the target noun (AMCRM-Precision (Weeds et al. 2004)); and (c) measures that select neighbours of similar frequency to the target noun (sim_{Lin}, sim_{J+PMI} (Lin 1998a) and Jaccard similarity coefficient).

The results can be interpreted in terms of distributional generality (Weeds et al. 2004). The first class of measures selects high frequency neighbours that have occurred

[2]Additive, mutual information-based, cooccurrence retrieval model: The feature values are the PMI of the target cooccurring with the corresponding feature. Only, the existence or inexistence of a feature is taken into account when computing similarity.

in more contexts than the target noun has (high recall), i.e. neighbours that can be considered as distributionally more general than the target noun. Methods identifying hypernyms of words could exploit this type of measures. In contrast, the second class of measures selects low frequency neighbours that have occurred in fewer contexts than the target noun has (high precision), i.e. neighbours that can be considered as distributionally less general than the target noun. In the same vein, methods identifying hyponyms of words could exploit this type of measures. The third class of measures tends to select neighbours of a similar distributional generality, both high recall and precision. Methods identifying co-hyponyms could exploit this class of measures.

Ingram & Curran (2007) integrated multiword expression recognition, using t-test and log-likelihood ratios, in a distributional similarity-based shallow parser and evaluated its performance in finding synonyms of the multiword expressions. They report negative results, arguing that the multiword expressions do not occur that often so as to extract significant context vectors for them. However, they state that in some cases the synonyms were of higher quality when multiword expression recognition was included.

2.4 Survey on Multiword Expression Compositionality

In this section, we focus more on the notion of compositionality of multiword expressions and present a review of the relevant literature. In particular, we only review approaches in the literature that address the task of quantifying multiword expressions' compositionality. These approaches exhibit in general much overlap with the techniques developed to recognise multiword expressions, that were presented in section 2.2.

Recall that compositionality is a property of multiword expressions that indicates whether the meanings of the components of a given multiword expression can be combined to predict the meaning of the whole multiword expression. For example, *lemon tree* is compositional, because the multiword expression refers to a *tree* that produces *lemons*. In contrast, *black maria* is non-compositional, since it does not refer to somebody that is called *Maria* and has *black* skin colour. *Black maria* refers to a special police van for transporting prisoners and also to the first American movie production studio in West Orange, New Jersey.

Most approaches in the literature are developed on the following basic idea that emerges from the definition of compositionality: The meaning of a given multiword expressions should be compared to some combination of the meanings of its components.

The multiword expressions should be characterised as compositional if the meanings are similar or as non-compositional if the meanings are dissimilar.

The above general approach to address the task poses several research questions which at the same time are the basic research directions of this field: (a) How can the meaning, i.e. the semantics, of a given multiword expression and its components, be represented? (b) How can the meanings of the multiword expressions components be composed to a single representation? (c) How can we compare the similarity of the multiword expression meaning representation to the combined meaning representation of its components? Except for these technical directions, approaches in the literature explore the task of resolving compositionality for different multiword expression classes. As expected, various properties of each class impose restrictions on the approaches themselves.

The majority of approaches in the literature addresses questions (a) and (c) in the straightforward way of distributional representations and distributional similarity, respectively. As discussed in section 2.3, the semantics of a target word or sequence occurring in a number of sentences can be represented as a vector whose dimensions are some of its context words and whose values are computed by some function over the counts of cooccurrence of the target word or sequence with the corresponding dimension context word (Schone & Jurafsky 2001; Bannard et al. 2003; Baldwin et al. 2003; McCarthy et al. 2003; Katz & Giesbrecht 2006).

One of the first works based on employing the distribution of context was Tapanainen et al. (1998). Hypothesising that "if an object appears only with one verb (or few verbs) in a large corpus we expect that it has an idiomatic nature" the authors proposed *distributed frequency* of object as a measure to determine the non-compositionality of verb-noun collocations. Let f_j be the frequency that object o is governed by verb v_j. The distributed frequency (DF) of object o in a corpus of n triples $\langle f_j, v_j, o \rangle$ with $f_j > C$ is defined as:

$$DF(o) = \sum_{j=1}^{n} \frac{f_j^a}{n^b} \tag{2.82}$$

where a, b and C are constants that depend on the corpus and the parser employed.

A number of approaches in the literature address the issue in a different way. They attempt to capture non-compositionality indirectly, through capturing non-substitutability. In other words, they base their compositionality judgement on the variety of different

words with which the components of the multiword expression cooccur; or on the likelihood of replacing a component with some other distributionally similar word (Lin 1999).

As discussed in sections 2.2.3.4 and 2.2.3.5, Fazly & Stevenson (2006), Fazly & Stevenson (2007) and Fazly et al. (2009) used statistical measures of syntactic behaviour to estimate the probability of a verb and noun combination being an idiom. Although these approached do not specifically detect compositionality, they argue that there is a strong correlation between non-substitutability and semantic idiosyncrasy.

The approach of Cook et al. (2007), discussed in section 2.2.3.8, exploits the idea that the same verb-particle construction appears sometimes as compositional and sometimes as non-compositional. Authors argue that non-compositional usages tend to occur in a small number of canonical forms, while compositional usages of it are less syntactically restricted. Using the automatically determined canonical forms of Fazly & Stevenson (2006) (subsection 2.2.3.4) they distinguished the context of compositional versus non-compositional usages and compared them in three different settings. Some of these settings assume that the literal uses of an noun-verb construction can be inferred by the uses of its head verb.

Several datasets accompanied with human judgements for compositionality have been made available:

1: light verb constructions (LVC) (McCarthy et al. 2003)
2: verb-particle constructions (VPC) (Venkatapathy & Joshi 2005)
3: verb-noun phrases (VNP) (Cook et al. 2008)

2.4.1 Lin (1999)

Lin (1999) performs automatic identification of non-compositional phrases indirectly, via detecting non-productive phrases. The authors hypothesise that non-productive expressions are non-compositional. They use distributional models, statistical measures and dependency triples. The statistical measures are based on the idea that the pointwise mutual information of non-compositional phrases differs significantly from the pointwise mutual information of phrases obtained by substituting each of their components with the 10 most similar words according to a corpus derived thesaurus (Lin 1998a).

Pointwise mutual information is computed over frequencies of dependency triples. A collocation is expressed by a dependency triple (Head (H), Type (T), Modifier (M)) and is treated as the conjunction of three events: $A \equiv (*T*)$, $B \equiv (H * *)$ and $C \equiv$

$(**M)$. Pointwise mutual information (PMI) of a collocation is computed as:

$$\begin{aligned} PMI(H,T,M) &= \log \frac{P(A,B,C)}{p(B|A)\,p(C|A)\,p(A)} \\ &= \log \frac{\frac{|HTM|}{|***|}}{\frac{|*T*|}{|***|}\frac{|HT*|}{|*T*|}\frac{|*TM|}{|*T*|}} \\ &= \log \frac{|HTM| \times |*T*|}{|HT*| \times |*TM|} \end{aligned} \tag{2.83}$$

The above computation is combined with a condition to judge 216 object-verb taken from a collocation database; in particular, noun-noun and adjective-noun pairs as compositional or not. The frequency count of a candidate is a random variable following binomial distribution. For large frequency counts, it can be approximated by normal distribution. If $|HTM| = k$ and $|***| = n$, then $\tilde{p} = \frac{k}{n}$ falls with $N\%$ chance within the interval:

$$\frac{k}{n} \pm z_N \sqrt{\frac{\tilde{p}(1-\tilde{p})}{n}} = \frac{k}{n} \pm z_N \frac{\sqrt{k(1-\frac{k}{n})}}{n} \approx \frac{k \pm z_N\sqrt{k}}{n} \tag{2.84}$$

Let α be a candidate and β be the result obtained by substituting the head of the modifier of α with a similar word. The candidate α is judged as non-compositional if there is no overlap between the 95% confidence interval of the pointwise mutual information of α and β. The results are evaluated manually and also using the Longman dictionary of English idioms (LDOEI). The results of manual evaluation are 15.7% for precision and 13.7% for recall. Against the Longman dictionary of English idioms the results are 39.4% for precision and 20.9% for recall.

2.4.2 Schone & Jurafsky (2001)

Schone & Jurafsky (2001) employ distributional similarity to resolve compositionality of multiword expressions. They compared the distributional vector of each multiword expression candidate to the weighted sum of the distributional vectors of the multiword components. The comparison was done using the cosine similarity measure. The authors evaluated their results against a gold-standard list extracted from WordNet and other machine readable dictionaries and reported that their results were extremely poor. A possible reason is that the coverage of multiword expressions in WordNet is very limited

(Baldwin 2006; Laporte & Voyatzi 2008).

Moreover, Schone & Jurafsky (2001) tried to capture non-compositionality indirectly; through non-substitutability. They used latent semantic analysis models to score non-substitutable multiword expression candidates higher. The results were positive but the authors felt that what is captured by this technique is already handled by collocation extraction statistics.

2.4.3 Bannard et al. (2003)

Bannard et al. (2003) approach the issue of resolving compositionality by assessing how much the semantics of the components of a verb-particle construction, i.e. the verb and the particle, contribute to the semantics of the verb-particle construction itself. They define four different tasks, formed below as questions:

1: is a given verb-particle construction compositional?
2: does one of the components contribute the meaning of the verb-particle construction?
3: does the verb contribute the meaning of the verb-particle construction?
4: does the particle contribute the meaning of the verb-particle construction?

40 verb-particle constructions, extracted from the British National Corpus (BNC), and their components were represented using distributional models. The authors propose four different methods, each of which is able to answer the four tasks discussed above:

1: An adaptation of the method of Lin (1999)
2: Similar to (1) but instead of using a thesaurus based on Lin (1999), this method uses a knowledge-free approach to create the context space.
3: Bannard et al. (2003) argue that the method of Lin (1999), which uses substitution of component words of a multiword expression with semantic neighbours, captures multiword expressions of limited productivity, but not necessarily of compositionality. Their main counter-example is institutionalised expressions, such as *frying pan*, since they are compositional but exhibit a non-productive behaviour similar to non-compositional expressions. To cover this weakness they propose a new substitution-based approach. Instead of computing the context similarity of the original verb-particle construction to the one formed by substitution, they employ corpus-based semantic similarity. A verb particle construction is judged as

compositional if an expression formed by substitution occurs among the nearest 100 verb-particle items to the original, or as non-compositional otherwise.

4: The authors argue that method (3) still confuses institutionalisation with non-compositionality. To overcome this confusion they propose a new method based on measuring how much semantic content the verb and the particle contribute to the semantics of a verb-particle construction. They used their knowledge-free semantic similarity measure (used in method (2)) to compute the cosine similarity of the verb-particle construction to the verb and to the particle, independently.

The classification performed by the above 4 methods for each of the 4 tasks was compared against a gold-standard classification from 26 judges on the same data. Results showed that on all tasks at least one of the methods improves in precision over the baseline of assigning the most frequent label to all items. The methods perform better as far as the contribution of the particles of verb-particle constructions are concerned.

2.4.4 Baldwin et al. (2003)

Baldwin et al. (2003) attempted to resolve compositionality of noun-noun compounds and verb particle constructions (phrasal verbs). They used LSA, as a construction-inspecific model to represent compound, phrasal verbs and their components and to reduce dimensions. Using the cosine measure they computed the similarity of each multiword expression to its head word. Higher similarity scores are hypothesised to indicate more compositional multiword expressions. Evaluation was done against WordNet similarity scores. The results showed a moderate correlation between LSA and WordNet scores, lower for noun-noun compounds. The authors reported that the LSA technique performs better on the low-frequency items than on more frequent items.

2.4.5 McCarthy et al. (2003)

McCarthy et al. (2003) attempt to resolve compositionality of phrasal verbs by computing the degree of overlap between the set of the most similar words to the phrasal verb and the set of the most similar words to the head verb itself (phrasal verb without the particle). Their approach is based on the idea that compositional phrasal verbs are expected to have similar neighbours as for their head verb, while non-compositional phrasal verbs should have different neighbours from their head verb.

Instead of characterising phrasal verbs as compositional or not, McCarthy et al. (2003) assign a score to each phrasal verb that shows how compositional it is. Test verb-particle constructions were extracted from the parse output of the British National Corpus (BNC). A variety of measures based on the nearest neighbours idea were tested and some of them are reported to exhibit significant correlation to human annotation judgements. In particular, the correlation of the proposed measures to human judgements is higher that the correlation of collocation measures, such as pointwise mutual information, to human judgements.

2.4.6 Venkatapathy & Joshi (2005)

Venkatapathy & Joshi (2005) attempt to capture the relative compositionality of multiword expressions that consist of a verb and a noun. They evaluate a number of collocation-based (1-5, below) and context-based measures (6-7, below):

1: Candidate frequency
2: pointwise mutual information (Church & Hanks 1990)
3: least mutual information difference with similar collocations, i.e. the method of Lin (1999) using the original thesaurus of Lin (1998a) for obtaining the similar collocations.
4: Distributed frequency of an object, i.e. the average frequency of occurrence of an object over all verbs cooccurring with the object more frequently than a predefined threshold.
5: Distributed frequency of an object using the verb information, i.e. the similarity of the target verb and the verbs cooccurring with the target object more frequently than a pre-defined threshold.
6: Dissimilarity of the context of the verb-object pair to the context of its constituent verb, representing context using latent semantic analysis (LSA) (Baldwin et al. 2003)
7: Similarity of the verb-object pair with the verbal form of the object, using the above context representation.

As test data, they used 800 verb-noun collocations extracted from the British National corpus (BNC). 2 human annotators scored these candidate as far as compositionality is concerned to create the gold-standard rating. Venkatapathy & Joshi (2005) show

that measures 3, 5 and 7 in the above list, correlate with human judgements better than the others. In particular, measure 7, which captures support verb constructions, e.g. *give a smile*, achieves the best correlation. Moreover, Venkatapathy & Joshi (2005) integrate the above 7 measures as support vector machine (SVM) features to compute a candidate ranking function. They show that the SVM-based ranking function correlates with human judgements significantly better than single features do.

2.4.7 Katz & Giesbrecht (2006)

Similarly to Baldwin et al. (2003), Katz & Giesbrecht (2006) used *latent semantic analysis* (LSA) to model the local context of multiword expressions and their components. To decide compositionality, they compute an approximation of the compositional meaning of each multiword expressions and compare it with its meaning as it is used on the whole. They characterise the multiword expression as compositional if the cosine similarity score is higher than a threshold or non-compositional otherwise.

The LSA model was trained on a local German newspaper corpus, Süddeutsche Zeitung for 2003, with about 42 million words. For evaluation they used the manually annotated preposition-noun-verb (PNV) dataset of Krenn (2000). Results showed that the threshold, above which multiword expressions are judged as compositional, much affects the correlation with human annotated data.

2.4.8 Piao et al. (2006)

Piao et al. (2006) employed a semantic field taxonomy based on Lancaster English semantic lexicon to address the problem of resolving compositionality. They propose a measure which retrieves semantic information from the taxonomy about the target phrasal verb and its components and then compares them based on this semantic information. Their approach is evaluated on 89 phrasal verbs from the data set of (McCarthy et al. 2003). The authors employed the Spearman correlation coefficient, which only compares the order of items in two lists and not the actual scores.

The results showed that the system ordering for the vast majority of the test data correlates strongly with the human annotations. In the system's output list, some test data instances are ranked in very different order than in the human annotation list. However the authors argue that the overall correlation is comparable or better than the correlation using other measures and state that the proposed taxonomy-based approach could be

used in cooperation with other measures.

2.4.9 McCarthy et al. (2007)

McCarthy et al. (2007) explore the use of selectional preferences for detecting non-compositional verb-object combinations (VPC). In particular, they propose three models which focus on argument types instead of tokens and use as representation classes the WordNet hierarchy or entries from a distributional thesaurus. The components of each target verb-object combinations are classified in some class of the employed selectional preferences model and thus probabilities are much more significant, effectively addressing data sparsity.

The authors used the manually annotated dataset of Venkatapathy & Joshi (2005) and show highly significant correlation between the human annotations and measures based on their proposed selectional models. In particular, their model based on the distributional thesaurus exhibits higher correlation with the human judgements than any of the features used in Venkatapathy & Joshi (2005). Moreover, they combined their most successful model with the four most successful features of Venkatapathy & Joshi (2005) (features 1, 2, 3 and 7 in the list of section 2.4.6) using a support vector machine (SVM). The resulting classifier achieves the highest correlation published for the dataset in use.

2.4.10 Villavicencio et al. (2007)

Villavicencio et al. (2007) attempt to classify multiword expressions of various types, in a task specific setting. They aim to include non-compositional multiword expressions as new lexical entries in the English Resource Grammar (ERG), a broad-coverage precision Head-Driven Phrase Structure Grammar (HPSG), so as to increase its coverage and accuracy. The authors argue that non-compositional multiword expressions can be recognised by their statistical properties, and evaluate three statistical measures for this task: (a) pointwise mutual information, (b) Pearson's chi-square test, and (c) permutation entropy. Let w_x be a word of some corpus and x its index within the corpus. The probability of a trigram $w_1w_2w_3$ can be estimated from the number of occurrences (n) of each permutation:

$$p(w_1w_2w_3) = \frac{n(w_1w_2w_3)}{\sum_{\forall i,j,k} n(w_iw_jw_k)} \tag{2.85}$$

Permutation entropy (*PE*) is defined as follows:

$$PE = -\sum_{\forall i,j,k} p(w_i w_j w_k) \ln [p(w_i w_j w_k)] \tag{2.86}$$

An interesting difference from the rest of the literature is that this approach evaluates whether the above measures can distinguish non-compositional multiword expressions from a pool of N-grams, which can be compositional or non-compositional multiword expressions or just sequences of words. Villavicencio et al. (2007) concluded that pointwise mutual information and permutation entropy seem to differentiate between compositional and non-compositional multiword expressions.

2.4.11 Discussion

In this section, we reviewed most of the literature that addresses the issue of resolving compositionality of multiword expressions. Authors proposed a large variety of methods, applied them on various types of multiword expressions and evaluated their results mainly on manually annotated data. The methods can be coarsely classified between those based in context distributions and those based on substitutions.

The former compare the context distribution of the target multiword expression to the context distribution of one of its components, or to a combination of the context distribution of its parts. These methods constitute the majority of the relevant literature. Substitution-based methods quantify how rigid a target multiword expression is. Hypothesising that non-compositional expressions are more rigid than compositional ones, substitution-based methods can indirectly decide compositionality.

All methods but Fazly & Stevenson (2007) and Fazly et al. (2009) ignore that the same multiword expression might occur both as compositional and non-compositional, in different contexts. Fazly & Stevenson (2007) and Fazly et al. (2009) support a per-instance view to the issue of compositionality of multiword expressions. They hypothesise that non-compositional occurrences are less syntactically flexible than compositional ones and employ the canonical forms of Fazly & Stevenson (2006) to distinguish compositional versus non-compositional instances. However, there is much space for developing methods addressing the problem of per-instance compositionality. Sense induction can be employed to partition the context vector of a target multiword expressions and its components.

Most methods were applied on light verb constructions (LVC), verb-particle constructions (VPC), or verb-noun phrases (VNP). For these classes of multiword expressions, datasets with human annotations have been made available. We observe that the compositionality of multiword expressions that consist of adjectives and nouns or nouns only is much less exploited.

2.5 Survey on Distributional Semantics Composition

In section 2.3, we discussed ways that the context of a target word can be represented as a vector. In subsection 2.3.2, we explained the distributional hypothesis (Harris 1954), which allows using context vectors as the representation of meaning of the corresponding target words.

As distributional semantics has proven over the years to be a viable solution to describe word meaning, it has been extended to word sequences. Two different ways to address this issue have been proposed: (a) by reformulation of the distributional hypothesis to cover word sequences (Lin & Pantel 2001); and (b) by definition of compositional distributional semantics (CDS) models (Mitchell & Lapata 2008; Jones & Mewhort 2007).

On the one hand, (a), Lin & Pantel (2001) propose a redefinition of the distributional hypothesis for patterns, i.e. word sequences representing partial verbal phrases. The extended hypothesis is called pattern distributional hypothesis and it has been defined for determining inference rules, i.e. equivalent patterns describing similar meanings. The distributional vectors representing the meaning of these patterns is derived directly by transforming into vectors their occurrences in a corpus. The major problem created by this decision is data sparsity. Patterns of different length appear with very different frequencies in the corpus. This fact detrimentally affects the statistics and makes comparisons difficult.

On the other hand, (b), compositional distributional semantics (CDS) propose models that are able to compute the distributional semantics of a sequence by composing the distributional vectors of the component words of the sequence (Mitchell & Lapata 2008; Jones & Mewhort 2007). This approach succeeds in overcoming the sparsity problem of the extended distributional hypothesis, since the distributional meaning of sequences of different length is obtained by composing distributional vectors of single words.

However, there are several issues relevant to CDS models that should be investigated. The proposed CDS models have not been explored in detail, since many of the models

have a large number of parameters. In the experimental part of Mitchell & Lapata (2008) only a selection of parameter values is evaluated.

Evaluation in Mitchell & Lapata (2008) is based on a word sequence similarity test (Kintsch 2001): CDS models are used to compose vectors for given verb-noun pairs, in which we know that the verb is ambiguous. However, the noun disambiguates the use of the verb in the pair. There are also two other verbs available for each pair, one of which matches the disambiguated meaning of the pair. Then, evaluation decides whether the composed vector is closer to the disambiguating verb than the other option. For example, "ran" means "gallop" if its subject is "horse", while it means "dissolve" if its subject is "colour". Given the pair ("horse","ran"), CDS models compose a vector for it and then the similarity between this and "gallop" is expected to be higher than its similarity to "dissolve".

The proposed CDS methods do not satisfactorily correlate with human annotated data mainly because the human inter-annotation agreement is low. As a result, it is not clear from the evaluation whether or not the resulting vectors for word sequences successfully represent their distributional semantics (Lenci 2008).

In the remainder of this section, we discuss the extension of the distributional hypothesis to word sequences, its implications and several related issues about representing context (subsection 2.5.1). Subsequently, we introduce a generic CDS model (subsection 2.5.2), and then present several CDS models of the literature. Finally, we discuss the shortcoming of this research field.

2.5.1 Modelling the Context of Word Sequences

Representing context distributions of words was discussed in section 2.3. In succession, the distributional hypothesis of Harris (1954) was presented in section 2.3.2. It can be operationally defined as: "similar words share similar contexts", which means that the distribution of the context of a word in some corpus is closely related to its semantics. Similarly to the distributional hypothesis for words, the distributional hypothesis for word sequences can be defined:

> Similar word sequences share similar contexts.

In other words, the context of occurences of a sequence of words in some corpus is closely related to its semantics. Contexts of different sequences of words can be compared to decide whether the sequences are semantically similar or not. Although, the

distributional hypothesis for word sequencies deals with similarity or relatedness, it cannot be used for more complex tasks such as textual entailment.

The above hypothesis poses issues about what should be considered as context and how it should be represented. Similarly to the distributional hypothesis for words, a number of different approaches have been proposed for modelling context. Usually, words that occur within a window before and after the target sequence are considered as context. This approach describes a bag-of-word vector space (Li et al. 2000). The context of word sequences of any length is defined as being in the same context space as the context of words (Mitchell & Lapata 2008). Two word sequences, $\mathsf{s} = \mathsf{w}_1^s \dots \mathsf{w}_k^s$ and $\mathsf{t} = \mathsf{w}_1^t \dots \mathsf{w}_m^t, k \neq m$, can be compared via their context distributions, $\vec{s}$ and $\vec{t}$ respectively, as follows:

$$\mathsf{s} \approx \mathsf{t} \Leftrightarrow \vec{s} \approx \vec{t} \tag{2.87}$$

Alternatively, the sentence in which the target sequence occurs can be parsed and context words together with the syntactic relations that connect them to the target sequence can be used as features. This approach is based on the idea of selectional preferences (Resnik 1993) and describes a syntactic vector space (Pado & Lapata 2007; Baroni & Lenci 2009). In the same vein, Lin & Pantel (2001) have extended the distributional hypothesis to patterns, i.e. sequences of words with two arguments representing a relation, e.g. "X is the director of Y". This augmented hypothesis is called pattern distributional hypothesis. Context words are fillers of the arguments of the patterns and context is modelled according to the observed syntactic structures. For example, the context words of pattern "X is the director of Y" are the fillers of variables X and Y in some corpus.

Moreover, document vector spaces (Deerwester et al. 1990) have been proposed as possible models for context, based on Latent Semantic Analysis (LSA). LSA (Deerwester et al. 1990) has been proposed in Information Retrieval to find efficient ways of storing and retrieving documents. As the original aim was to compress the vector space of documents, LSA is not a direct application of the distributional hypothesis. Contexts of words are defined differently; as documents. However, the idea is similar: in LSA, words are considered similar (or related) if they occur in similar documents. The major problem in combining this representation with the distributional hypothesis foe word sequences is that context consists of documents and these documents are single units that cannot be analysed into components.

Several approaches propose to use tensor algebra for composition and interpretation of word sequence meaning (Smolensky 1990; Clark & Pulman 2007). In tensor algebra, the definition of context space for word sequences depends on the length of the word sequences: vectors for words, bi-dimensional matrices for 2-word sequences, tri-dimensional matrices for 3-word sequences, on so on. This poses a serious issue, because the representation allows comparing sequences of the same length, only. However, there are several advantages of these approaches against the dominant bag-of-words approach: Firstly, sequences that consist of the same words in different order are successfully represented by different context matrices (Clark & Pulman 2007). For example, the context representations of "John likes Mary" and "Mary likes John" are different. Secondly, the representation accommodates a natural way to compare two word sequences of the same length. For example, suppose two word sequences: s="man reads newspaper" and t="woman browses magazine". Using the following property of tensor algebra:

$$(w_1 \otimes w_2).(w_3 \otimes w_4) = (w_1.w_3) \times (w_2.w_4) \tag{2.88}$$

the similarity between word sequences s and t can be calculated by simply comparing the respective pairs multiplying the inner products:

$$\overrightarrow{man}.\overrightarrow{woman} \times \overrightarrow{reads}.\overrightarrow{browses} \times \overrightarrow{newspaper}.\overrightarrow{magazine} \tag{2.89}$$

where $\vec{w}$ stands for the context vector of word "w".

Despite the advantages of semantic composition approaches based on tensor algebra, that were discussed above, we choose not to pursue them in this thesis. There are several reasons for this decision. Firstly, in these approaches, sequences of different length are not by any means comparable. Secondly, the fact that the number of features increases exponentially with sequence length creates data sparsity, thus one needs vast amounts of data to reliably compose representations of long sequences. Thirdly, the resulting representations of tensor algebra composition models cannot be directly compared with gold-standard representations for the same sequences. This inability leads to indirect evaluations of semantic composition models, which we intend to avoid.

2.5.2 A generic compositional distributional semantic model

A compositional distributional semantic model is a function $\odot$ that derives compositionally the distributional vector for a sequence of words. Given a sequence of words $\mathsf{s} = \mathsf{w}_1 \cdots \mathsf{w}_n$, a compositional distributional semantic model computes the distributional vector $\odot(\mathsf{s})$ that describes the sequence by combining the distributional vectors $\vec{w}_i$ of the component words w_i. This general definition of compositional distributional semantic models can be formally expressed as:

$$\odot(\mathsf{s}) = \odot(\mathsf{w}_1 \ldots \mathsf{w}_n) = \vec{w}_1 \odot \cdots \odot \vec{w}_n \tag{2.90}$$

A good compositional distributional semantic model should be able to define a function $\odot$ that produces a good distributional meaning for any word sequence. This generic model has been fairly well studied and many different models have been proposed and tested. In the following subsections we will present in detail three compositional distributional semantic models: (a) a model building on basic vector operations, i.e. weighted addition and multiplication; (b) a model based on selectional preferences of the components of sequence s proposed in (Erk & Padó 2008); and (c) a model that projects all vectors into the same dimensional space, called BEAGLE and proposed in (Jones & Mewhort 2007).

2.5.2.1 Mitchell and Lapata Model

Mitchell & Lapata (2008) introduce a general setting for compositional distributional semantic models. Their setting instantiates a class of all compositional distributional semantic models of equation 2.90. They deal with 2-word sequences $\mathsf{s} = \mathsf{xy}$ and the proposed equation is still general and need further specification:

$$\odot(\mathsf{s}) = \odot(\mathsf{xy}) = f(\vec{x}, \vec{y}, R, K) \tag{2.91}$$

$\vec{x}$ and $\vec{y}$ are the distributional vectors of x and y, R is the particular syntactic and/or semantic relation connecting x and y, and, K represents the amount of background knowledge that the vector composition process takes into account.

This general compositional distributional semantic model is then realised in three different models: an additive, a multiplicative and a combined one. The generic *additive*

CDS model sums the vectors of the components of the sequence $\vec{x}$ and $\vec{y}$ in a new vector $\vec{z}$ which represents the composed distributional semantics of sequence s:

$$\odot(\mathsf{s}) = \vec{z} = A\vec{x} + B\vec{y} \tag{2.92}$$

A and B are two square matrices which capture the relation R and the background knowledge K, of equation 2.91. The fact that A and B are matrices and not vectors allows the value of the i^{th} dimension of $\vec{z}$ to depend upon all dimensions of vectors $\vec{x}$ and $\vec{y}$. The model of equation 2.92 is still a general model because matrices A and B should be further defined.

In Mitchell & Lapata (2008), only one additive model is explored: the basic additive model (BAM), for which A and B are instantiated as unary scalars. The resulting model is a linear combination of the vectors which represent the components of sequence s:

$$\odot(\mathsf{s}) = \vec{z} = \alpha\vec{x} + \beta\vec{y} \tag{2.93}$$

Following a simplistic parameterisation, in (Mitchell & Lapata 2008) two versions of this model are evaluated. One in which the scalar parameters are unary: $\alpha = \beta = 1$, and a trainable one in which the parameters are trained on a small held-out set.

The generic *multiplicative* CDS model is based on the tensor product between the two component vectors $\vec{x} \times \vec{y}$:

$$\odot(\mathsf{s}) = \vec{z} = C \times \vec{x} \times \vec{y} \tag{2.94}$$

C is a tensor of rank 3 that projects the tensor $\vec{x} \times \vec{y}$ in the original space. Equation 2.94 presents a generic multiplicative model, because the construction of C, which represents the syntactic and/or semantic relation R and the background knowledge K of equation 2.91, needs to be explained.

Mitchell & Lapata (2008) exploit a simplified instantiation of C in equation 2.94. Their basic multiplicative model (BMM) assumes a unary scalar in the position of the tensor C:

$$\odot(\mathsf{s}) = \vec{z} = \vec{x}^T\vec{y} \tag{2.95}$$

The equation assumes that vectors $\vec{x}$ and $\vec{y}$ are row vectors, so that the transposition creates a column vector.

	vector dimensions				
	between	gap	process	social	two
close	$< 27,$	$3,$	$2,$	$5,$	$24 >$
interaction	$< 23,$	$0,$	$3,$	$8,$	$4 >$

Table 2.17: Example frequency vectors of the components of s = "close interaction"

The combined model computes a linear combination of the basic additive and multiplicative models:

$$\odot(\mathsf{s}) = \vec{z} = \alpha\vec{x} + \beta\vec{y} + \gamma\vec{x}^T\vec{y} \tag{2.96}$$

α, β and γ are again trainable parameters.

Below we explain how the basic additive model (BAM) and the basic multiplicative model (BMM) work using the example vectors of table 2.17. The table shows a vector for each of the words "close" and "interaction". The representation uses only five indicative features: "between", "gap", "process", "social" and "two". These features are among the most frequent features for the target words in the British National Corpus. The values are scaled down, keeping their proportionality.

Supposing the vectors of Table 2.17, the basic additive model and the basic multiplicative model are computed as:

$$\begin{aligned} \odot_{BAM}\,(\text{close interaction}) &= < 27,3,2,5,24 > + < 23,0,3,8,4 > \\ &= < 50,3,5,13,28 > \end{aligned} \tag{2.97}$$

$$\begin{aligned} \odot_{BMM}(\text{close interaction}) &= < 27,3,2,5,24 >^T < 23,0,3,8,4 > \\ &= < 621,0,6,40,96 > \end{aligned} \tag{2.98}$$

2.5.2.2 Erk and Pado Model

Erk & Padó (2008) proposed a model able to disambiguate the distributional meaning of a word w in the context of the sequence s. Given the general distributional vector $\vec{w}$ of a word w and a sequence s, the model computes a vector $\vec{w}_{\mathsf{s}}$ that represents the specific distributional meaning of word w occurring in s. Although this model was originally developed to address this setting, it can be modified into a compositional distributional semantic model. The model has been tested on a lexical substitution task and on the experimental setting of Mitchell & Lapata (2008). The authors report significant im-

provement over the state-of-the-art on the former task and insignificant improvements on the latter.

To express the model formally we introduce the following operator, $\oslash$:

$$\vec{w}_{\mathsf{s}} = \oslash(\mathsf{w}, \mathsf{s}) \tag{2.99}$$

The following properties of operator $\oslash$ explain its functionality:

- $\oslash$ computes a different vector for each word w_i in the sequence $\mathsf{s} = \mathsf{w}_1 \dots \mathsf{w}_n$:

$$\oslash(\mathsf{w}_i, \mathsf{s}) \neq \oslash(\mathsf{w}_j, \mathsf{s}), \quad \forall i, j \in [1, n] : i \neq j \tag{2.100}$$

- $\oslash$ computes different vectors for a word w_i appearing in different sequences s_k and s_l:

$$\oslash(\mathsf{w}_i, \mathsf{s}_k) \neq \oslash(\mathsf{w}_i, \mathsf{s}_l), \quad \text{for } k \neq l \tag{2.101}$$

The model of equation 2.99 can be modified to compose the distributional meaning of a sequence s, based on the semantic head word of the sequence (c.f. Pollard & Sag (1994)). The modification hypothesises that the meaning of a sequence s is shaped by the word that governs the sequence. For example, the meaning of the word sequence "eats mice" is governed by the verb, "eats". Assuming that h is the semantic head word of s, the model of Erk & Padó (2008) can be modified into a compositional distributional semantic model:

$$\odot(\mathsf{s}) \approx \oslash(\mathsf{h}, \mathsf{s}) \tag{2.102}$$

This model is significantly different to the model of Mitchell & Lapata (2008), presented in subsection 2.5.2.1. The generic additive model of equation 2.92 and the generic multiplicative model of equation 2.94 integrate the information provided by the relation R and the background knowledge K of the general equation 2.91 into matrices A and B, and tensor C, respectively. The model of Erk & Padó (2008) (equation 2.102) uses the relation R and the background knowledge K more explicitly; by exploiting the relation of the sequence s with the head word h.

Supposing a sequence that consists of two words, $\mathsf{s} = \mathsf{xy}$, the model of Erk & Padó (2008) uses the selectional preferences of each word and the syntactic and/or semantic relation that links the two words. The two words x and y are related with an oriented

syntactic relation r (e.g. r=adjectival_modifier, r=noun_object). The syntactic relation can be integrated into the notation of the sequence: $\mathsf{s} = \mathsf{x} \overset{r}{\leftarrow} \mathsf{y}$.

To keep track of its selectional preferences, each word w is represented with a triple:

$$(\vec{w}, R_{\mathsf{w}}, R_{\mathsf{w}}^{-1}) \tag{2.103}$$

$\vec{w}$ is the distributional vector of word w, R_{w} is the set of the distributional vectors that represent the direct selectional preferences of the word w, and R_{w}^{-1} is the set of vectors that represent the indirect selectional preferences of the word w. In particular, given a set of syntactic relation types $\mathcal{R}$, the sets R_{w} and R_{w}^{-1} contain respectively a selectional preference vector $R_{\mathsf{w}}(r)$ and $R_{\mathsf{w}}(r)^{-1}$ for each $r \in \mathcal{R}$.

Erk (2007) proposes a way to compute the selectional preferences of a word w. Let the triple $(\mathsf{w}', r, \mathsf{w})$ denote the cooccurrence of w' and w in oriented syntactic relation r. $f(\mathsf{w}', r, \mathsf{w})$ denotes the frequency of the triple $(\mathsf{w}', r, \mathsf{w})$ in the corpus and $N = |\{(\mathsf{w}', r, \mathsf{w}) : f(\mathsf{w}', r, \mathsf{w}) > 0\}|$. The distributional vector that represents the direct selectional preferences of word w when occurring in some relation r is:

$$R_{\mathsf{w}}(r) = \frac{1}{N} \sum_{w'} f(\mathsf{w}', r, \mathsf{w}) \vec{w'} \tag{2.104}$$

Accordingly, the distributional vector that represents the indirect selectional preferences of word w when occurring in some relation r is:

$$R_{\mathsf{w}}^{-1}(r) = \frac{1}{N} \sum_{w'} f(\mathsf{w}, r, \mathsf{w}') \vec{w'} \tag{2.105}$$

Taking into account the above definitions, the model of Erk & Padó (2008) can be further specified. If x is the semantic head of sequence $\mathsf{s} = \mathsf{x} \overset{r}{\leftarrow} \mathsf{y}$:

$$\odot\,(\mathsf{s}) = \oslash(\mathsf{x}, \mathsf{x} \overset{r}{\leftarrow} \mathsf{y}) = \vec{x} \odot R_{\mathsf{y}}(r) \tag{2.106}$$

Otherwise, if y is the semantic head:

$$\odot\,(\mathsf{s}) = \oslash(\mathsf{y}, \mathsf{x} \overset{r}{\leftarrow} \mathsf{y}) = \vec{y} \odot R_{\mathsf{x}}^{-1}(r) \tag{2.107}$$

Operator $\odot$ can be realised using any chosen compositional distributional semantic model, e.g. the basic additive model or the basic multiplicative model of Mitchell &

Lapata (2008) discussed in subsection 2.5.2.1.

The following example explains the functionality of the model of Mitchell & Lapata (2008) in practice. We will use it to compute the distributional vector of a sequence of two words, $\mathsf{s} = \mathsf{xy}$, where the first word, x, is an adjective and the second word, y, is a noun. The semantic head of the sequence is the noun, i.e. y. Given the syntactic relation type r = adjectival_modifier $\in \mathcal{R}$ that is oriented from the noun to the adjective, the sequence can be rewritten as: $\mathsf{s} = \mathsf{x} \xrightarrow{r} \mathsf{y}$. The distributional vector representing the meaning of the sequence can be computed as:

$$\odot(\mathsf{xy}) = \oslash(\mathsf{y}, \mathsf{y} \xrightarrow{r} \mathsf{x}) = \vec{y} \odot R_{\mathsf{a}}(r) \tag{2.108}$$

Following the example of table 2.17 and hypothesising that the adjectival_modifier relation between "close" and "interaction" occurs 5 times, between "close" and "gap" 3 times and between "close" and "process" 2 times:

$$R_{\text{close}}(\text{adjectival\_modifier}) = 5 \cdot \overrightarrow{\text{interaction}} + 3 \cdot \overrightarrow{\text{gap}} + 2 \cdot \overrightarrow{\text{process}}$$

2.5.2.3 BEAGLE

Jones & Mewhort (2007) present a compositional distributional semantic model called BEAGLE, in which all distributional vectors are of the same dimension. The feature space in which each distributional vector is represented is defined independently of the features that have been discussed sofar: bag-of-words, selectional preferences and syntactic features. The idea of this new feature space which is called *environmental feature space*, makes the model of Jones & Mewhort (2007) substantially different to the other models.

To explain the way that context is represented in the *environmental feature space*, we hypothesise some initial representation of context. For simplicity, context is defined as a bag-of-words, i.e. given a target word w, each feature i of its contextual vector $\vec{w}$ represents a word w_i. Jones & Mewhort (2007) define a signature vector $\vec{e}^{(i)}$ of dimension D (called *environmental vector*) for each context word w_i. The *environmental vector*, $\vec{e}^{(i)}$, is the representation in the new feature space of each context word, w_i, i.e. feature of the initial feature space. *Environmental vectors* are obtained from random distributions. In particular each dimension $e_j^{(i)}, j \in [1, D]$ is a randomly extracted value from a Gaussian distribution with mean $\mu = 0$ and variance $\sigma = 1/\sqrt{D}$ where D is defined beforehand.

We provide an example to explain the mapping of the initial feature space to the environmental feature space. Firstly, we choose a value for the dimensions of the environment feature space, $D = 16$. Usually, this value is some power of 2. Subsequently, we define a Gaussian random generator $G(\mu, \sigma)$ fixing $\mu = 0$ and $\sigma = \frac{1}{\sqrt{16}} = 0.25$. For each context word, w_i, of the target word, w, the model defines a vector $\vec{e}^{(i)}$ of 16 values. Each one of these values v is generated from the Gaussian random generator $G(\mu, \sigma)$:

$$\vec{e}^{(i)}_j = v, \quad j \in [1, 16] \tag{2.109}$$

Given the contextual vector $\vec{w}$ of a word w in the initial feature space, the contextual vector $\vec{\overline{w}}$ of this model in the new feature space is obtained by summing up the signature vectors of the words found in the contexts of w:

$$\vec{\overline{w}} = \sum_i w_i \vec{e}^{(i)} \tag{2.110}$$

where w_i is the value of the i-th feature of the original contextual vector $\vec{w}$.

In Jones & Mewhort (2007) composition is realised using circular convolution. Each feature value $z_i, i \in [1, D]$ of the composed distributional vector $\vec{z}$ depends not only on the corresponding environmental vector values $\overline{x}_i$ and $\overline{y}_i$ of the components of sequence $\mathsf{s} = \mathsf{xy}$. In contrast, all environmental feature values in vectors $\overline{x}$ and $\overline{y}$ contribute in the computation of every feature value $z_i, i \in [1, D]$. The following equation explain how this dependencies are realised in the circular convolution multiplicative model with environmental vectors:

$$\begin{aligned} z_i &= \sum_{j=1}^{n} \overline{x}_j \overline{y}_{\lceil i-j \rceil} \\ \lceil i-j \rceil &= \begin{cases} i-j, & \text{if } i > j \\ n+i-j, & \text{if } i \leq j \end{cases} \end{aligned} \tag{2.111}$$

Below, we include an example to clarify the computation of circular convolution. However, the example is performing the computation on vectors in the initial bag-of-words feature space, and not the environmental one. For $\mathsf{s} =$ close interaction and using

the feature vectors of table 2.17, the composed vector can be computed as:

$$\begin{aligned}
\odot_{CCMM}(\mathsf{s}) &= \odot_{CCMM}(\text{close interaction}) \\
&= < 27 \cdot 4 + 3 \cdot 8 + 2 \cdot 3 + 5 \cdot 0 + 24 \cdot 23, \\
&\quad 27 \cdot 23 + 3 \cdot 4 + 2 \cdot 8 + 5 \cdot 3 + 24 \cdot 0, \\
&\quad 27 \cdot 0 + 3 \cdot 23 + 2 \cdot 4 + 5 \cdot 8 + 24 \cdot 3, \\
&\quad 27 \cdot 3 + 3 \cdot 0 + 2 \cdot 23 + 5 \cdot 4 + 24 \cdot 8, \\
&\quad 27 \cdot 8 + 3 \cdot 3 + 2 \cdot 0 + 5 \cdot 23 + 24 \cdot 4 > \\
&= < 108 + 24 + 6 + 0 + 552, \\
&\quad 621 + 12 + 16 + 15 + 0, \\
&\quad 0 + 69 + 8 + 40 + 72, \\
&\quad 81 + 0 + 46 + 20 + 192, \\
&\quad 216 + 9 + 0 + 115 + 96 > \\
&= < 690, 664, 189, 339, 436 >
\end{aligned}$$

2.5.3 Discussion

There are three major issues identified in this review as potential fields of further research. (a) The general additive and multiplicative CDS models in Mitchell & Lapata (2008) define several parameters. This parametric space is not exploited to a great extent. (b) Evaluations of the proposed models in Mitchell & Lapata (2008) and Jones & Mewhort (2007) suffer drawbacks so that it is not clear whether the models perform well or badly, especially in comparison with each other. (c) Although combinations of the models proposed in Mitchell & Lapata (2008), Erk & Padó (2008) and Jones & Mewhort (2007) are straightforward to define, they are not exploited yet.

In equations 2.92 and 2.94 we presented the general additive and multiplicative CDS of Mitchell & Lapata (2008). The general additive CDS model uses two matrices A and B which control the contributions of the component words into the composed vector of sequences. However, in the experimental stage of Mitchell & Lapata (2008) only a simplistic substitution of these matrices with scalars is evaluated: a version with unary scalars (i.e. all features are considered to contribute equally to the composed sequence vector) and a version where the scalars are trained on a small corpus. The general mul-

tiplicative CDS model uses a tensor of rank 3 to project the tensor $\vec{x} \times \vec{y}$ on the original feature space. Again, in terms of evaluation, this tensor is oversimplified to a unary scalar. Thus, there is substantial space for further exploiting these parametrisations. Moreover, the possibility of a supervised setting in which the matrix parameters could be trained is not exploited.

Evaluation of the CDS models in Mitchell & Lapata (2008) is based on a word sequence similarity test (Kintsch 2001). In particular, the similarity of the composed vector representing a sequence is compared with the vector of each of two given alternatives. The most similar of these alternatives is chosen, and then it is compared with the gold-standard choice. BEAGLE (Jones & Mewhort 2007) was evaluated on a widely-used evaluation setting for semantic models; the synonym section of the TOEFL (Landauer & Dumais 1997). Each item consists of a target word and four alternative words; the task is to select the alternative that is most similar in meaning to the target.

Thus, the task of composing distributional meaning of sequences is evaluated as a multiple selection task. It is difficult to assess the contribution of the model to the final result, given data sparsity issues. Moreover, the human inter-annotation agreement is low (0.40) for the Mitchell & Lapata (2008) dataset.

It is worth exploring combinations of various components of the CDS models. For example, the basic additive model of Mitchell & Lapata (2008) could be used to compose the environmental vectors of BEAGLE. In the same vein, circular convolution could be tested as a CDS model over the bag-of-words feature space.

2.6 Summary

In this chapter, we presented an elaborate literature survey on several issues relevant to multiword expressions and to some useful tools of the field of semantics. In particular, we started with reviewing methods and approaches for recognising multiword expressions and multiword terms; i.e. domain-specific multiword expressions. In succession, we presented an introduction to context distributions, similarity and measures. Then, we presented several methods that attempt to capture the compositionality of multiword expressions and finally, we presented the field of distributional semantics composition and discussed several state-of-the-art models.

Methods for multiword expression recognition were classified as linguistic, statistical or hybrid. Linguistic methods are based on linguistic properties to decide for mul-

tiword expressions, statistical methods use occurrence counts and context while hybrid methods comprise various combinations of linguistic and statistical components and possibly machine learning tools. Statistical methods were divided into unithood-based and termhood-based ones. The former assess the attachment strength of the constituents of a multiword expression candidate, while the latter assess the degree that a candidate multiword expression refers to a concept. The basic research direction emerging is towards evaluation methods that will allow comparing among recognition methods and possibly assess the contribution of each informative source to the result.

Context distributions appear to be the fundamental tool for exploiting the semantics of words and sequences. In section 2.3 we introduced the vector space model, i.e. the standard way of representing context distributions into vectors and discussed several variations concerning feature selection, feature values, etc. Then, the distributional hypothesis, which allows accepting context distributions as a representation of meaning, was presented. Finally, we discussed a multitude of distributional similarity measures and commented on their performance as reported in the literature.

In section 2.4 we review literature that addresses the problem of resolving multiword expressions' compositionality. This review was placed in a separate section, since this problem is identified as being really important for the applications of multiword expressions to other Natural Language Processing tasks. Literature methods were classified in those that are based on comparing context distributions and to those based on substitutions. The latter approach the task indirectly; hypothesising that non-compositional multiword expressions are more rigid than compositional ones. The state-of-the-art lacks methods that take into account the fact that the same multiword expression might have compositional and non-compositional uses. Moreover, not many methods are evaluated on noun phrases consisting of nouns or adjectives and nouns.

In section 2.5 we reviewed several state-of-the-art methods for composing context distributions. Several research directions emerged. Firstly, CDS models are evaluated indirectly; on tasks that are not originally developed for this purpose. Thus, it is unclear which model performs best. Moreover, some CDS models can be further evaluated experimentally as far as their parameters are concerned. Also, the basic additive model and the basic multiplicative model could be combined with a model for selectional preferences.

CHAPTER 3

Analysing Automatic Term Recognition Methods

Executive Summary

Terms are sequences consisting of one or more words that represent domain specific concepts. Multiword terms are the domain specific subset of multiword expressions. Multiword term recognition was introduced long time ago and a plethora of approaches have been proposed. Not only typical collocation extraction methods have been used for this task, but also more sophisticated methods have been proposed, attempting to quantify how likely candidates refer to domain specific concepts. In particular, *unithood-based* approaches come from collocation extraction and measure the attachment strength of candidate term constituents. *Termhood-based* approaches are developed for automatic term extraction and measure the degree that a candidate term refers to a domain specific concept.

Despite the large variety of automatic term recognition methods in the literature, there is significant lack of a common evaluation framework that would allow comparing across different methods. The methods have been evaluated using different, often incompatible evaluation schemes and datasets. In this chapter we propose an evaluation framework which allows comparable experimentation with automatic term recognition methods. Under this framework we evaluate thoroughly a handful of state-of-the-art automatic term recognition methods and we show that *termhood-based* methods achieve

in general superior performance.

Further, we attempt to investigate which is the most successful manner of capturing nestedness; the property of some terms or term candidates to occur as subsequences of other terms or term candidates. Nestedness information comprises a more effective criterion, for distinguishing valid terms from a candidate list, than the strength of association among the constituents of a multiword candidate term. We analysed the *termhood-based* methods, that were evaluated earlier, into their basic components, each of which captures different sources of information and we evaluated these components, separately. Results revealed that the marginal frequency of candidate terms, i.e. the number of independent occurrences of a candidate term is the most effective source for estimating term nestedness, improving automatic term recognition performance.

3.1 Introduction

As discussed in 2.2, *terms* are words or sequences of words that map to concepts of some specific domain of knowledge, usually scientific of technical (Kageura & Umino 1996). A terminology bank is a vocabulary that contains all terms; i.e. all words and sequences which refer to the concepts of a domain. Constructing such a vocabulary is crucial, because it is the starting point for many applications such as machine translation, indexing, knowledge organisation and ontology learning (Kageura & Umino 1996). Manual construction is time-consuming, error-prone and labour-intensive. In many cases, annotators need to be experts of the field and this makes the annotation even more expensive. More importantly, manually constructed terminologies need to be maintained over time; so as to deal with the rapid growth of the number of technical terms. Automatic term recognition is the task of recognising domain-specific terms automatically, and focuses at addressing the above obstacles.

Multiword terms comprise a subclass of multiword expressions. Due to the fact that automatic term extraction can directly serve other scientific domains, research in this field started before research on general text multiword expressions. However, there is large overlap between the methods that have been employed to extract multiword terms and multiword expressions. A detailed survey on proposed methods for extracting terminology and multiword expressions was presented in section 2.2.

Statistical approaches for automatically recognising multiword expressions and terms analyse occurrence statistics of words or sequences. According to Kageura & Umino

(1996), they can be divided into two broad categories: *unithood-based* and *termhood-based* ones. *Unithood* refers to the attachment strength of the constituents of a candidate multiword expression or term. *Termhood* refers to the degree that a candidate term is related to a concept. For example, in an eye-pathology corpus, *"soft contact lens"* is a valid term, which has both high *termhood* and *unithood*. However, its frequently occurring substring *"soft contact"*, has high *unithood* and low *termhood*, since it does not refer to a key domain concept.

Unithood-based methods were initially developed for extracting collocations, which are defined as words that cooccur together more frequently than chance. Methods such as *t-test*, χ^2*-test*, *log-likelihood ratios test* (*LL*) (Dunning 1993) and *pointwise mutual information (PMI)* (Church & Hanks 1990), have been thoroughly evaluated for the task of collocation extraction (Dunning 1993; Evert & Krenn 2001; Dias et al. 2001; Pecina & Schlesinger 2006).

In contrast, *termhood-based* methods, such as statistical barrier (Nakagawa 2000), C-Value and NC-Value (Frantzi et al. 2000) were originally developed for identifying words of sequences that map to domain-specific concepts; i.e. terms. Usually, *termhood-based* methods can as well recognise terms that consist of a single word, however, in this chapter we focus in analysing them as far as their ability to extract multiword terms is concerned.

Unithood-based and *termhood-based* methods have been evaluated using different technical corpora, under different evaluation frameworks, with different set of parameters depending on the domain and test corpus (Frantzi et al. 2000; Dunning 1993; Church & Hanks 1990; Nakagawa 2000). There were some efforts to comparably evaluate small numbers of *unithood-based* methods. For example, Dunning (1993); Evert & Krenn (2001) showed that log-likelihood ratio performs better than t-test, Pearson's chi-square test and pointwise mutual information due to its milder tendency to overestimate rare events. However, there is no published work on comparing *unithood-based* and *termhood-based* methods under a common evaluation framework, i.e. a standard evaluation method and evaluation corpus.

Given that *unithood-based* and *termhood-based* methods capture different types of information, it is still unclear whether the former are able to perform better than the latter methods, such as C-Value (Frantzi et al. 2000) and Statistical Barrier (*SB*) (Nakagawa 2000). This lack of a common evaluation scheme complicates the interpretation

of results. It is unclear which are the strengths and weaknesses of each method, making unmanageable the choice of an appropriate automatic term recognition method as a starting point for other applications.

In this chapter, we define an evaluation framework appropriate for comparable experimentation with methods for extracting multiword terms. Putting the framework into practice, we extensively compare state-of-the-art approaches, so as to identify the strengths and weaknesses of each. Experimenting with part-of-speech patterns (discussed in section 2.2.1.5) showed that accepting adjective and noun sequences as term candidates achieves the best combination of precision and recall compared to patterns that accept less or more parts of speech. This outcome is clearly related to the properties of the employed domain-specific corpora.

We experimentally show that *termhood-based* approaches, which take into consideration the nestedness of a candidate term into others, such as C-Value (section 2.2.2.2.1), NC-Value (section 2.2.2.2.2) and statistical barrier (section 2.2.2.2.3), have in general superior performance over methods which measure the strength of association among the tokens of a multi-word candidate term, such as log-likelihood ratio (section 2.2.2.1.3.4) and pointwise mutual information (section 2.2.2.1.4).

In further experiments we analysed the components of which the termhood-based methods consist and evaluated them separately. The main focus of these experiments is to find which approach for capturing nestedness performs best. We show that the marginal frequency is the most effective source of nestedness information. Marginal frequency is the number of independent occurrences of a term, i.e. the number of occurrences on its own, without being nested within others candidate terms. Marginal frequency clearly improves the performance of automatic term recognition methods, in this evaluation setting.

The rest of the chapter is structured as follows: In section 3.2 we present an overview of the proposed evaluation framework, and explain its functionality in high level. In section 3.3 we briefly review linguistic filtering, based on parts of speech. Section 3.5 presents the proposed evaluation scheme in detail, including the experimental setting (subsection 3.5.1), our experimental results and discussion (subsection 3.5.2). Further experiments towards finding the best way to capture nestedness are presented in subsection 3.5.3. Finally, section 3.6 summarises this chapter.

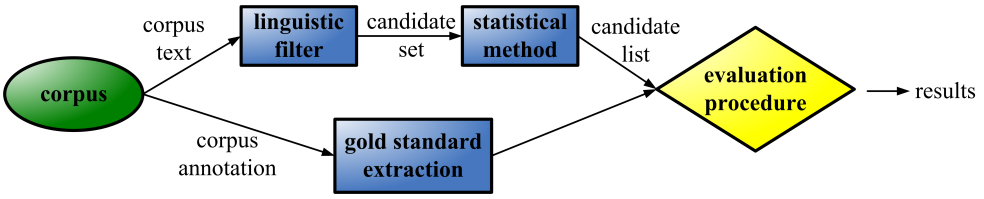

Figure 3.1: Evaluation framework overview

3.2 Evaluation framework overview

Figure 3.1 presents a block diagram of the proposed evaluation framework. The scheme consists of a manually annotated corpus, and an evaluation method which assesses the performance of automatic term recognition methods at a fine-grained scale; i.e. increments of 0.5% of the candidate term ranked list, based on the one proposed in Wermter & Hahn (2004).

The framework utilises a manually annotated corpus of some scientific or technical domain. Any corpus could be used as long as it comprises annotations of multiword terms or multiword expressions. For the experiments of this chapter we used two English domain-specific corpora: the *GENIA* corpus (Gu 2006) and the *PennBioIE* corpus (Kulick et al. 2004). More information about the nature, statistical properties and annotations of these corpora are discussed is section 3.5.1.

Initially, the manual corpus annotations are separated from the corpus text. The annotations are useful to create the gold-standard; a list of word sequences which are domain-specific terms. The gold-standard is used at the final evaluation stage so as to know whether terms accepted by the term recognition system under inspection are actual terms or not.

A linguistic filter is applied on the corpus text to identify candidate terms. Linguistic filters are part of speech patterns which apply as regular expressions on text. They are discussed in section 3.3. In succession, a statistical method is employed to rank the candidates identified by the linguistic filter. The output of this step is a candidate term list in decreasing order of scores.

The evaluation scheme compares the statistical method output list to the gold-standard terms, generated by the corpus annotation. Information retrieval metrics, precision, recall and F-score towards the gold-standard list are computed for portions of the ranked

candidate term list always starting from the top. Portions start from 0.5% of the ranked candidate list and grow incrementally, in 0.5% increments. The evaluation scheme and the visual presentation of results is presented in detail in section 3.5.1.

3.3 Linguistic filters

Initially, automatic term recognition research focused on exploiting the parts of speech of multiword expression constituents. Based on the fact that terms usually consist of some specific parts of speech, different pattern-based models were proposed to identify terms. A number of these models, also known as linguistic filters, was presented and discussed in section 2.2.1.5.

The choice of linguistic filter depends on the language and the domain of the corpus and the application (Frantzi et al. 2000). If the target is to identify terms with high recall a lenient filter should be used, i.e. a filter that verifies many candidates. Towards high precision terms, a strict filter rejecting many lower-quality candidates is appropriate.

In this chapter, four lenient part of speech filters are employed to capture as many terms as possible. Their performance is compared experimentally. The most basic, *Nouns*, accepts sequences of nouns (*N*) only, since terms mainly consist of nouns:

$$N^+ \; N \tag{3.1}$$

The second, *A&N*, applies on sequences consisting of adjectives (*A*) and nouns ending with a noun:

$$(A|N)^+ \; N \tag{3.2}$$

The third linguistic filter, *J&K* was introduced by Justeson & Katz (1995) and has been widely used:

$$(\; (A|N)^+ \mid (A|N)^* \; (NP)? \; (A|N)^* \;) \; N \tag{3.3}$$

Its first part is identical to *A&N*, whereas the second applies on sequences which start with one or more nouns or adjectives, continue with a noun followed by a preposition and end with zero or more nouns or adjectives followed by a noun. Justeson & Katz (1995) used this filter to extract multi-word terms from large text collections in a variety of domains -metallurgy, space engineering and nuclear energy- reporting coverage of 97% or 99% when prepositions are allowed.

Nouns, *A&N* and *J&K* extract sequences of adjectives, prepositions and nouns. However, inspecting *GENIA* annotations revealed that approximately 6% of *GENIA* gold-standard terms contain numbers. To capture those, we extended the linguistic filter of Justeson & Katz (1995) (*J&K*) to *J&K+Ns*, so as to accept numbers (*#*) whenever it accepts nouns or adjectives.

$$(\ (A|N|\#)^+\ |\ (A|N|\#)^*\ (NP)?\ (A|N|\#)^*\ )\ N \tag{3.4}$$

The part of speech filter of equation 3.4 is more lenient than the one of equation 3.3, because it also applies on numbers (*#*) and prepositions (*P*).

3.4 Statistical Automatic Term Recognition Approaches

Approaches to automatic term recognition have been largely based on statistical information. However, most of them include some linguistic part; usually a linguistic filter, to produce a list of candidate terms, as discussed in section 3.3. The statistical part assigns to each candidate term a score indicating how likely the candidate is a valid term. The simplest statistical measure is frequency of occurrence, which captures terms occurring frequently in the corpus. Frequency of occurrence is used as a baseline in our evaluation, since automatic term recognition approaches are expected to achieve superior performance than this simple technique.

Kageura & Umino (1996) define two important concepts relevant to automatic term recognition. The first one, *unithood*, refers to the degree of strength of syntagmatic combinations or collocations. The second, *termhood*, refers to the degree that a candidate term is related to a domain-specific concept. For example, in an eye-pathology corpus, *"soft contact lens"* is a valid term, which has both high *termhood* and *unithood*. However, its frequently occurring substring *"soft contact"*, will have a high *unithood* and a low *termhood*, since it does not refer to a key domain concept.

In the experimental part of this chapter, the *termhood-based* methods C-Value (section 2.2.2.2.1), NC-Value (section 2.2.2.2.2) and statistical barrier (section 2.2.2.2.3), and the *unithood-based* methods log-likelihood ratio (section 2.2.2.1.3.4) and pointwise mutual information (section 2.2.2.1.4) are evaluated under the proposed framework, and their performance is compared.

3.5 Evaluation

In this section, we discuss the setting of the proposed evaluation framework under which the previously discussed termhood-based and unithood-based automatic term recognition methods are evaluated. The experimental setting subsection (3.5.1) contains details about the chosen corpora and their characteristics, a presentation of the employed evaluation metrics and an overview of the experiments executed. In succession, subsection 3.5.2 shows a number of representative experimental results and discusses the performance of automatic term recognition methods under evaluation. Finally, in subsection 3.5.3 we present an analysis of the termhood-based methods into their components, which capture different sources of information about candidate terms. Results of separate evaluation of these components under the proposed framework are presented and discussed to identify the most successful way of capturing nestedness.

3.5.1 Experimental setting

GENIA[1] and *PennBioIE*[2] were chosen for experimentation. They are widely used corpora of the biomedical domain, freely available for research purposes. Both corpora consist of abstracts from *MEDLINE*, a very large collection of biomedical articles, and terms are manually annotated. *GENIA* (Gu 2006) contains $2,000$ abstracts. *PennBioIE* (Kulick et al. 2004) contains $2,257$ coming from two biomedical domains: 1100 abstracts are about inhibition of the cytochrome P450 family of enzymes, and 1157 abstracts are about molecular genetics of cancer. In *PennBioIE*, quantitative values and units are separately annotated, but these annotations were ignored for this evaluation. Table 3.1 shows statistics of these two corpora that indicate their size in sentences, tokens and terms. As term types, it reports the number of different categories in which the terms are classified. For *GENIA*, these are classes of the *GENIA ontology*, while for *PennBioIE* they defined in the annotation guidelines of the *UPenn Biomedical Information Extraction Project*.

Figure 3.5.1 shows an example sentence from the *GENIA* corpus. Term annotation uses two *xml* attributes in parallel, the lexical attribute (lex), and the semantics attribute (sem). Annotation attributes are presented in green, while their values in red colour. The actual *GENIA* text is printed in blue.

[1] www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi
[2] bioIE.ldc.upenn.edu

	GENIA	*PennBioIE*
sentences	18,546	32,692
tokens	454,848	712,551
terms	97,876	76,535
distinct terms	35,947	13,759
term types	36	22

Table 3.1: *GENIA* and *PennBioIE* corpus statistics

In contrast to *PennBioIE*, *GENIA* annotation terms are not part of the text, but of separate *xml* attributes. Thus, *GENIA* gold-standard is created by collecting these *xml* values and cleaning most non-alphanumerical characters. We observed that in a few cases annotation tokens are not lemmatised (e.g. "activators of transcription", "activating function") or erroneous (e.g. "latent proviru"). However, we hypothesise that a corpus with low level of noise is acceptable for our purposes. The text of both *GENIA* and *PennBioIE* was similarly cleaned. Then, both corpora were tokenised and part of speech tagged using the *GENIA* tagger[3].

Tables 3.2 and 3.3 show for various term lengths gold-standard term counts and candidate term counts separately for the four linguistic filters introduced in section 3.3, i.e. Noun (N), Adjective Noun (A&N), Justeson and Katz (J&K) or Justeson and Katz + Numbers (J&K+Ns). The first columns of tables 3.2 and 3.3 show gold-standard term counts of *GENIA* and *PennBioIE*, respectively. The following columns present candidate term counts, identified by each linguistic filter. Linguistic filters are shown in order of descending strictness. For example, the *A&N* filter identified far fewer candidates than the *J&K*. However, even the most strict filter, *Nouns*, creates more candidate terms than the valid ones. Note that, for each column, the count of candidates of any length (in the first row of tables 3.2 and 3.3) is not equal to the sum of all N-grams, because candidates of any length include sequences up to 12 tokens long.

The standard information retrieval evaluation metrics *precision* and *recall* (Mikheev, Moens & Grover Mikheev et al.; Radev et al. 2003) were used for evaluating automatic

[3] www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger

```
<sentence>
    <cons lex="IL-2_gene_expression" sem="G#other_name">
        <cons lex="IL-2_gene" sem="G#DNA_domain_or_region">
            IL-2 gene
        </cons>
        expression
    </cons>
    and
    <cons lex="NF-kappa_B_activation" sem="G#other_name">
        <cons lex="NF-kappa_B" sem="G#protein_molecule">
            NF-kappa B
        </cons>
        activation
    </cons>
    through
    <cons lex="CD28" sem="G#protein_molecule">
        CD28
    </cons>
    requires reactive oxygen production by
    <cons lex="5-lipoxygenase" sem="G#protein_molecule">
        5-lipoxygenase
    </cons>
    .
</sentence>
```

Figure 3.2: Example *GENIA* sentence

term recognition statistical methods:

$$\text{precision} = \frac{\text{\# correctly identified terms}}{\text{\# identified terms}} \tag{3.5}$$

$$\text{recall} = \frac{\text{\# correctly identified terms}}{\text{\# gold-standard terms}} \tag{3.6}$$

F-Score is defined as the weighted harmonic mean of precision and recall:

$$\text{F-Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3.7}$$

	GENIA				
Length	GS	N	A&N	J&K	J&K+Ns
Any	28,142	29,751	69,457	85,978	138,251
2-grams	12,654	17,103	33,021	33,021	36,866
3-grams	9,051	8,813	21,401	28,071	37,146
4-grams	3,839	3,199	9,356	15,204	29,803
5-grams	1,559	1,020	3,699	6,339	18,099
6-grams	606	297	1,317	2,239	9,005

Table 3.2: Gold-standard (*GS*) term counts and candidate term counts per linguistic filter and term length in the *GENIA* corpus.

	PennBioIE				
Length	GS	N	A&N	J&K	J&K+Ns
Any	7,447	46,519	80,205	99,194	178,939
2-grams	4,034	28,489	44,072	44,072	58,086
3-grams	1,820	11,421	22,530	31,930	49,570
4-grams	821	4,157	8,629	14,945	35,746
5-grams	388	1,486	3,070	5,447	20,019
6-grams	207	694	1,172	1,822	9,105

Table 3.3: Gold-standard (*GS*) term counts and candidate term counts per linguistic filter and term length in the *PennBioIE* corpus.

F-Score favours for roughly equal values of precision and recall, which is meaningful for the automatic term recognition task.

Tables 3.4 and 3.5 show precision and recall for every linguistic filter for candidates of any length and N-grams for both corpora. A first observation is that all liguistic filters achieve in these corpora much lower recall than previously reported for the *J&K* filter. Justeson & Katz (1995) reported a recall of 99% in a large text collection in a variety of domains. Secondly, we observe that the less strict a filter is, the higher the recall and the lower the precision. *A&N* achieve the best compromise between recall and precision. Statistical methods for automatic term recognition re-rank the list of candidates, aiming to output the real terms higher than non-term candidates. Thus, considering the whole list, the performance of all statistical methods is the same as the corresponding value of table 3.4 or 3.5, with respect to the employed corpus, candidate term length and linguistic

	GENIA							
	Nouns		A&N		J&K		J&K+Ns	
Length	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*
Any	35.4	33.5	80.2	32.5	80.2	26.3	85.4	17.4
2-grams	48.1	35.6	88.0	33.7	88.0	33.7	90.6	31.1
3-grams	31.9	32.8	80.4	34.0	80.5	25.9	84.5	20.6
4-grams	21.3	25.5	67.0	28.7	70.4	17.8	78.9	10.2
5-grams	14.9	22.7	63.8	26.9	64.2	15.8	77.0	6.6
6-grams	9.2	18.9	54.5	25.1	54.5	14.7	71.0	4.8

Table 3.4: Recall (*R*) and precision (*P*) percentages (%) per linguistic filter and length of candidate term in the *GENIA* corpus.

	PennBioIE							
	Nouns		A&N		J&K		J&K+Ns	
Length	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*
Any	37.2	6.0	63.1	5.9	63.7	4.8	76.1	3.2
2-grams	52.6	7.5	78.1	7.1	78.0	7.1	90.6	6.3
3-grams	26.8	4.3	60.7	4.9	61.6	3.5	73.1	2.7
4-grams	15.8	3.1	42.8	4.1	45.2	2.5	56.5	1.3
5-grams	4.7	1.2	17.1	2.2	18.7	1.3	38.3	0.7
6-grams	3.9	1.3	13.0	2.3	13.5	1.5	24.6	0.6

Table 3.5: Recall (*R*) and precision (*P*) percentages (%) per linguistic filter and length of candidate term in the *PennBioIE* corpus.

filter.

As discussed in subsection 2.2.2.1.3.4, the *log-likelihood ratios* method can only be applied separately for sequences of a specific length. We implemented the extended log-likelihood ratios algorithm for N-grams, $N \in [2, 6]$. There are only 433 *GENIA* gold-standard terms and 177 *PennBioIE* gold-standard terms longer than 6 tokens, very few to experiment with. The results of the log-likelihood ratios algorithm for different values of N are not comparable to each other. Thus, we set separate experiments up for each value of $N \in [2, 6]$.

For example, for 2-grams we first apply a linguistic filter to identify candidates of which we keep 2-grams only. Then, 2-grams are re-ranked according to one of the

Length
Any
2-grams
3-grams
4-grams
5-grams
6-grams

Linguistic filter
Nouns
A&N
J&K
J&K+Ns

Statistical approach
C-Value
NC-Value
PMI (N-grams only)
LL (N-grams only)
SB (Nouns and A&J only)

Table 3.6: Executed experiments

implemented statistical methods: pointwise mutual information, log-likelihood ratio, C-value, NC-Value or statistical barrier. Evaluation is performed towards the 2-gram gold-standard terms. Experiments for the other values of N were set up identically.

Except for N-grams, we ran experiments taking into account sequences of any length, equal or higher than 2. For each one, candidate terms are identified using one of our four linguistic filters: *Nouns*, *A&N*, *J&K* or *J&K+Ns*. Then, one of *C-Value*, *NC-Value* or statistical barrier re-ranking method is applied. The results are compared against the whole gold-standard term set. Note that the statistical barrier method makes sense only when it follows the *Nouns* or the *A&N* linguistic filter. Table 3.6 summarises all executed experiments, referring to the combination of length of candidate terms, filtering and statistical approach used.

The NC-Value algorithm takes as input a list of candidates, ranked by the C-value algorithm and is subject to two parameters: the percentage of the list, starting from the top, that it will take into account to identify context terms and the size of the context window. We experimented using values 5%, 7.5% and 10% for the former one and 2, 4, 6, 8, 10 for the latter.

To visualise the results, we employed an approach similar to the one defined in Wermter & Hahn (2004). Recall and precision values were calculated at 0.5% increments of the candidate list and plotted on graphs, such as figures 3.3 and 3.4. For each increment on the list, precision refers to the ratio of true positives over the overall number of candidates and recall refers to the ratio of true positives over the number of gold-standard terms. The x-axis shows the percentage of the list taken into account. Frequency of occurrence is used as baseline.

Intuitively, the precision curve of a bad performing method would be relatively ho-

rizontal indicating that the true positives were dispersed uniformly throughout the list rather than pushed towards the top. Contrarily, the precision curve of a well-performing method would be 100% until the percentage point at which all gold-standard terms would have been retrieved, where a sharp decrease would occur (McInnes 2004).

3.5.2 Results

Figure 3.3 shows the 2-gram precision and recall curves of NC-Value for all 15 parameter combinations using the *J&K* linguistic filter on *GENIA* corpus. We observe that different combinations do not significantly alter the results. This behaviour remains the same for all linguistic filters and for all term lengths. Interestingly, for all the above experiments the performance of C-Value and NC-Value is almost identical, both for *GENIA* and *PennBioIE*.

Figure 3.4 shows the F-Score curves for 3-gram candidate terms of *GENIA* and *PennBioIE* as identified by the *Nouns* linguistic filter. We observe that *termhood-based* methods outperform *unithood-based* ones. Statistical barrier, C-Value and NC-Value perform similarly with statistical barrier having a slightly better F-Score on *GENIA*, where statistical barrier achieves superior performance followed by C-Value and NC-Value, for all 0.5% increments of the candidate list up to 60%. After that point their performance is similar. The ranking of automatic term recognition methods remains the same as in figure 3.4 for any N-gram using both the *Nouns* and the *A&N* linguistic filter, on both corpora.

Pointwise mutual information curves are below the baseline on both corpora. On the contrary, log-likelihood ratio outperforms the baseline of frequency of occurrence on *PennBioIE* but not on *GENIA*. The main reason for pointwise mutual information performing worse than log-likelihood ratio is that the former overestimates rare events, which seem to dominate the candidate lists created by any linguistic filter. For example, according to table 3.2 *A&N* identifies $69,457$ *GENIA* candidate terms. $52,998$ occur only once, and $16,459$ twice. Log-likelihood ratio also overestimates rare events, but its overall behaviour is better than pointwise mutual information (Manning & Schutze 1999).

Figure 3.5 shows the precision and recall curves for 3-gram candidate terms of *GENIA* as identified using by the *J&K+Ns* linguistic filter. The performance for N-gram candidate terms as identified by *J&K* and *J&K+Ns* demonstrate the following trends: On

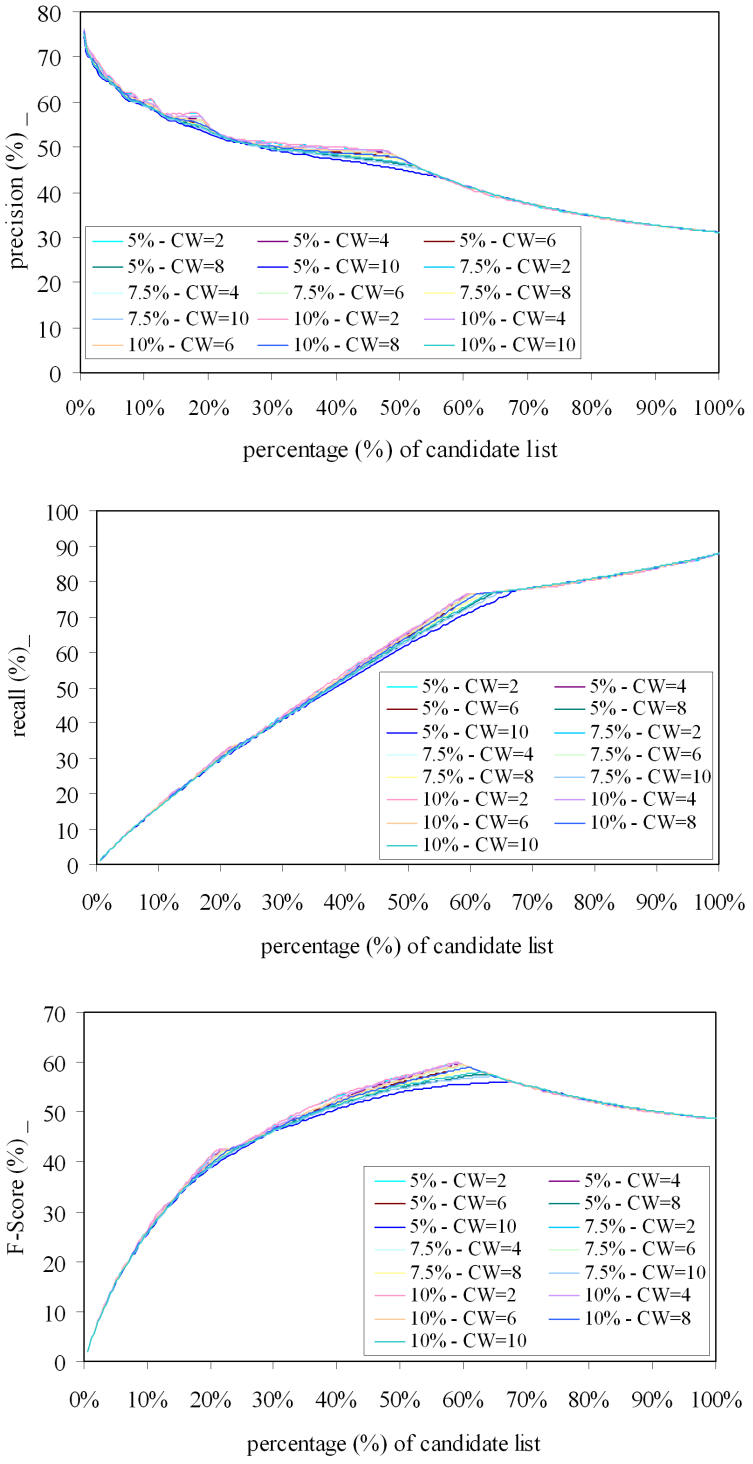

Figure 3.3: NC-Value results on *GENIA* 2-grams. J&K filter. Precision, recall and F-Score.

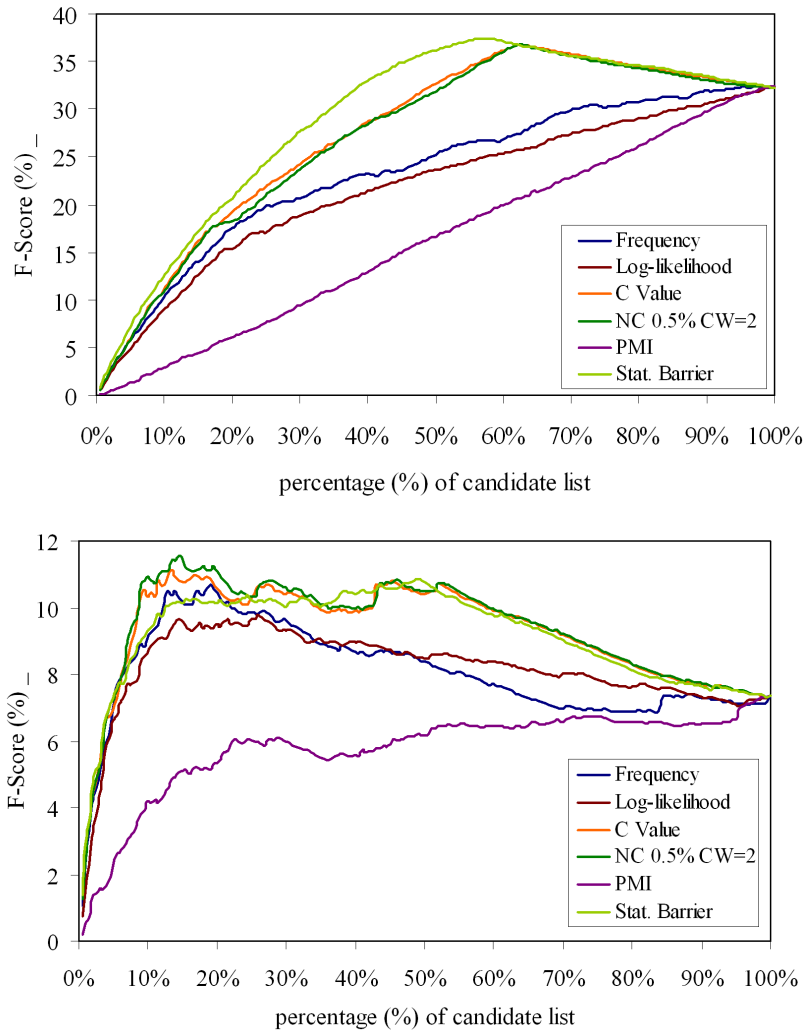

Figure 3.4: Statistical methods on *GENIA* and *PennBioIE* 3-grams. Noun filter. F-Score.

GENIA the highest performance is achieved by statistical barrier, C-value and NC-Value methods throughout the plots. The remaining methods in order of decreasing F-Score are: frequency of occurrence, log-likelihood ratio and pointwise mutual information. The bigger N is, the closer raw frequency, log-likelihood ratio and pointwise mutual information curves are to each other.

In *PennBioIE*, the performance differences between frequency of occurrence, log-likelihood ratio, C-value, NC-Value and statistical barrier are insignificant, while pointwise mutual information clearly performs worse. In this corpus we observe that *termhood-*

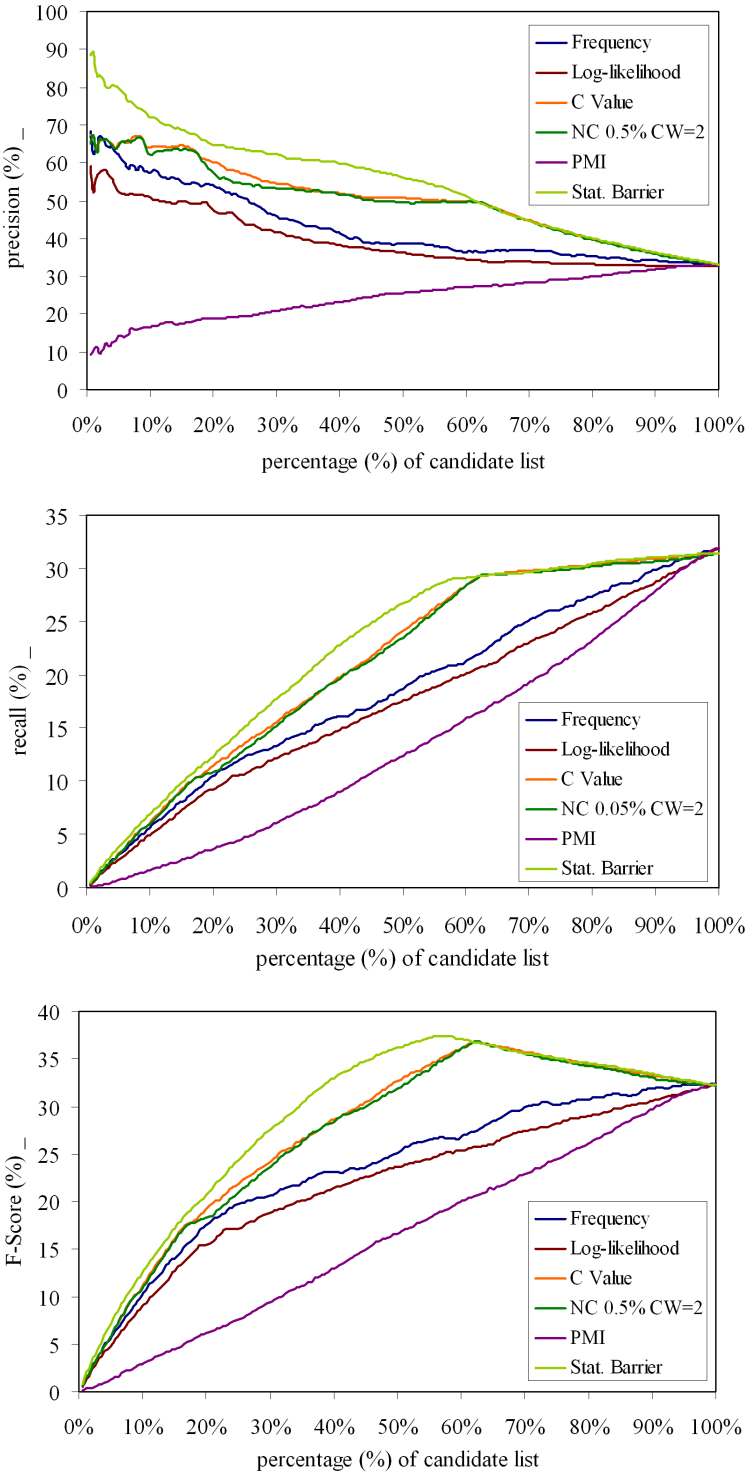

Figure 3.5: Statistical methods on *GENIA* 3-grams. *J&K+Ns* filter. Precision, recall and F-Score.

based methods have a comparable performance with the baseline. 6-gram results follow the same trends in general, but they are not very reliable due to the small number of candidates.

On both corpora for candidates of any length identified by *Nouns* and *A&N*, statistical barrier, C-Value and NC-Value methods exceed the baseline of frequency of occurrence, achieving similar levels of performance. Using the *J&K* and *J&K+Ns*, the performances of C-Value, NC-Value and frequency of occurrence are similar for increments up to 10% of the candidate list for both corpora. In *GENIA*, for increments between 10% and 30%, frequency of occurrence performs better than C-Value and NC-Value. The same behaviour happens in *PennBioIE* for increments between 10% and 50%. After 30% in *GENIA* and 50% in *PennBioIE*, C-Value and NC-Value perform better than frequency of occurrence.

3.5.3 Further Experiments and Results

In the previous section, results show that *termhood-based* methods re-rank the candidate list equally well or better than *unithood-based* methods, irrespective of the length of the candidate terms and the linguistic filter used. A possible reason is that *unithood-based* methods measure the strength of attachment of the candidate term constituents, in effect assigning high scores to candidate terms, which might not refer to domain specific concepts. For example, in *GENIA*, *"allergic inflammatory"*, substring of the term *"allergic inflammatory disease"*, occurs at least equally often as the term, although the former is not a term itself. In contrast, *unithood-based* methods do not consider that although substrings of multiword terms might occur often, they might not be actual terms.

The only setting in which a *unithood-based* method, in particular log-likelihood ratio, performed equally well to the *termhood-based* methods was when using *J&K* or *J&K+Ns* to extract N-gram candidates from *PennBioIE*. A possible explanation for this peculiarity is the limited amount of nestedness information in *PennBioIE*, which degrades the performance of *termhood-based* approaches. Particularly for 3-grams the average nested frequency in *PennBioIE* is 1.03, while in *GENIA* is 1.16. Note that *PennBioIE* is almost double the size of *GENIA*.

C-Value and NC-Value exploit nestedness information, in the sense that the more often a candidate appears as nested, the less likely it is a valid term. Statistical barrier considers this information indirectly; through marginal frequency counts. NC-Value

attempts to improve C-Value ranking, by exploiting contextual information. However, NC-Value appears unsuccessful under our evaluation scheme. To investigate this, we executed an experiment, in which we adjusted the interpolation constant of equation 2.38 to assess the contribution of the context factor (CF) only.

Precision curves are almost uniform across most of the plot. Our results appear in apparent contradiction with the results of Frantzi et al. (2000). It is reported that NC-value successfully rearranges the term candidate list and leads to a 5% increase in precision close to the top of the list. Frantzi et al. (2000) used for their experiments a corpus consisting of eye-pathology medical records. Our experiments showed that NC-Value achieves exactly the same precision as C-value. A possible reason for this outcome might be the nature of the corpora, in which contextual information are possibly not strong enough.

Statistical barrier exploits two sources of information: firstly, *GM* (equation 2.39), assumes that complex terms consist of existing simple terms. Secondly, *MF* (equation 2.40), refers to the marginal frequency counts. To evaluate the contribution of each, we executed two experiments, which re-rank the candidate term list taking into account *GM* and *MF*, separately.

The resulting precision, recall and F-Score plots on *GENIA* are shown in figure 3.6. Interestingly, precision of *GM* is roughly uniform. This means that it contributes negatively to statistical barrier throughout the plot. On the contrary, marginal frequency successfully redistributes candidates towards the top of the list. Thus, the corresponding precision curve is higher than the curve of statistical barrier in the x-axis interval [0%, 30%]. The same experiments on *PennBioIE* verified these results.

Our results contradict with the results of Nakagawa (2000), who used the NTCIR1 TMREC test collection for their experiments. They compared *GM* to *SB* and concluded that it performs slightly worse, especially close to the top of the term candidate list. For our data, this difference appears to be large. However, no experimental comparison between *MF* and either *GM* or *SB* is reported.

The C-Value algorithm hypothesises that the more frequently a candidate term appears as nested the less likely it is a valid term. It also considers that a candidate occurring as substring of many distinct candidates is possibly a valid term itself. Hence, C-Value calculates a weighted version of marginal frequency, which we call here modi-

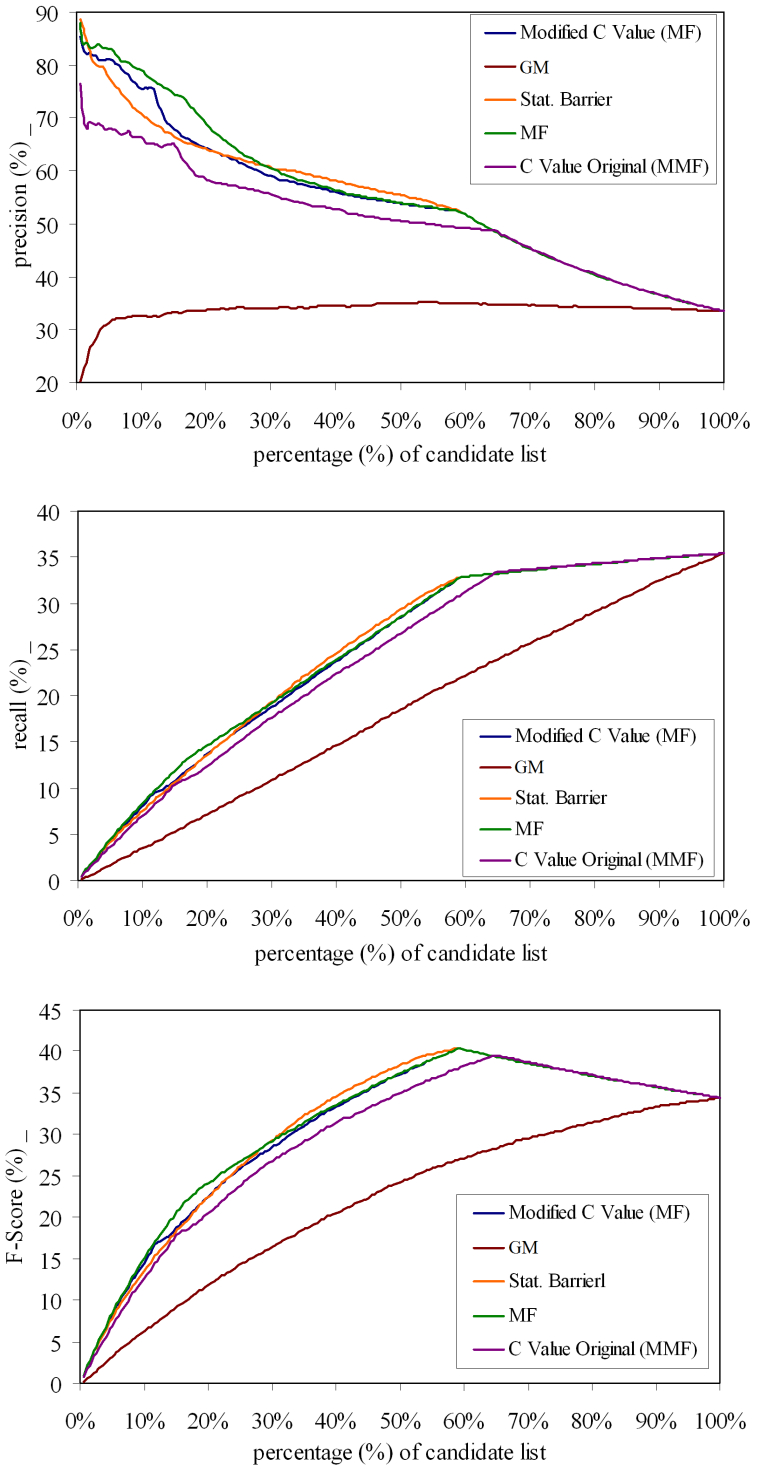

Figure 3.6: *GENIA* sequences of any length, *Nouns* filter, various methods, Precision, Recall and F-Score.

fied marginal frequency. Recall the definition of C-Value score:

$$C\text{-}value(ct) = \begin{cases} \log_2(|ct|) \times [f(ct) - NST(ct)], & \text{if } ct \text{ is nested} \\ \log_2(|ct|) \times f(ct), & \text{otherwise} \end{cases}$$

$NST(ct)$ is the ratio of the frequency of the candidate as nested over the number of distinct terms, in which it appears nested. In the previous equation, modified marginal frequency (*MMF*) is defined as:

$$MMF(ct) = f(ct) - NST(ct) \tag{3.8}$$

To examine the effect of modified marginal frequency in C-Value, we replaced modified marginal frequency in the C-Value equation with marginal frequency. Results show that the new version of C-Value, using marginal frequency, outperforms the original C-Value. Thus, marginal frequency captures nestedness better than modified marginal frequency, in this evaluation setting. However, figure 3.6 shows that marginal frequency outperforms even this modified version of C-Value, for increments up to 25% of the candidate list.

3.6 Summary

In this chapter we proposed an evaluation framework for evaluating automatic term recognition methods under common terms, so as to be able to compare their performance. The framework consists of the evaluation corpora, chosen evaluation metric and visualisation of the results. Two freely available, widely used, manually annotated corpora of the biomedical domain were used: *GENIA* and *PennBioIE*. To assess performance the standard information retrieval metrics were adopted: precision, recall and F-Score. Results are plotted at a scale of 0.5% increments of the candidate term ranked list, the output of the statistical method under evaluation.

Using the proposed evaluation framework, we thoroughly evaluated a handful of state-of-the-art linguistic filters and statistical automatic term recognition methods. We discussed and compared the results to identify the strengths and weaknesses of each method and experimentally showed that *termhood-based* approaches, which are based on nestedness information, outperform *unithood-based* methods, which measure the

strength of association among the constituents of a multiword candidate term.

Secondly, further experiments which evaluated separately each component of the *termhood-based* methods showed that marginal frequency comprises the most effective source of nestedness information for automatic term recognition. Marginal frequency is defined as the number of independent occurrences of a term, i.e. not nested in other candidate terms. Marginal frequency clearly improves the performance of C-Value and statistical barrier. Even more interestingly, it performs by itself better that these more sophisticated *termhood-based* methods, under the current evaluation framework.

CHAPTER 4

Resolving Compositionality

Executive Summary

In this chapter we propose a new unsupervised approach to decide compositionality of some classes of multiword expressions: compound nominals, proper names and adjective-noun constructions, in English. The approach attempts to exploit the contribution of sense induction towards this purpose. Partitioning the context distribution of a multiword expression and its semantic head before comparing them is expected to enhance the ability of judging it as compositional or not. The reason is that in many cases, a multiword expression is ambiguous; it sometimes occurs with compositional senses while sometimes with non-compositional ones.

The evaluation set was extracted from *WordNet*. A semi-supervised algorithm that minimises human effort is proposed to add compositionality judgements to the multiword expressions test set. Results showed that sense induction can assist in identifying compositional multiword expressions when estimating parameter values manually.

In the second part of this chapter, we propose an unsupervised scheme for estimating the parameters of graph-based sense induction systems. It exploits several measures which assess how well connected are the induced clusters that represent the senses of a multiword expression or semantic head. These clusters correspond to partitions of the corresponding context distributions. Experiments show that graph connectivity measures

are able to identify useful differences regarding the degree of connectivity of induced clusters for different parameter combinations. Hence, they be can successfully applied since they achieve comparable performance with manually choosing the best performing parameters.

4.1 Introduction

This chapter presents our research in resolving the compositionality of multiword expressions. The notion of compositionality was discussed in section 1.1. It refers to the degree to which the meaning of a multiword expression can be predicted by combining the meanings of its components. For example, *black maria* is non-compositional, since it does not refer to a person called *Maria* whose skin is *black*. *Black maria* refers to a special police van for transporting prisoners and also to the first American movie production studio in West Orange, New Jersey. In contrast, *parking brake* is compositional, since it is a *brake* that is in use while a vehicle is *parked*.

Semantic compositionality is continuous (Baldwin 2006) and, thus exhibits various levels. For example *fish finger* is a non-compositional expression, because it does not refer to the *finger* of a *fish*. However, it refers to *finger*-shaped food which contains *fish*. One can observe that the semantics of *black maria* are more dissimilar to the semantics of its components than the semantics of *fish finger* to the semantics of its components. Another example is *high jump*, which is also non-compositional because it refers to the Olympic sport and not to any case of *jumping high*. However, the sport includes exactly what the expression says; *jumping high*. Thus, one can argue that *high jump* is less non-compositional than *fish finger* and much less non-compositional than *black maria*.

In section 2.4 of the literature survey of this thesis, we reviewed a number of approaches attempting to decide compositionality of various types of multiword expressions. In general, methods can be classified as context distribution-based or substitution-based. Context distribution-based methods build on the distributional hypothesis and on the definition of compositionality. They compare a combination of the context distributions of the constituents of the target multiword expression to the context distribution of the target multiword expression itself. Substitution-based methods address compositionality in an indirect way; they substitute one or more components of a multiword expression candidates and decide how syntactically rigid is the candidate by looking at how probable the result of the substitution is. High syntactic rigidity characterises insti-

tutionalised expressions and substitution-based methods hypothesise that most of them are non-compositional, as well. This assumption has been largely criticised in the literature (Bannard et al. 2003).

We identified that only very few methods take into account the fact that some multiword expressions appear both as compositional and non-compositional. For example, *green light* is non-compositional in most cases, because it refers to acceptance, a signal to continue or approval. However, in the sentence "The traffic lights were not working because a technician was replacing the *green light* bulb" the multiword expression appears as compositional. Fazly & Stevenson (2007) and Fazly et al. (2009) support a per-instance view to the issue of compositionality of multiword expressions. They hypothesise that non-compositional occurrences are less syntactically flexible than compositional ones and employ the canonical forms of Fazly & Stevenson (2006) to distinguish compositional versus non-compositional instances.

In this chapter, we inspect the extent to which sense induction can serve as a component to decide multiword expression compositionality in a unsupervised manner. In particular, sense induction is employed to partition the context distribution of a target multiword expressions and its governing component.

We propose a novel unsupervised approach that compares the major senses of a multiword expression and its semantic head using context distributions and distributional similarity measures to decide multiword expression compositionality. The senses under comparison are induced by a graph-based sense induction system. The method partitions the context space and only uses the major senses, filtering out minor senses.

Secondly, we propose an unsupervised manner to estimate the free parameters of graph-based sense induction systems. It explores a number of graph connectivity measures, which are able to assess the quality of the induced senses. Given a parameter setting and the associated induced clustering solution, each induced cluster corresponds to a subgraph of the original unclustered graph. A graph connectivity measure scores each cluster by evaluating the degree of connectivity of its corresponding subgraph. Each clustering solution is then assigned the average of the scores of its clusters. Finally, the highest scoring solution is selected, without any need of manually tagged data. The above unsupervised parameter tuning scheme is then applied to estimate the parameters of the proposed system for resolving compositionality.

The proposed method is evaluated on compound nominals, proper names and adjective-

noun constructions. We propose a semi-supervised approach for distinguishing compositional versus non-compositional multiword expressions extracted from *WordNet*, to decrease annotation cost.

The results show that, firstly, sense induction can assist in identifying compositional multiword expressions. Secondly, unsupervised parameter tuning, employing graph connectivity measures, results in accuracy that is comparable to the best manually selected combination of parameters.

The remaining of this chapter is structured as follows: In section 4.2 the proposed method is described in high level. It is followed by subsections 4.2.1, 4.2.2, 4.2.3, 4.2.4 each of which describes one of the components of the system. Subsection 4.2.2 describes the employed sense induction component, although the proposed method is able to cooperate with any other sense induction component. The only restriction is that is should be based on graphs, so that the unsupervised parameter scheme of section 4.6 can be applied. Section 4.3 discusses how the multiword expression test set was constructed. It includes a semi-supervised algorithm able to construct a set with compositionality annotation with minimum human intervention. In section 4.4 we describe all evaluation setting details and in section 4.5 we present and discuss the evaluation results, when estimating parameter values manually. Section 4.6 present an unsupervised parameter tuning scheme based on graph connectivity measures and section 4.7 shows the results of applying the scheme for the current task. Section 4.10 summarises the chapter.

4.2 Sense induction for resolving compositionality

Let us consider an example multiword expression to aid explaining the functionality of the proposed system in high level. The non-compositional multiword expression "red carpet" mainly refers to a strip of red carpeting laid down for dignitaries to walk on. However, it is possible to encounter instances of "red carpet" referring to its minor sense; any carpet of red colour. The context distribution of a multiword expression, created over the contexts of all its instances in a corpus, is the sum of the context distribution of its senses and represents them all together. The context distribution of "red carpet" represents both the non-compositional and compositional senses, concurrently.

According to most methods found in the literature, one can decide compositionality of a multiword expression by comparing its context distribution to a combination of the context distribution of its components. Given that a compound nominal, proper name

or adjective-noun construction is compositional, its semantics can be approximated by the semantics of its semantic head. Words of the multiword expression other than the semantic head most likely modify the semantics of the head. Combining the above, multiword expression compositionality can be decided by comparing the context distribution of the multiword expression to the context distribution of its semantic head.

Returning to the example, the context distribution of "red carpet" should be compared to the context distribution of "carpet". Given that the former represents both the non-compositional and compositional senses of the multiword expression together, the comparison will result in a lowest value than it would if the context distribution of "red carpet" was representing its non-compositional sense, only. In other words, if a multiword expression occurs both with non-compositional and compositional senses, then the context distributional methods to decide compositionality are likely to fail. The more frequent the compositional sense of the multiword expression is, the more likely distributional methods are expected to fail.

To address this problem, we propose to partition the context distribution of the multiword expressions and of its semantic head before comparing them. In contrast to Fazly & Stevenson (2007) and Fazly et al. (2009), we choose to address the problem of resolving compositionality on a type-basis; i.e. we intend to develop a system that will be able to decide compositionality of a given multiword expression independently of context. Thus, we assume that when people listen to a multiword expression they decide whether it is compositional or not according to its most frequent sense (major sense). For example, we assume that the question "Is *red carpet* compositional or not?" will in general be answered positively, since its major sense is non compositional.

Consequently, our approach partitions the context distributions of the multiword expression and its semantic head and in succession compared their major induced senses to decide whether the multiword expression is compositional or not. The more diverse the major induced senses are, the more possibly the multiword expression is non-compositional. Figure 4.1 shows an overview of the proposed system.

The proposed algorithm consists of 4 steps:

4.2. 1: Corpora collection and preprocessing

4.2. 2: Sense induction of the multiword expression and its semantic head

4.2. 3: Comparison of their major induced senses

4.2. 4: Determining compositionality of the target multiword expression

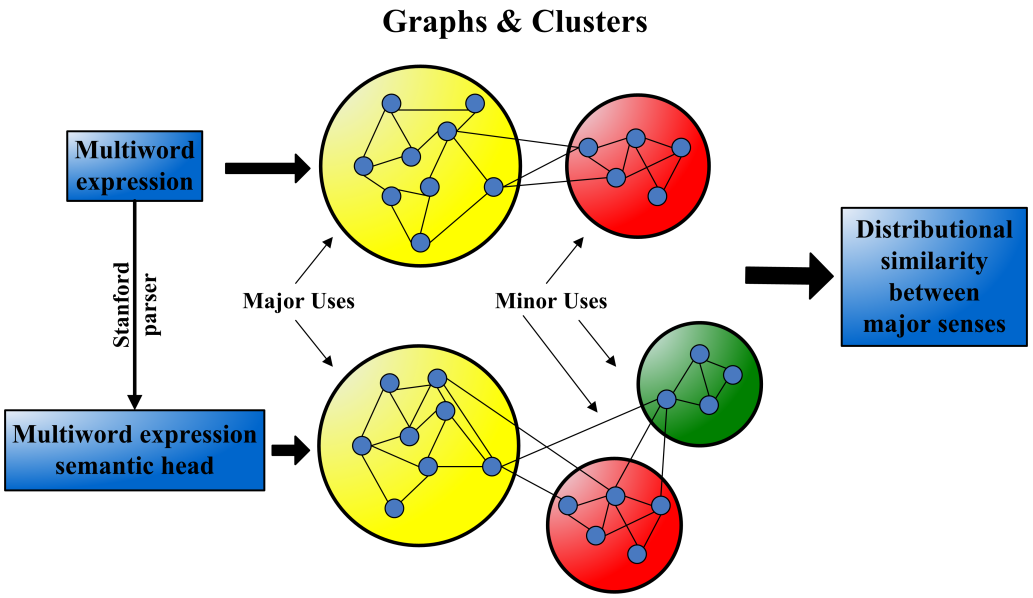

Figure 4.1: System overview

4.2.1 Corpora collection and preprocessing

The proposed approach receives as input a multiword expression, for example "red carpet". The input multiword expression is passed to *Stanford Parser* (Klein & Manning 2003). Its dependency output is inspected to locate the multiword expression's semantic head. This semantic head identification is shown on the left side of figure 4.1.

In most cases, locating the semantic head of a *compound nominal* or a *proper name* in English is easy, since it is usually the last word. However, there are cases that this is not true; for example in the multiword expression "prince Albert". Moreover, this is not true in other languages, such as in French.

Two different corpora are collected; one containing instances of the multiword expression and one containing instances of its semantic head. Each corpus consists of webtext snippets of length 15 to 200 tokens. Most search-engines return 1000 urls per query, which is some cases provides insufficient data for the task. Thus, we employed WordNet to expand the target multiword expression or its semantic head into a larger set of queries.

Given a multiword expression, a set of queries can be created as follows: All syn-

onyms of the multiword expression in WordNet are collected. The multiword expression is paired with each synonym to create a new queries. For example, the synonyms of "red carpet" are "rug", "carpet" and "carpeting", creating the following queries:

1: "red carpet"
2: "red carpet" rug
3: "red carpet" carpet
4: "red carpet" carpeting

In succession, an html-parser is employed to extract web-text snippets of length 15 to 200 tokens from each url returned by *Yahoo!*. The union of all snippets produces the multiword expression corpus. The corpus for a semantic head is created equivalently.

To keep the computational time reasonable, only $3,000$ snippets of medium length are kept from each corpus. The corpora corresponding to each multiword expression and semantic head pair are part of speech tagged using the *GENIA* tagger. In common with Agirre et al. (2006), only nouns are kept and lemmatised, since they are more discriminative than verbs adjectives and adverbs, which may appear in many different contexts.

4.2.2 Sense Induction

The senses of a word or a multiword expression are traditionally represented as a fixed-list of definitions of a manually constructed lexical database. The fixed-list of senses paradigm has several disadvantages. Firstly, lexical databases often contain general definitions and miss many domain specific senses (Agirre et al. 2001). Secondly, they suffer from the lack of explicit semantic and topical relations between concepts (Agirre et al. 2001). Thirdly, they often do not reflect the exact content of the context in which the target word or expression appears (Veronis 2004). Automatic sense induction aims to overcome these limitations of hand-constructed lexicons.

Practically, any unsupervised approach able to derive a set of senses of a given expression can be used as a component of our algorithm. Sense induction methods can be broadly divided into vector-space models and graph based models. In vector-space model *SI*, each context of a target word is represented as a feature vector. The dimensions of feature vectors are usually cooccurring words and the corresponding values are the cooccurrence frequencies. (For more details on this issue see subsection 2.3.1). Context vectors are clustered and the resulting clusters represent the induced senses.

Recently, graph-based methods have been used for sense induction (Dorow & Widdows 2003; Veronis 2004; Agirre & Soroa 2007b). Typically, graph-based approaches represent each word cooccurring with the target expression, within a pre-specified window, as a vertex. Two vertices are connected via an edge if they cooccur in one or more contexts of the target expression. Once the cooccurrence graph has been constructed, different graph clustering algorithms are applied to induce the senses. Each cluster (induced sense) consists of a set of words that are semantically related to the particular sense (Veronis 2004).

Sense induction methods are evaluated under the *SemEval-2007* framework (Agirre & Soroa 2007a). The potential advantage of graph-based methods is that they can combine both local and global cooccurrence information (Agirre et al. 2006). The latest effort to systematically evaluate sense induction systems took place in *SemEval-2007 - Evaluating Word SI and Discrimination Systems task* (Agirre & Soroa 2007a)[1]. Participants were asked to induce the senses of words in an unlabelled corpus and to cluster their instances. Systems were evaluated according to an unsupervised and a supervised scheme.

We reviewed all sense induction systems that were evaluated under this framework and chose the best performing one. Klapaftis & Manandhar (2008) propose a graph-based system that exploits the limited polysemy of collocations and *small world* properties of collocational graphs. It is shown to perform consistently well in both evaluation schemes, mainly because it represents context as collocations, which are much less ambiguous than single tokens. It usually induces a few more clusters than the gold-standard senses, but due to their small size their effect on *F-Score* is minor. The method leads to a skewed distribution of instances that the induced senses disambiguate, similar to the power law, that describes the gold-standard senses. However, using collocations as context is more vulnerable to data sparsity than using single words.

Figure 4.2 presents a running example, which describes an overview of the application of the chosen sense induction system for the multiword expression “red carpet”. The left side of part I shows the nouns of each of the four input snippets A-D of a really small corpus. In this work, a collocation is defined as a pair of nouns cooccurring within a snippet. Part II shows the collocations created by pairing the nouns of the example snippets in the left side of part I. The right side of part I shows the set of collocations that

[1]The Word Sense Induction task of SemEval-2010 is running as this thesis is being written

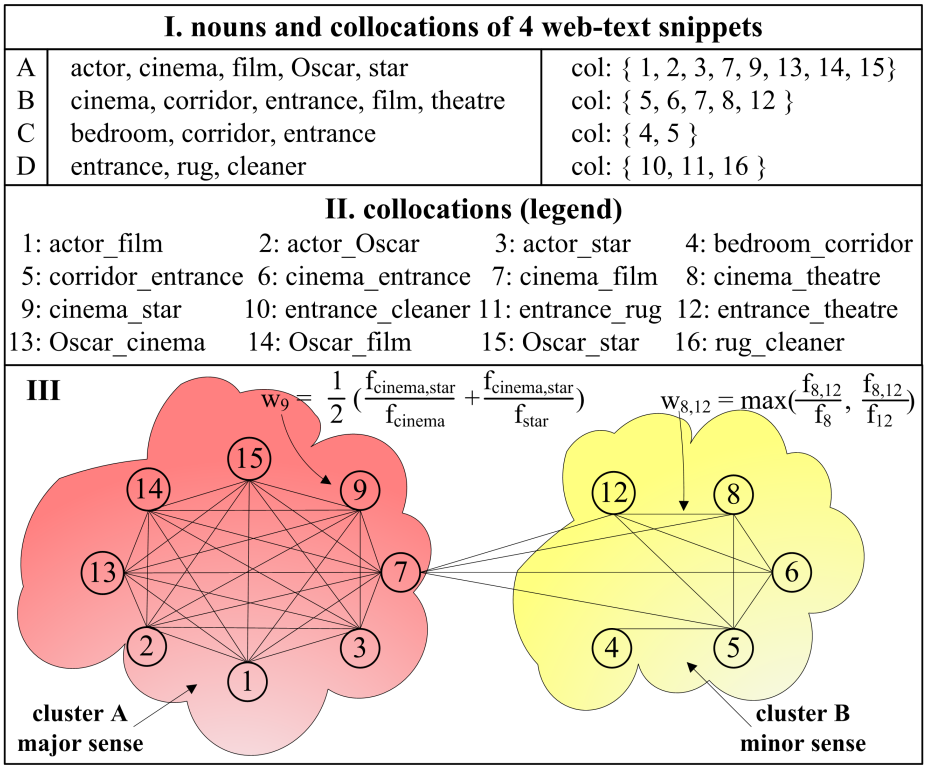

Figure 4.2: “red carpet”, sense induction example

corresponds to each snippet A-D. The sense induction method consists of 3 stages:

4.2.2. 1: Locating semantically important nouns

4.2.2. 2: Graph creation

4.2.2. 3: Graph clustering

4.2.2.1 Locating semantically important nouns

The target of this stage is to inspect the paragraphs of the base corpus, i.e. the corpus collected for a multiword expression or its semantic head, so as to determine which nouns are related and filter out the remaining. Kept words, topically and potentially semantically related to the target multiword expression or semantic head, will form the collocations in the next stage.

For filtering out common nouns which are noisy; i.e. contextually not related to the target, a large reference corpus is employed for frequency comparisons. We used *Web*

1T 5-gram Corpus (Brants & Franz 2006) as a reference corpus. Initially, the target multiword expression or the semantic head is removed from the base corpus. Then, log-likelihood (Dunning 1993) is employed to compare the distribution of each noun in the base corpus to its distribution in the reference corpus. The null hypothesis is that the two distributions are similar. Let n be a noun, b the base corpus and c the reference corpus; then the null hypothesis can be written as: $p(n|b) = p(n|r)$ The smaller the log-likelihood value, the most similar these two distributions are. If the log-likelihood value is less than a predefined threshold, $P1$, and the corresponding noun is removed.

The log-likelihood filtering process identifies nouns that are more indicative in the base corpus than in the reference corpus and vice versa. However, we are only interested in nouns which have a distinctive frequency in the base corpus. To filter out nouns with a distinctive frequency in the reference corpus relative frequency can be used. Target nouns cw whose relative frequency in the base corpus is less than their frequency in the relative corpus are filtered out. At the end of this stage, each snippet in the original base corpus is transformed into a list of nouns which are assumed to be contextually related to the target multiword expression or semantic head. In the running example of figure 4.2 the left side of part I shows the selected nouns for each snippet A-D.

4.2.2.2 Graph creation

Graph creation consists of choosing the vertices and edges of the graph and computing edge weights. A collocation is defined as a pair of nouns cooccurring within a snippet of the base corpus. Each noun within a snippet is combined with every other, generating $\binom{n}{2}$ collocations. For our running example, the collocation set corresponding to each snippet is shown on the right side of part I in figure 4.2. Part II explains the collocation numbering.

Each collocation is assigned a weight, measuring the relative frequency of the two cooccurring nouns. Let collocation c_{ij} consist of nouns i and j. Let f_x be the number of paragraphs in which noun x occurs, and f_{xy} the number of paragraphs in which nouns x and y cooccur. The weight, w_{ij}, corresponding to collocation c_{ij} is defined as the average of conditional probabilities of nouns i and j:

$$w_{ij} = \frac{1}{2}\left[p(i|j) + p(j|i)\right] = \frac{1}{2}\left[\frac{f_{ij}}{f_j} + \frac{f_{ij}}{f_i}\right] \tag{4.1}$$

For example, see w_9 in part III of figure 4.2.

In contrast to Klapaftis & Manandhar (2008), the computation of these conditional probabilities was not based on frequency counts of i and j in the base corpus. Using collocations as context suffers from data sparsity and thus frequency counts obtained from web-corpora are very weak to depend upon. Instead frequencies of nouns or collocations were estimated as the number of web-pages, in which they occur together with the target expression according to the *Yahoo!* search engine.

Only collocations whose frequencies and weights are greater than the prespecified thresholds $P2$ and $P3$, respectively, are represented as graph vertices. In the example of figure 4.2, "bedroom_entrance", was filtered out. This filtering appears to compensate for inaccuracies in the log-likelihood threshold, and for low-frequency distant collocations that are ambiguous.

Collocations which survive the previous thresholds are represented as graph vertices. Two vertices are connected with an edge, if they cooccur, in one or more snippets of the base corpus. The weight, w_{ab}, of an edge connecting vertices v_a and v_b that correspond to collocations a and b is computed as the maximum of their conditional probabilities, $p(a|b)$ and $p(b|a)$:

$$w_{ab} = \max\left[p(a|b), p(b|a)\right] = \max\left[\frac{f_{ab}}{f_b}, \frac{f_{ab}}{f_a}\right] \tag{4.2}$$

Frequencies of collocations and collocation pairs were again estimated by the number web-pages returned by *Yahoo!*. As a collocation weight example, see $w_{8,12}$ in part III of figure 4.2. Only, the largest connected component of the resulting graph was kept. Possible smaller disconnected components were discarded, such as the component consisting of collocations 10, 11 and 16 in figure 4.2. The output of this stage is a connected graph consisting of weighted vertices which represent collocations and weighted edges which represent collocation cooccurrences.

4.2.2.3 Graph clustering

In this stage, the collocational graph is clustered to produce the senses of the target word. Each cluster correspond to a sense. In part III of figure 4.2 the coloured clouds correspond to different clusters or in other words induced senses.

The clustering method employed is *Chinese Whispers* (Biemann 2006). Chinese

Whispers is a randomised graph-clustering method, time-linear to the number of edges. It offers the advantage that it does not require any input parameters, since the number of clusters it produces is automatically inferred. It is not guaranteed to converge, but experimentation showed that very few changes occur after a small number of iterations, such as 20. Thus, 200 was adopted as the maximum number of iterations for the experiments of this chapter, a number that seems more than enough to cover any extreme cases.

Chinese Whispers works as follows: Initially, it assigns all vertices to different clusters. Each vertex is processed for a number of iterations and inherits the strongest cluster in its local neighbourhood in an update step. Local neighbourhood is defined as the set of vertices which share an edge with the current one. In each iteration for the current vertex i, each cluster, cl, receives a score equal to the sum of the weights of edges (i, j), where j has been assigned to cluster cl. The maximum score determines the strongest cluster. In case of multiple strongest clusters, one is chosen randomly. Clusters are updated immediately, meaning that a vertex can inherit from its local neighbourhood clusters that were introduced in the same iteration.

Evaluation of *Chinese Whispers* has shown that it suits sense induction applications well, because the class distributions are often highly skewed. Our experiments showed that *Chinese Whispers* produces less clusters using a constant mutation rate (5%). Thus, we adopted this mutation rate for all experiments.

Our experiments agree with Klapaftis & Manandhar (2008) where it is reported that Chinese Whispers produces larger number of clusters than expected. To reduce it we exploit the *one sense per collocation* property (Yarowsky 1995) to develop a cluster merging technique. Given two clusters l_i and l_j, we compute the sets of snippets S_i and S_j that contain at least one collocation of l_i and l_j, respectively. Clusters l_i and l_j should be merged if $S_i \subseteq S_j$ or if $S_j \subseteq S_i$. In other words, two clusters are merged if the set of snippets that one of them disambiguates is a subset of the set of snippets disambiguated by the other.

4.2.3 Comparison of major induced senses

We used two techniques to measure the distributional similarity of the major senses of the multiword expression and its semantic head. "Major sense" denotes the cluster of collocations which disambiguates the most snippets in the base corpus, i.e. the cluster whose collocations occur in the most snippets. In all our experiments, the major sense is

the cluster that consists of the most collocations. For example in part III of figure 4.2 the left side cluster in red is the major sense while the right side cluster in yellow is a minor one.

Both techniques are based on Jaccard coefficient which was discussed in section 2.3.3. A comparison of different similarity measures shows that Jaccard coefficient performs better than other symmetric similarity measures such as *Cosine*, *Euclidean distance*, *Jensen-Shannon divergence*, etc. (Lee 1999). Jaccard coefficient is a combinatorial measure that computes the similarity of two sets as the cardinality of their intersection over the cardinality of their union. The first distributional similarity measure that we use is Jaccard coefficient on sets of collocations: Let A and B be two sets of collocations. Then, Jaccard coefficient on sets of collocations, J_c, is defines as:

$$J_c = J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.3}$$

Usually, the major use of the head of the multiword expression is much larger in number of collocations than the major use of the multiword expression itself. Thus, Jaccard coefficient on sets of collocations is usually very close to 0, restricting its discriminating ability. To overcome this problem we used a second distributional similarity measure, Jaccard coefficient on snippets. It computes similarity based on the number of snippets that are tagged by the induced senses. Let K_i be the set of snippets in which at least one collocation of the sense i occurs. Given that j and k are the major uses of the multiword expression and its semantic head, respectively, Jaccard coefficient on snippets, J_{sn}, is computed as:

$$J_{sn} = J(K_j, K_k) = \frac{|K_j \cap K_k|}{|K_j \cup K_k|} \tag{4.4}$$

4.2.4 Determining compositionality

At this stage, the major sense of the each multiword expression has been compared to the major sense of its semantic head. For this comparison, either Jaccard similarity on sets of collocations or Jaccard similarity on disambiguated snippets has been employed. The result of this comparison should be used to judge the corresponding multiword expression as compositional or not.

For this purpose we introduce a threshold. A multiword expression is considered

as compositional, if the corresponding distributional similarity measure value is above a parameter threshold, *sim*. Otherwise, it is considered as non-compositional. *sim* mainly depends on the quantity of data available for the multiword expression and its head.

4.3 Test set of multiword expressions

To the best of our knowledge there are no noun compound datasets accompanied with compositionality judgements available. Thus, we developed an algorithm to aid human annotation in adding compositionality judgements to multiword expressions extracted from *WordNet 3.0* (Miller 1995). *WordNet 3.0* contains $52,217$ multiword expressions. For each occurrence of each multiword expression in some synset we collect:

1: all its synonyms, i.e. the items of the same synsets.
2: all its hypernyms
3: its sister-synsets, within distance 3
4: synsets that are in holonymy or meronymy relation to it, within distance 3.

All these words or sequences are put in a different set for each sense of the current multiword expression, i.e. for each of its occurrences in *WordNet*. We choose to call these sets sense neighbourhoods of the target multiword expression.

The definitions of the latter two items of the above list use a distance threshold. Locating sister-synsets at distance *D* implies ascending *D* steps within the *WordNet* hierarchy and then descending *D* steps, so that the destination is at the same level with the origin. In the case of locating sister-synsets these steps follow hypernym relations when ascending and hyponym relations when descending. Locating synsets in holonymy or meronymy relations to the target in distance *D*, implies that the steps follow holonymy relations when ascending and meronymy relations when descending.

Evidence about the compositionality of each sense of a given multiword expression in *WordNet* can be collected according to the following criterion: If the semantic head of the multiword expression occurs in the sense neighbourhood of the multiword expression then this sense of the multiword expression is likely to be compositional. Otherwise it is likely that the corresponding sense of the multiword expression is non-compositional.

Using the above algorithm, $6,287$ multiword expressions were judged as potentially non-compositional. The vast majority of them, $5,489$, appear in one *WordNet* synset, only. 294 appeared in more than one synsets and were in all occurrences judged as

Non-compositional multiword expressions							
	B	*M*	*A*		*B*	*M*	*A*
agony aunt	✗	✓	✓	black maria	✓	✓	✓
dead end	✓	✓	✓	dutch oven	✗	✗	✓
fish finger	✓	✓	✓	fool's paradise	✗	✓	✓
goat's rue	✓	✓	✓	green light	✓	✓	✓
high jump	✓	✓	✓	joint chiefs	✗	✓	✗
lip service	✓	✓	✓	living rock	✓	✓	✓
monkey puzzle	✓	✓	✓	motor pool	✓	✓	✓
prince Albert	✓	✓	✓	stocking stuffer	✓	✓	✓
sweet bay	✓	✓	✓	teddy boy	✓	✓	✗
think tank	✓	✓	✓				

Compositional multiword expressions							
	B	*M*	*A*		*B*	*M*	*A*
box white oak	✗	✗	✗	cartridge brass	✓	✗	✗
common iguana	✓	✗	✓	closed chain	✗	✗	✗
eastern pipistrel	✗	✗	✗	field mushroom	✗	✗	✗
hard candy	✓	✗	✗	king snake	✓	✗	✓
labor camp	✓	✓	✓	lemon tree	✗	✗	✗
life form	✓	✓	✓	petit juror	✓	✓	✓
parking brake	✓	✗	✓	taxonomic category	✗	✗	✓
relational adjective	✓	✗	✗	tea table	✗	✗	✗
telephone service	✓	✓	✓	parenthesis-free	✗	✗	✗
upland cotton	✓	✓	✓	notation			

Table 4.1: Test multiword expressions with compositionality annotation and information about whether their compositionality was successfully detected by the *1c1word* baseline (*B*), manual parameter selection (*M*), and *average cluster coefficient* (*A*).

potentially non-compositional. 504 appeared in more than one synsets and at least one of these occurrences was judged as potentially non-compositional.

We randomly chose 19 potentially non-compositional multiword expressions and checked them manually. Those that were compositional were replaced by other randomly chosen ones. The process was repeated until we ended up with 19 non-compositional examples. Similarly, 19 potentially compositional multiword expressions were collected.

Parameter	Description	Range
$P1$	log-likelihood filter	5.0, 10.0, 15.0
$P2$	Collocation frequency	10^2, 10^3, 10^4, 10^5
$P3$	Collocation weight	0.2, 0.3, 0.4

Table 4.2: Chosen parameter values.

The upper part of table 4.1 shows the chosen non-compositional multiword expressions, while its lower part the compositional ones.

4.4 Evaluation setting

The sense induction component of the proposed algorithm depends upon 3 parameters:

1: in corpus preprocessing, P_1 is the log-likelihood threshold below which noun are removed from corpora
2: in graph creation, P_2 thresholds collocation frequencies
3: in graph creation again, P_3 thresholds collocation weights

The system can potentially work for any parameter values, given that the corresponding threshold constraints are fulfilled by some nouns, collocations and collocation weights. We choose to evaluate the proposed system in the parameter subspace shown in table 4.2. The log-likelihood values for confidence levels of 95%, 99%, 99.9% and 99.99% are 3.84, 6.63, 10.83 and 15.13, respectively. The values of parameter $P1$ were chosen to cover this range. Parameter $P2$ thresholds *Yahoo!* web-page counts and parameter $P3$ thresholds conditional probabilities. Values for parameters $P2$ and $P3$ have been chosen empirically.

To assess the performance of the proposed algorithm we compute *accuracy*, the percentage of multiword expressions whose compositionality was correctly determined against the gold-standard. We compared the system's performance against a baseline, *1c1word*, that assigns all vertices to a single cluster and no graph clustering is performed. This baseline is also used in *SemEval-2007* (Agirre & Soroa 2007a). *1c1word* is a sensible baseline for this task, since it corresponds to a vector space model for words. It considers the whole contextual vector, while the proposed system considers the largest partition of it. Baseline *1c1word* helps in showing whether sense induction can assist

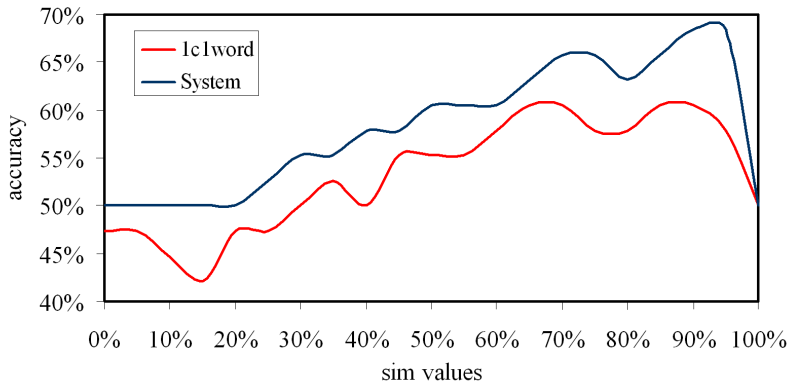

Figure 4.3: Proposed system and *1c1word* accuracy.

determining compositionality, since it represents applying exactly the same procedures except sense induction.

4.5 Evaluation results

The proposed method was evaluated for each $\langle P_1, P_2, P_3 \rangle$ combination and similarity measures J_c and J_{sn}, separately. Given a similarity value to threshold the decision of compositionality, we chose the best performing parameter combination manually. In all experiments, J_{sn} outperforms J_c. Figure 4.3 shows the comparison of the proposed system against the *1c1word* baseline for all similarity values (*sim*), at a scale of 0.5% increments. The system line shows the accuracy achieved by the best performing combination of parameter values $\langle P_1, P_2, P_3 \rangle$ using J_{sn} to compare the major senses of each multiword expression and its semantic head. The best combination of parameter values is $\langle 10.0, 10^2, 0.3 \rangle$ for the multiword expressions and $\langle 10.0, 10^2, 0.4 \rangle$ for semantic heads.

The initial hypothesis holds; sense induction improves multiword expression compositionality detection, since the proposed system outperforms the baseline for all distributional similarity threshold values. The best results for manual parameter selection were obtained for $sim = 95\%$ giving an accuracy of 68.42% for detecting non-compositional multiword expressions. Columns *B* and *M* of table 4.1 show for each multiword expression of the test set whether its compositionality was predicted correctly by the baseline and the system using manual parameter selection, respectively.

In table 4.1, all systems appear to predict non-compositional expressions more suc-

cessfully that compositional ones. As far as the baseline system and the manual parameter selection system are concerned, the table shows results for $sim = 95\%$, since this is the best performing value. Recall that sim is the threshold contextual similarity value above which a multiword expression is judged as compositional. For threshold values close to 100%, as the one showed in table 4.1, it is very unlikely to judge a multiword expression as compositional. This is the main reason for the tendency observed in table 4.1.

4.6 Unsupervised parameter tuning

In this section we investigate unsupervised ways to address the issue of choosing the parameter values for graph-based sense induction systems. For this purpose, we employ a variety of graph connectivity measures, which measure the relative importance of each vertex and assess the overall connectivity of the corresponding graph. These measures are *average degree*, *cluster coefficient*, *graph entropy* and *edge density* (Navigli & Lapata 2007; Zesch & Gurevych 2007).

The approach employed by Klapaftis & Manandhar (2008) as well as the majority of state-of-the-art word sense induction systems estimate their parameters either empirically or by employing supervised techniques. The SemEval-2007 word sense induction task (SWSI) participating systems *UOY* and *UBC-AS* used labelled data for parameter estimation (Agirre & Soroa 2007a), while the authors of *I2R*, *UPV_SI* and *UMND2* have empirically chosen values for their parameters. This issue imposes limits on the unsupervised nature of these algorithms, as well as on their performance on different datasets.

More specifically, when applying an unsupervised word sense induction system on different datasets, one cannot be sure that the same set of parameters is appropriate for all datasets (Karakos et al. 2007). In most cases, a new parameter tuning might be necessary. Unsupervised estimation of free parameters may enhance the unsupervised nature of systems, making them applicable to any dataset, even if there is no tagged data available.

Graph connectivity measures quantify the degree of connectivity of the produced clusters (subgraphs), which represent the senses of the target word for a given parameter setting and the corresponding clustering solution. Each clustering solution (parameter setting) is assigned a score according to each graph connectivity measure and the highest scoring setting is then selected. Higher values of graph connectivity measures indicate

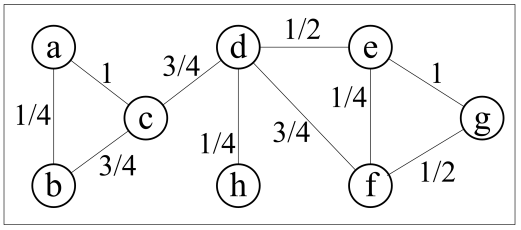

Figure 4.4: An example undirected weighted graph

subgraphs (clusters) of higher connectivity. Given a parameter setting, the induced clustering solution and a graph connectivity measure, each induced cluster is assigned the resulting score of applying the graph connectivity measure on the corresponding subgraph of the initial unclustered graph. Each clustering solution (parameter setting) is assigned the average of the scores of its clusters, and the highest scoring one is selected.

For each graph connectivity measure, we developed two versions. An unweighted one, that considers the edges of the subgraph corresponding to each cluster but not the edge weights, and a weighted one, which considers subgraph edge weights, as well. In the following discussion, terms graph and subgraph are interchangeable.

Let $G = (V, E)$ be an undirected graph (induced sense), where V is a set of vertices and $E = \{(u, v) : u, v \in V\}$ a set of edges connecting vertex pairs. Each edge is weighted by a positive weight, $W : w_{uv} \rightarrow [0, \infty)$. The maximum edge weight in the graph is:

$$mew = \max_{(u,v)\in E} w_{uv} \tag{4.5}$$

Figure 4.4 shows a small example to explain the computation of graph connectivity measures. The graph consists of 8 vertices, $|V| = 8$, and 10 edges, $|E| = 10$. Edge weights appear on edges, e.g. $w_{ab} = \frac{1}{4}$. The maximum edge weight in the graph of figure 4.4 is: $mew = 1$.

4.6.1 Average Degree

The *degree (deg)* of a vertex, u, is the number of edges connected to it:

$$deg(u) = |\{(u, v) \in E : v \in V\}| \tag{4.6}$$

	a	b	c	d	e	f	g	h
$deg(u)$	2	2	3	4	3	3	2	1
$wdeg(u)$	$\frac{5}{4}$	1	$\frac{5}{2}$	$\frac{9}{4}$	$\frac{7}{4}$	$\frac{3}{2}$	$\frac{3}{2}$	$\frac{1}{4}$
T_u	1	1	1	1	1	2	1	0
$cc(u)$	1	1	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{2}{3}$	1	0
WT_u	$\frac{3}{4}$	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{1}{4}$	0
$wcc(u)$	$\frac{3}{4}$	1	$\frac{1}{12}$	$\frac{1}{24}$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{4}$	0
$p(u)$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{3}{20}$	$\frac{1}{5}$	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{1}{10}$	$\frac{1}{20}$
$en(u) \times 100$	33	33	41	46	41	41	33	22
$wp(u)$	$\frac{1}{16}$	$\frac{1}{20}$	$\frac{1}{8}$	$\frac{9}{80}$	$\frac{7}{80}$	$\frac{3}{40}$	$\frac{3}{40}$	$\frac{1}{80}$
$we(u) \times 100$	25	22	38	35	31	28	28	8

Table 4.3: Computations of graph connectivity measures and relevant quantities on the example graph of figure 4.4

The *average degree (AvgDeg)* of a graph can be computed as:

$$AvgDeg(G(V,E)) = \frac{1}{|V|}\sum_{u \in V} deg(u) \tag{4.7}$$

The first row of table 4.3 shows the vertex degrees of all vertices in the example graph of figure 4.4. Average degree is the mean of all vertex degrees: $AvgDeg(G) = \frac{20}{8} = 2.5$.

4.6.2 Average Weighted Degree

Edge weights can be integrated into the computation of vertex degrees. The *weighted degree (w_deg)* of a vertex is defined as:

$$w\_deg(u) = \frac{1}{|V|}\sum_{(u,v) \in E} \frac{w_{uv}}{mew} \tag{4.8}$$

Average weighted degree (AvgWDeg), similarly to *AvgDeg*, is averaged over all vertices of the graph:

$$AvgWDeg(G(V,E)) = \frac{1}{|V|}\sum_{u\in V} W\_deg(u) \tag{4.9}$$

The second row of table 4.3 shows the weighted degrees of all vertices. Average weighted degree in this graph is: $AvgWDeg(G) = \frac{48}{36} \simeq 1.33$.

4.6.3 Average Cluster Coefficient

The *cluster coefficient* of a vertex quantifies how strongly the neighbours of the vertex are connected with each other. It is defined as the number of edges between the neighbours of the vertex over the maximum number of edges that could exist between its neighbours. Let u be a vertex with $k_u = deg(u)$ neighbours. The number of edges between these neighbours can be computed as follows:

$$T_u = \sum_{(u,v)\in E} \sum_{\substack{(v,x)\in E \\ x\neq u}} 1 \tag{4.10}$$

If the graph defined by all k_u neighbours of vertex u was fully connected, there would be $2^{-1}k_u(k_u-1)$ among them. Thus, the *cluster coefficient (cc)* of a vertex u can be computed as:

$$cc(u) = \frac{2\times T_u}{k_u \times (k_u - 1)} \tag{4.11}$$

Average cluster coefficient (AvgCC) is averaged over all vertices of the graph.

$$AvgCC(G(V,E)) = \frac{1}{|V|}\sum_{u\in V} cc(u) \tag{4.12}$$

The computations of T_u and $cc(u)$ on the example graph are shown in the third and fourth rows of table 4.3. Consequently, $AvgCC(G) = \frac{9}{16} = 0.5625$.

4.6.4 Average Weighted Cluster Coefficient

Weighted cluster coefficient takes edge weights into account. Thus, the sum, WT_u, of edge weights between the neighbours of u is computed as:

$$WT_u = \frac{1}{mew} \sum_{(u,v)\in E} \sum_{\substack{(v,x)\in E \\ x\neq u}} w_{vx} \tag{4.13}$$

Weighted cluster coefficient (wcc) can be computed as:

$$wcc(u) = \frac{2 \times WT_u}{k_u \times (k_u - 1)} \tag{4.14}$$

Average weighted cluster coefficient (AvgWCC) is averaged over all vertices of the graph.

$$AvgWCC(G(V,E)) = \frac{1}{|V|} \sum_{u\in V} wcc(u) \tag{4.15}$$

The computations of WT_u and $wcc(u)$ on the example graph of figure 4.4 are shown in the fifth and sixth rows of table 4.3. In the example graph, the average weighted cluster coefficient is: $AvgWCC(G) = \frac{67}{8\times 24} \simeq 0.349$.

4.6.5 Graph Entropy

Entropy measures the amount of information, or alternatively the uncertainty, in a random variable. For a graph, high *entropy* indicates that many vertices are equally important and low *entropy* that only few vertices are relevant (Navigli & Lapata 2007). The probability, $p(u)$, of a vertex, u, can be determined by the degree distribution:

$$p(u) = \left\{ \frac{deg(u)}{2|E|} \right\}_{u\in V} \tag{4.16}$$

Then, following the original definition of entropy, the entropy *(en)* of a vertex, u, can be defined as:

$$en(u) = -p(u) \log_2 p(u) \tag{4.17}$$

Entropy is undefined for isolated vertices, due to the logarithm definition[2]. *Graph entropy (GE)* is computed by summing all vertex entropies and normalising:

$$GE(G(V,E)) = \frac{1}{\log_2 |V|} \sum_{u \in V} en(u) \tag{4.18}$$

Returning to the example graph of figure 4.4, the seventh and eighth row of table 4.3 show the computations of $p(u)$ and $en(u)$, respectively. Consequently, the graph entropy is: $GE \simeq 0.97$.

4.6.6 Weighted Graph Entropy

The weighted probability, $wp(u)$, of a vertex, u, can be computed as:

$$wp(u) = \left\{ \frac{w\_deg(u)}{2 \times mew \times |E|} \right\}_{u \in V} \tag{4.19}$$

Similarly to the previous graph connectivity measures, *weighted entropy (wen)* of a vertex u is a weighted generalisation of *entropy*:

$$we(u) = -wp(u) \log_2 wp(u) \tag{4.20}$$

Weighted graph entropy (WGE) is computed by summing the weighted entropies of all vertices and normalising:

$$WGE(G(V,E)) = \frac{1}{\log_2 |V|} \sum_{u \in V} we(u) \tag{4.21}$$

The last two rows of table 4.3 show the computations of $wp(u)$ and $we(u)$ on the example graph. Consequently, the weighted graph entropy is: $WGE \simeq 0.73$.

4.6.7 Edge Density

Edge density (ed) quantifies how many edges the graph has, as a ratio over the number of edges of a fully connected graph of the same size, $A(V)$:

$$A(V) = 2 \binom{|V|}{2} \tag{4.22}$$

[2]However, this never occurs in this work, since the largest connected component is kept during the graph creation stage discussed in section 4.2.2.2.

Edge density (ed) is a global graph connectivity measure; it refers to the whole graph and not a specific vertex. It is defined as follows:

$$ed(G(V,E)) = \frac{|E|}{A(V)} \tag{4.23}$$

In the example graph of figure 4.4: $A(V) = 2\binom{8}{2} = 28$. Consequently, edge density is: $ed(G) = \frac{10}{28} \simeq 0.357$.

4.6.8 Weighted Edge Density

Weighted edge density (wed) is defines as a portion of *edge density* as large as the sum of all edge weights:

$$wed(G(V,E)) = \frac{1}{A(V)} \sum_{(u,v)\in E} w_{u,v} \tag{4.24}$$

In the example graph of figure 4.4: $\sum w_{u,v} = 6$. Consequently, weighted edge density is: $wed(G) = \frac{6}{28} \simeq 0.214$. ■

The use of the aforementioned graph connectivity measures allows the estimation of a different parameter setting for each multiword expression and semantic head. These parameters affect how the collocational graph is constructed, and in effect the quality of the induced clusters.

4.7 Evaluation results of unsupervised parameter tuning

The graph connectivity measures of the previous subsection are used to choose a set of parameters $\langle P1, P2, P3\rangle$ for each multiword expression and each semantic head, separately. In all experiments, J_{sn} performs better than J_c. Figures 4.5 and 4.6 present a comparison between the unweighted and weighted versions of all graph connectivity measures, respectively, for all similarity values (*sim*), at a scale of 0.5% increments, using J_{sn} as distributional similarity measure.

Unweighted versions of graph connectivity measures perform in general better than weighted ones. *Average cluster coefficient* performs better or equally well to the other *graph connectivity measures* for all sim values up to 80%. The accuracy of *average cluster coefficient* is equal (68.42%) to that of manual parameter selection, which

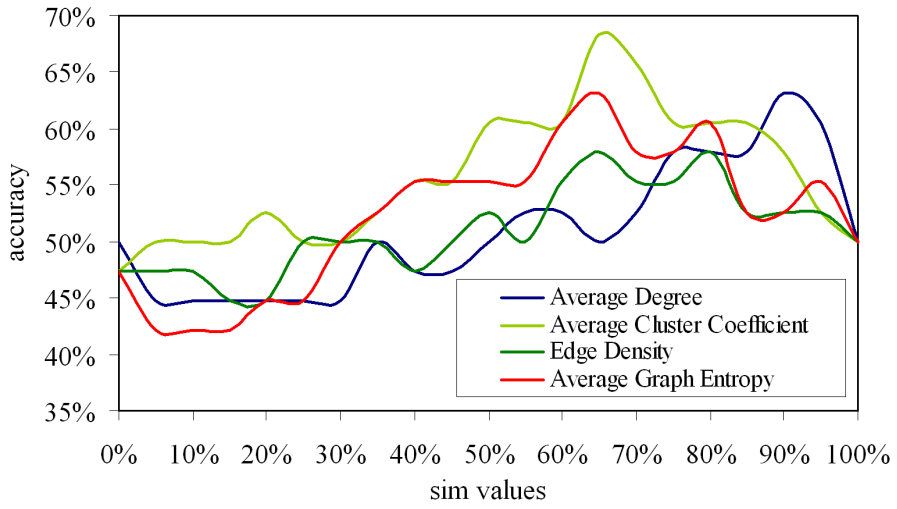

Figure 4.5: Unweighted graph connectivity measures.

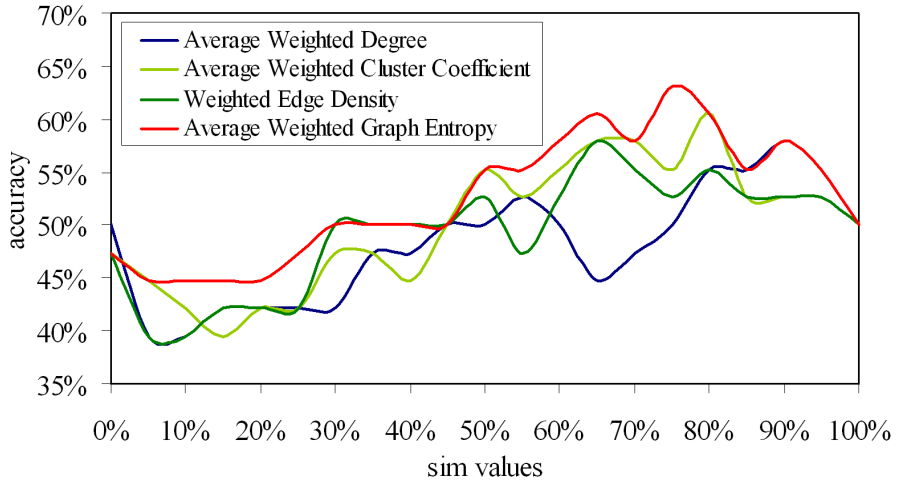

Figure 4.6: Weighted graph connectivity measures.

was plotted in figure 4.3 and discussed in section 4.5. The second best performing unweighted *graph connectivity measure* is *average graph entropy*. For weighted *graph connectivity measures*, *average graph entropy* performs best, followed by *average weighted clustering coefficient*.

Figure 4.7 presents a comparison between the following systems:

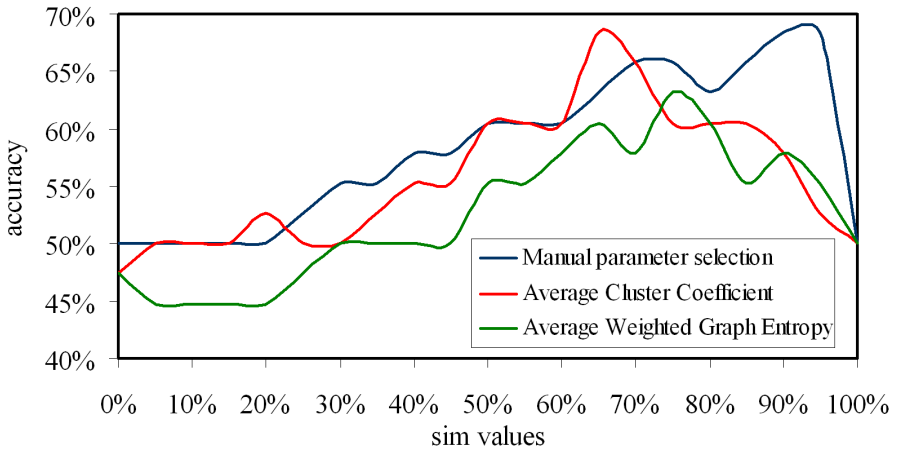

Figure 4.7: Comparison of manual parameter tuning and the two best performing graph connectivity measures.

1: the basic system using manual parameter selection, as described in subsection 4.5.

2: the basic system using the best performing unweighted graph connectivity measure, *average cluster coefficient*, to automatically estimate parameter values for each multiword expression and semantic head, separately.

3: the basic system using the best performing weighted graph connectivity measure, *average weighted graph entropy*, to automatically estimate parameter values.

The plot shows that *average cluster coefficient* performs closely to manual parameter selection and better than *average weighted graph entropy*.

For sim values greater than 80% manual parameter selection outperforms *average cluster coefficient*. In this region, manual parameter selection leads to clustering solutions consisting of few clusters of large size. High values of sim mean that the system judges a multiword expression as compositional only if the set of snippets tagged by its major sense is too similar to the set of snippets tagged by the major sense of its semantic head.

In contrast, *average cluster coefficient* favours solutions consisting of many small clusters. *Average cluster coefficient* assigns high scores to graphs whose number of edges is comparable to the number of edges of the corresponding fully connected graph. Hence, *average cluster coefficient* assigns high score to a clustering solution with many small clusters and correspondingly lower score to a clustering solution containing a few

large clusters. This is more possible to happen for a solution consisting of many small clusters than for a solution consisting of few large clusters. For sim values greater than 80% *average cluster coefficient* selects different parameter combinations to the best performing ones. The higher accuracy achieved by *average cluster coefficient* for $sim = 65\%$ is equal to the maximum manual parameter selection accuracy for $sim = 95\%$ (68.42%).

4.8 Evaluation on a larger dataset

The evaluation results that have been presented sofar look promising. However, there are concerns about the statistical significance of the improvement in terms of accuracy that the system achieves over the baseline. We employed two non parametric statistical tests: Fisher's exact test and McNemar's test. The latter is not suitable for small contingency table values, while the former one is. The values in the contingency tables representing the current experiments are marginal; thus, we applied both statistical significance tests.

According to Fisher's exact test, the proposed system with manually estimated parameters performs significantly better than the *1c1word* baseline for some similarity values (*sim*), only: 15%, 40%, [50%-55%], and [70%-95%] (see figure 4.3). In contrast, according to McNemar's test the same improvement is statistical insignificant for all similarity values (*sim*). Our intuition about this result is that most possibly the size of the dataset is very small.

To inspect whether the proposed system actually performs better than the baseline, we created a bigger dataset, shown in table 4.4. It consists of 100 multiword expressions, half of which are compositional and half non-compositional. This dataset is a super-set of the previous one.

On this new dataset, we evaluated the proposed system using exactly the same evaluation settings as before. Figure 4.8 presents a comparison between the accuracies achieved by the *1c1word* baseline, the system with manually estimated parameters, and the system with parameters automatically estimated by the best performing weighted and unweighted graph connectivity measures. We observe that for the meaningful range of similarity values (*sim*), [20%, 95%], the system with manual selected parameters performs better than the *1c1word* baseline. According to Fisher's exact test, this increase in accuracy is statistically significant for the whole range. In contrast, according to McNemar's test, the increase in significant for: [20% ,45%], 65%, and 90%.

Compositional multiword expressions			
action officer	basic color	car battery	box white oak
cartridge brass	checker board	closed chain	common iguana
corn whiskey	corner kick	cream sauce	cubic meter
eastern pipistrel	field mushroom	flight simulator	graphic designer
hard candy	honey cake	ill health	jazz band
jet plane	king snake	labor camp	laser beam
lemon tree	life form	love letter	luggage van
male parent	medical report	memory device	mythical monster
parking brake	petit juror	red fox	relational adjective
sausage pizza	savoy cabbage	surface fire	taxonomic category
tea table	telephone service	thick skin	touch screen
toxic waste	upland cotton	water snake	water tank
wood aster	parenthesis-free notation		

Non-compositional multiword expressions			
agony aunt	air conditioner	black maria	dead end
dutch oven	fire brigade	fish finger	fool's paradise
goat's rue	golden trumpet	green light	high jump
joint chiefs	lip service	living rock	magnetic head
monkey puzzle	motor pool	oyster bed	palm reading
paper chase	paper gold	paper tiger	personal equation
personal magnetism	petit four	picture palace	pill pusher
pink lady	pink shower	powder monkey	prince Albert
public eye	quick time	rat race	red devil
red dwarf	red tape	road agent	round window
sea lion	small beer	small voice	spin doctor
stocking stuffer	sweet bay	teddy boy	think tank
vegetable sponge	winter sweet		

Table 4.4: Test multiword expressions with compositionality annotation

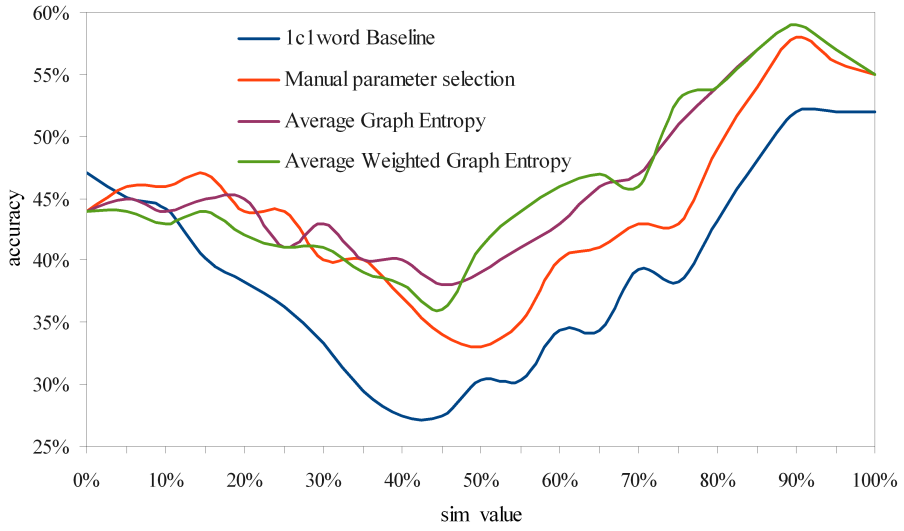

Figure 4.8: Comparison of baseline system, manual parameter tuning and the two best performing graph connectivity measures

As far as automatic parameter tuning is concerned, the best performing unweighted and weighted graph connectivity measures are average graph entropy and weighted average graph entropy. They perform very similarly to each other and it is not clear which one is best. However, both systems with automatically tuned parameters perform significantly better than the baseline for a meaningful range of similarity values (*sim*), [20%, 95%], according to both statistical significance tests.

Interestingly, most values of the four systems in figure 4.8 for similarity values in [0%, 75%] are less than 50%. However, the dataset consists of an equal number of compositional and non-compositional multiword expressions. There are several reasons why accuracy for all systems happens to be lower than 50%. A major one is that small similarity values are expected to judge most multiword expressions as compositional. At the same time, some vectors are very noisy, since the data is downloaded from the web. Due to the great differences in frequency of the multiword expressions, different settings are mostly suitable for each. This is only taken into account by the parameter estimation scheme that employs graph connectivity measures.

Figures 4.9 and 4.10 show the accuracy achieved by the systems using unweighted and weighted graph connectivity measures for automatic parameter estimation, respectively. We observe that the worst performing ones, average degree and weighted average

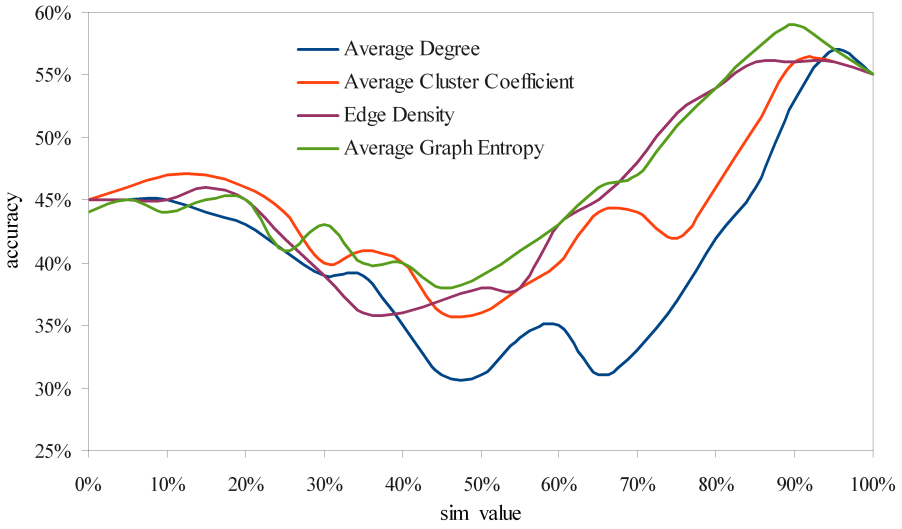

Figure 4.9: Unweighted graph connectivity measures.

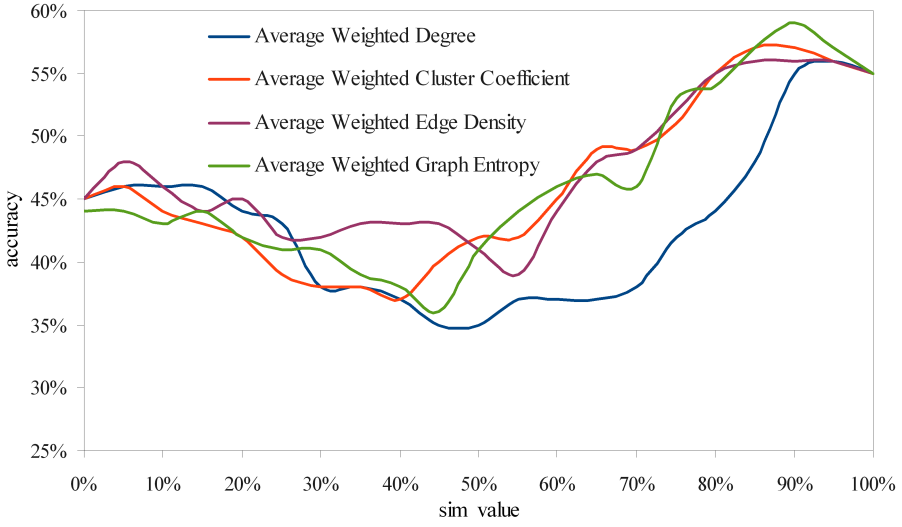

Figure 4.10: Weighted graph connectivity measures.

degree are still not much worse than the others. The remaining ones, unweighted and weighted versions of average cluster coefficient, edge density, and average graph entropy perform similarly. Average graph entropy and weighted average graph entropy achieve the highest accuracy value.

Experimentation on this expanded dataset proved that the proposed system is signi-

ficantly better than the *1c1word* baseline. Also, it was shown that in some cases a system whose parameters are automatically tuned can perform better than one whose parameters were chosen manually. The reason is that during manual parameter estimation the best "universal" parameter combination was chosen. This means that for all multiword expressions and their corresponding semantic heads the parameters are the same. In contrast, the automatic parameter estimation scheme, that was presented in section 4.6, selects a different parameter setting for each word or multiword expression whose senses are induced.

4.9 Further evaluation of unsupervised parameter tuning

The automatic parameter estimation scheme presented in section 4.6 was used in further experiments, so as to exploit the extent to which they are useful for graph-based sense induction systems. Both the weighted and unweighted graph connectivity measures were used to choose the parameters of the approach of Klapaftis & Manandhar (2008) evaluated on the nouns of the word sense induction task (SWSI) of SemEval-2007. The results showed that:

1: all graph connectivity measures estimate a set of parameters which significantly outperform the worst performing parameter setting in both SWSI evaluation schemes, although they are below the best performing parameter setting.
2: all graph connectivity measures estimate a set of parameters which outperform the Most Frequent Sense baseline by a statistically significant amount in the supervised evaluation scheme.
3: The best performing graph connectivity measures, *average degree* and *weighted average degree*, estimate a set of parameters that performs closely to a set of parameters estimated in supervised manner.

All of these findings, suggest that graph connectivity measures are able to identify useful differences regarding the degree of connectivity of induced clusters for different parameter combinations, in effect being useful for unsupervised parameter estimation. In this subsection we briefly summarised this evaluation, since it is not relevant to multiword expressions. Below, we describe in short the corpus and evaluation schemes used for the SWSI task. More details can be found in Korkontzelos et al. (2009).

The collocational word sense induction approach Klapaftis & Manandhar (2008) was evaluated under the framework and corpus of SemEval-2007 word sense induction task (SWSI) (Agirre & Soroa 2007a). The corpus consists of some text from the Wall Street Journal corpus, and is hand-tagged with OntoNotes senses (Hovy et al. 2006). The evaluation focuses on all 35 nouns of the Semeval-2007 task. SWSI task employs two evaluation schemes.

In unsupervised evaluation, the results are treated as clusters of contexts and gold-standard senses as classes. In a perfect clustering solution, each induced cluster contains the same contexts as one of the classes (homogeneity), and each class contains the same contexts as one of the clusters (completeness). F-Score is used to assess the overall quality of clustering. Entropy and purity are also used, complementarily. F-Score is a better measure than entropy or purity, since F-Score measures both homogeneity and completeness, while entropy and purity measure only the former. In the second scheme, supervised evaluation, the corpus is split in a training and a testing part. The training part is used to map the induced clusters to gold-standard senses. The testing part is then used to measure word sense disambiguation performance.

4.10 Summary

In this chapter, we started with the hypothesis that sense induction can assist in identifying compositional multiword expressions and exploited the extent to which this hypothesis holds. Following the context distributional approach to decide compositionality, we developed a novel unsupervised system that employs a graph-based sense induction component. This component is useful to partition the context distribution of a target multiword expressions and its semantic head. Given the partitioned context distributions, the proposed method locates the major ones for the multiword expression and its semantic head, respectively, and compares them using distributional similarity measures. It is expected that the major sense of a word or multiword expression describes what is meant when it occurs without context. Finally, the similarity value of the major senses of the multiword expression and its head is compared with a threshold to judge if the expression is compositional or not.

The proposed method was evaluated on adjective-noun constructions, compound nominals and proper names, in English. The test set is extracted from *WordNet*. We propose a semi-supervised approach for adding compositionality annotation to its mul-

tiword expressions, that minimises human effort. It is shown that the initial hypothesis holds when the parameters of the system are estimated manually, i.e. that sense induction can assist in identifying compositional multiword expressions.

In succession, we propose an unsupervised scheme for estimating the parameters of graph-based sense induction systems. It is based on connectivity properties of graphs and attempts to distinguish high quality graphs exploring a number of graph connectivity measures. The scheme scores the partition output (clustering) of context distributions corresponding to each parameter setting; and thus indirectly scores each parameter setting. The highest scoring parameter setting is selected, without any need of manually annotated data.

Then, this unsupervised parameter selection scheme is used to estimate the parameters of the proposed system for resolving compositionality. The results show that the scheme for unsupervised parameter tuning achieves comparable accuracy to the best manually selected combination of parameters.

CHAPTER 5

Multiword Expressions and Parsing

Executive Summary

This chapter presents our research towards evaluating the contribution of multiword expression knowledge into shallow parsing. We adopt a simple but effective way of encoding multiword expression annotation into the input text and compare the performance of the shallow parser with or without this extra information. We analyse the contribution of shallow parsing for each type of multiword expressions, separately, as far as parts of speech and compositionality are concerned. The evaluation set of multiword expressions is derived from *WordNet* and the textual data are downloaded from the web.

To evaluate whether the shallow parser output improves or not after supplying multiword expression information, we exploit an automatic approach. The approach allows experimenting on large amounts of unannotated data keeping human contribution to a minimum and is based on two naturally emerging hypotheses, which are also manually tested on a small dataset. Differences in the shallow parse output are classified into a number of change classes; while the properties of each class allow assessing its contribution to the final result as positive or negative. Experiments show that knowledge about multiword expressions leads to an increase of between 7.5% and 9.5% in accuracy of shallow parsing in sentences that contain these multiword expressions. As expected, the contribution of multiword expression knowledge is larger for non-compositional than for

compositional multiword expressions and for adjective-noun sequences than for noun-noun sequences. Change classes aid in exploiting the types of changes that are mainly responsible for the result so as to explain it.

5.1 Introduction

As discussed in the introductory chapter of the thesis, we believe that direct integration of multiword expression information, especially for non-compositional ones, into other fields and applications of NLP is equally important to multiword expression research, itself. Approaches that utilise the outcomes of research relevant to multiword expressions as parts or components of other NLP applications are very limited in the literature. However, the vast majority of publications that exploit various multiword expression tasks identify as their motivation the potential contribution of multiword expressions to syntax and semantics-related tasks, such as deep and shallow parsing.

Baldwin et al. (2004) report coverage results for the English Resource Grammar (ERG), a broad-coverage precision Head-Driven Phrase Structure Grammar (HPSG). Among others, they have parsed a random sample of 20,000 strings from the written component of the British National Corpus (BNC), to investigate the causes of parse failure. Missing constructions accounted for 39% of the errors, while missing multiword expressions for 8%. The percentages clearly indicate that there is space for improving parse coverage by integrating multiword expressions.

On the other side of the issue, the state-of-the-art parsing systems seem to ignore the fact that treating multiword expressions as syntactic units would potentially increase parsers' accuracy. To the best of our knowledge there are two approaches of integrating multiword expression knowledge in deep parsing. Zhang et al. (2006) adopted a "word with spaces" model (Sag et al. 2002); i.e. represented each multiword expression as a new lexicon entry. They judged multiword expressions candidates using as frequency indicator the number of pages returned by Google when querying for exact match. 311 multiword expressions candidates occurring in 6248 BNC sentences were judged as being multiword expressions. For each distinct occurrence type of each of the 311 multiword expressions occurring more than 5 times an entry was added to the parser lexicon; resulting in 373 entries. Adding these entries led to an increase of 14.4% in coverage.

Villavicencio et al. (2007) argue that although the "word with spaces" approach of

Zhang et al. (2006) enhances parser coverage, the quality of the parser output is linguistically less interesting. Instead, the authors adopt a constructional approach of integrating non-compositional multiword expressions in the parser (Sag et al. 2002). For the head word of each multiword expression one lexical entry is added; e.g. for *"foot the bill"* a new entry is added to allow the reading of *"foot"* as a transitive verb. Villavicencio et al. (2007) evaluated this approach on 30 multiword expressions that were previously tested to be non-compositional. This set was accompanied by a set of 674 sentences, each of which contained at least one of the 30 multiword expressions. To cover these non-compositional multiword expressions, 21 new lexical entries were added to the parser. The parse output before and after adding the new entries was manually evaluated. Grammar coverage increased by 15.6% and grammar accuracy increased by 14.3%. The authors highlight that they achieved an increase in coverage similar to the one achieved by Zhang et al. (2006) by adding significantly less entries per multiword expression; 0.7 versus 1.2.

The result that non-compositional multiword expressions can increase grammar coverage and accuracy agrees with our intuition. However, there are several disadvantages of the approach of Villavicencio et al. (2007): multiword expressions are judged as compositional or not by combining the scores of mutual information, χ^2 and permutation entropy. These measures are reported to correlate well with non-compositionality but imperfectly. Instead we use the method presented in section 4.3 that helps manual annotation of multiword expressions as compositional or non-compositional. Manually annotating parse outputs as correct or wrong is an expensive process and thus the number of multiword expressions in Villavicencio et al. (2007) is kept small, 30. Moreover, the participating multiword expressions are of various types, e.g. phrasal verbs, nominal compounds, institutionalised phrases, making difficult to explain the increase in accuracy; in other words what is happening before adding multiword expressions. We propose an unsupervised evaluation based on classifying the changes before and after adding multiword expression information; allowing for (a) large scale experiments, and (b) analysing the reasons of increase or decrease in accuracy. The classification is based on two intuitional hypotheses that are evaluated on a manually annotated set of instances.

Apart from the approach of Villavicencio et al. (2007), there are several attempts to integrate other forms of lexical semantics into parsing. Bikel (2000) merged the Brown portion of the Penn Treebank with SemCor, and used it to evaluate a generative bilex-

ical model for joint word sense disambiguation and parsing. Penn Treebank is a corpus whose sentences are manually deep-parsed and SemCor is a collection of texts semantically annotated with WordNet senses. Similarly, Agirre et al. (2008) integrated semantic information in the form of semantic classes and observed significant improvement in parsing and prepositional phrase attachment tasks.

Two successful applications of word sense information to parsing are reported in Xiong et al. (2005) and Fujita et al. (2007). Xiong et al. (2005) integrated first-sense and hypernym features in a generative parse model applied to the Chinese Penn Treebank and achieved significant improvement over their baseline model. Fujita et al. (2007) extended this work by implementing a discriminative parse selection model, incorporating word sense information and achieved great improvements as well. There are also several attempts to integrate into parsing selectional preference information (Dowding et al. 1994; Hektoen 1997). Selectional preferences are a representation of semantics, as discussed in subsections 2.2.1.4 and 2.5.1.

In this chapter, we perform an experimental investigation attempting to estimate the contribution of integrating multiword expressions into shallow parsing. We focus on multiword expressions in English that consist of two successive tokens; in particular, *compound nominals* (e.g. lemon tree), *proper names* (e.g. prince Albert) and *adjective-noun constructions* (e.g. red carpet). The reason for this choice is that the majority of multiword expressions in WordNet are of these three classes. In section 5.3, we discuss a variety of ways to automatically evaluate the task and conclude to use a bank of multiword expressions (WordNet) as the starting point of the evaluation setting.

We conclude that even a very simple way of integrating multiword expressions leads to an increase of between 7.5% and 9.5% in accuracy of shallow parsing of sentences containing these multiword expressions. Increase percentages are higher for multiword expressions that consist of an adjective followed by a noun (12% to 15%); and even higher for non-compositional multiword expressions that consist of an adjective and a noun (15.5% to 19.5%). Our experimental outcome that non-compositional multiword expressions clearly improve shallow parsing agree with Villavicencio et al. (2007).

The remaining of this chapter is structured as follows: In Section 5.2 we present how multiword expressions can be annotated in text and used by any shallow parser. In Section 5.3 we present an overview of the evaluation procedure. Section 5.4 explains how the set of target multiword expressions and textual corpora were created. In Section

5.5 we present and discuss the results of the experimental process. Finally, section 5.6 summarises the chapter.

5.2 Annotating multiword expressions

Deep or shallow parsing should treat multiword expression as units that cannot be divided in any way. For non-compositional multiword expressions this is very easy to accept due to the fact that the semantics of the multiword expression is different from the semantics of its components. If the components of a multiword expression were assigned to different phrases in the shallow parsing output, then the semantics of the expression would be replaced by the semantics of the components.

For compositional multiword expressions the argument that if its components are assigned to different phrases then the semantics of the multiword are altered does not hold. However, multiword expression components are still expected to be assigned to the same phrase, since the union of them is not based on semantics, only. A possible assignment of the components of a multiword expression in different phrases would mean that each component is more closely related to the other words in the phrase that it is assigned to and not to the other component. From this point of view, it makes sense to force the shallow parser to assign multiword expressions tokens in the same phrase. However, this choice is incorrect if we consider adding multiword expressions as new units in some lexicon because semantics of compositional multiword expressions are not significantly different from the semantics of their components. For these experiments, we see all multiword expressions derived from WordNet as single units, and we expect the results to verify that this decision is more meaningful for non-compositional than compositional multiword expressions.

Based on the previous arguments we can adopt the following way of integrating multiword expressions annotation into the input text: We replace the multiword expression tokens with a special made up token, i.e. the multiword expression constituents joined with an underscore. In other words we externally force the components to be assigned to the same phrase. For example, we replace all occurrences of “lemon tree” with “lemon_tree”. This approach is similar to the “words with spaces” approach proposed in Sag et al. (2002).

Our choice introduces a new word, that does not exist in the dictionary of the part of speech tagger. This is quite important, because it will trigger whatever special way the

parser has to treat unknown words. Some parsers use back-off models to estimate the probabilities of unknown words. Part of speech taggers usually assign to an unknown words the part of speech that best fits to it with respect to the parts of speech of the words around it and the training data. This expected behaviour of both the part of speech tagger and the parser is desirable for our purposes.

The experimental results of our study quantify the differences between the shallow parsing output of a big number of sentences after the replacement and the shallow parsing output of the same sentences before the replacement. The comparison is done ignoring changes of parts of speech, assigned by the part of speech tagger, since these changes are due to another component.

5.3 Evaluation

The target of the evaluation procedure is to evaluate whether replacing the multiword expression tokens with a single token, unknown to the part of speech tagger, improves shallow parsing accuracy. The ideal way to perform this evaluation would be to use a corpus with manual annotation about parsing and multiword expressions. Given this corpus we would be able to measure the accuracy of a shallow (or deep) parser before and after replacing multiword expressions. However, to the best of our knowledge there is no corpus available to include this type of annotations in English.

Instead, there are two options: Firstly, we can use treebank data, where manual parsing annotation is readily available, and manually annotate multiword expressions. The advantage of this approach is that results are directly comparable with other results of the literature, due to the use of benchmark data. Manual annotation of multiword expressions is a very time- and effort-consuming process due to the large size of most treebanks. Alternatively, multiword expression annotation could be done using a method of recognition. Annotating the multiword expressions that appear in *WordNet* could be a safe decision, in terms of correctness, however, *WordNet* is reported to have limited coverage of multiword expressions (Baldwin 2006; Laporte & Voyatzi 2008).

Secondly, we can use a set of multiword expressions as a starting point and then create corpora that contain instances of these multiword expressions. In succession, these sentences need to be manually annotated in terms of parsing, and this requires huge human effort. Alternatively, we can parse the corpora before and after replacing the multiword expression and then compare the parsing output. This is the evaluation

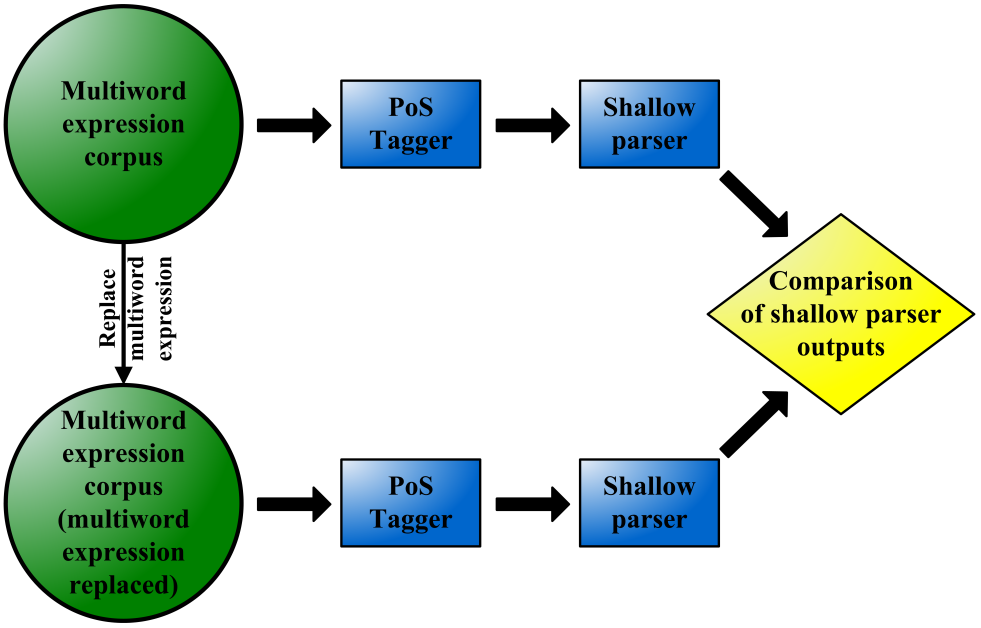

Figure 5.1: Evaluation process

procedure that we follow, and is shown in Figure 5.1.

The chosen evaluation approach compares the parse outputs of the same sentences before and after replacing the multiword expressions which they contain. As a result, it is only able to retrieve instances where the replacement of the multiword expression leads to a different parse output, a different allocation of tokens to phrases. It cannot spot instances where the parsing output remains unchanged after the replacement, no matter if they were correct and remained correct, or they were wrong and remained wrong. Since we are interested in measuring if replacing multiword expressions with a single token improves parsing accuracy, we are not interested in instances that remain unchanged.

Example parse outputs before and after the replacement are presented in two-column format. The left column presents the parse output before the replacement and the right column the parse output after the replacement. Both parses are presented as numbered lists, where *B* stands for "*B*efore the replacement" and *A* stands for "*A*fter the replacement". Each line consists of either a phrase, within square brackets, or a word that is not assigned to any phase in the shallow parsing output. We call the latter words "leaves". Tags outside parentheses denote phrase names: *NP* stands for "Noun Phrase", *VP* for

"Verb Phrase", *PP* for "Prepositional Phrase", *PRT* for "Particle", etc.[1] Tags inside parentheses denote the part of speech of the following word. Part of speech tags whose first letter is *N* or *V* denote nominal or verbal forms, respectively. The following letters denote some subclass of nominal and verbal forms, e.g. *VBD* tags the past form of the verb "be" and *NN* tags common noun. *DT* stands for determiners, *IN* for prepositions, etc.[2]

Suppose the following example sentence[3]:

> They jumped over a bonfire and rolled a fire wheel.

Its parse outputs before and after the replacement of the multiword expressions are:

B 01: [NP (PRP they)]	A 01: [NP (PRP they)]
B 02: [VP (VBD jumped)]	A 02: [VP (VBD jumped)]
B 03: [PP (IN over)]	A 03: [PP (IN over)]
B 04: [NP (DT a) (NN bonfire)]	A 04: [NP (DT a) (NN bonfire)]
B 05: (CC and)	A 05: (CC and)
B 06: [VP (VBD rolled)]	A 06: [VP (VBD rolled)]
B 07: [NP (DT a) (NN **fire**) (NN **wheel**)]	A 07: [NP (DT a) (NN **fire_wheel**)]
B 08: (. .)	A 08: (. .)

We observe that the only difference between the left and the right column is the replacement of the multiword expressions tokens (B07) with a special single token (A07). This means that the parse output has not been affected by the replacement.

In contrast, the sentence:

> The blades ignited and he threw the fire wheel up into the air.

produced the following shallow parse outputs:

B 01: [NP (DT the) (NNS blades)]	A 01: [NP (DT the) (NNS blades)]
B 02: [VP (VBN ignited)]	A 02: [VP (VBN ignited)]
B 03: (CC and)	A 03: (CC and)

[1] A complete list of phrasal tags can be found in Bies et al. (1995), section 2.1.1.

[2] The complete list of part of speech tags, CLAWS-5, can be found at: http://ucrel.lancs.ac.uk/claws5tags.html

[3] All examples of this chapter are taken from real data that were collected following the method described in section 5.4

B 05: [NP (PRP he)]	A 05: [NP (PRP he)]
B 06: [VP (VBD threw)]	A 06: [VP (VBD threw)]
B 07: [NP (DT the) (NN **fire**)]	A 07: [NP (DT the) (NN **fire_wheel**)]
B 08: (WRB **wheel**)	
B 09: (RP up)	A 08: [PRT (RP up)]
B 10: [PP (IN into)]	A 09: [PP (IN into)]
B 11: [NP (DT the) (NN air)]	A 10: [NP (DT the) (NN air)]
B 12: (. .)	A 11: (. .)

We observe that before the replacement *fire* was assigned to an NP together with the preceding determiner *the* (B06). The second component word of the multiword expression, *wheel*, is wrongly part of speech tagged and then remains unassigned to any phrase (B07). Also, the particle *up* of the phrasal verb *threw up* is a leaf (B08). Replacing the multiword expression with a single token, *fire_wheel*, corrects all the above errors (A06,A07).

The evaluation approach focuses on instances whose parse output changed when the multiword expression components are replaced with a special single token. Given that the correct parse output is unique for a given sentence, each sentence either (a) was at first parsed correctly and then parsed wrongly, or (b) was at first parsed wrongly and then correctly, or (c) was parsed wrongly before and after the replacement.

Manually classifying input sentences into the above three categories requires large amounts of effort and expertise, and thus it is very expensive. Alternatively, based on two intuitional hypotheses about the form of correct parse outputs compared to erroneous ones, we identify a number of parse output change classes under which we classify all sentences. Each change class is restricted enough, so as to know whether its instances should be classified under (a), (b) or (c), above.

5.3.1 Shallow parsing change classes

In this section, we present a classification of cases where the shallow parsing output of the sentence is different from the parsing output of the same sentence after replacing the multiword expression with a single token. Secondly, we discuss whether the specific form of each change class can lead to a safe conclusion about whether the parsing output of the sentence under discussion: (a) was wrong before the replacement and was then corrected, (b) was correct before the replacement and was then made wrong, or (c) was

Change classes	Set 1	Set 2	Set 3	Set 4	Set 5
P2LMw	80%	100%	100%	100%	100%
P2L	100%	90%	100%	90%	80%
L2PMw	100%	100%	100%	100%	100%
L2P	100%	80%	90%	90%	100%
PL2P	90%	90%	80%	80%	70%
P2PL	80%	80%	80%	70%	90%
P2P	90%	100%	90%	90%	90%
MwA	100%	100%	100%	100%	100%
Set average	93%	93%	93%	90%	91%
Total average	**91.75%**				

Table 5.1: Cross validation datasets assessing the validity of both hypotheses together.

wrong before the replacement and remained wrong. For this discussion we hypothesise that among the possible output shallow parses for a given sentence the correct one has (a) the smallest number of phrases, and (b) the smallest number of tokens not assigned to any phrase.

> **Hypothesis 1:** Among the possible output shallow parses for a given sentence the correct one has the smallest number of phrases.
>
> **Hypothesis 2:** Among the possible output shallow parses for a given sentence the correct one has the smallest number of leaves.

These hypotheses are based on a number of theoretical intuitions for shallow parsing. Generally, words that are not assigned to any phrase in the parse output, i.e. leaves, are not desirable. They indicate that the corresponding full parse trees are partial and hence should not be preferred over complete parse trees. Also, in phrasal level mistaken parse trees are generally larger, with more phrases.

To strengthen the intuitional arguments, we checked the hypotheses by manually annotating 400 randomly chosen instances; 50 for each change class (see below). We used 5-fold cross validation; i.e. 5 disjoint sets of 10 instances per change class (table 5.1). We counted as positive towards the verification of both hypotheses sentences whose shorter parse output, in terms of phrases or leaves, was manually checked as correct. In contrast we counted as negative sentences whose correct parse output was not the

Class Name	Short description	Contribution
P2LMw	a **P**hrase transformed into **L**eaves that include the **M**ulti**w**ord expression	✗
P2L	a **P**hrase transformed into **L**eaves excluding the multiword expression	✗
L2PMw	**L**eaves transformed into a **P**hrase including the **M**ulti**w**ord expression	✓
L2P	**L**eaves transformed into a **P**hrase excluding the multiword expression	✓
PL2P	**P**hrases or **L**eaves transformed into a **P**hrase	✓
P2PL	a **P**hrase transformed into **P**hrases or **L**eaves	✗
PN	**P**hrase label **N**ame change	*?*
PoS	**P**art **o**f **S**peech tags change	*?*
P2P	**P**hrases transformed into less **P**hrases	✓
MwA	**M**ulti**w**ord expression **A**llocation change	✓

Table 5.2: Summary of the basic change classes. ✓or ✗ denote change classes that count positively or negatively towards improving shallow parsing. *?* denotes classes that are treated specially.

shortest one. The total accuracy is 91.75%, thus, the two hypotheses together are judged as marginally correct, and can be used as a basis for this unsupervised evaluation method.

Table 5.2 summarises all basic change classes, briefly describes each one and shows the contribution of each one towards the final result, improvement in parsing. Change classes that describe changes that lead to shorter shallow parses have a positive contribution while change classes whose changes lead to longer parses have a negative contribution. Below, we present change classes one by one, accompanied with examples.

5.3.1.1 Change class *P2LMw*

This change class includes sentences whose corresponding parse is shorter before the replacement, thus the sentences are counted negatively towards shallow parsing improvement using multiword expression information. Before replacing the multiword expression sequence with a single token, the multiword expression is assigned to some phrase, possibly together with other words. After the replacement, the components of that phrase are not assigned to any phrase, but instead appear as leaves. The class name stands for

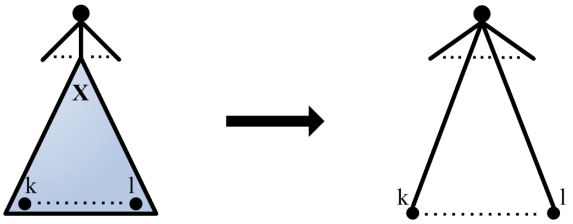

Figure 5.2: Change class *P2LMw*.

"a ***P***hrase was transformed to ***L***eaves which include the ***M***ulti***w***ord expression". Figure 5.2 represents the change class following the notation of Bille (2005). Triangles denote phrases and uppercase bold letters $V...Z$ denote phrase labels. Lowercase letters $k...n$ denote parsing leaves. As an example of change class *P2LMw*, suppose the following input sentence:

> The action officer logistic course is designed to educate and train military and civilian personnel in the logistics staff processes from translation of requirements t set army logistics goals through development of plans, programs and acquisition of the army's equipment.

Part of speech tagging and shallow parsing this sentence before and after the replacement leads to[4]:

B 01: [NP (DT the) (NN **action**) (NN **officer**)]	A 01: (DT the)
	A 02: (NN **action_officer**)
B 02: [NP (JJ logistic) (NN course)]	A 03: [NP (JJ logistic) (NN course)]
B 03: [VP (VBZ is) (VBN designed)]	A 04: [VP (VBZ is) (VBN designed)]
B 04: [VP (TO to) (VB educate)]	A 05: [VP (TO to) (VB educate)]
⋮	⋮

We observe that after replacing the multiword expression, the noun phrase that contains the determiner and the multiword expression tokens disappeared (B01,A01-A02). Thus, a phrase that includes the multiword expression transformed in leaves.

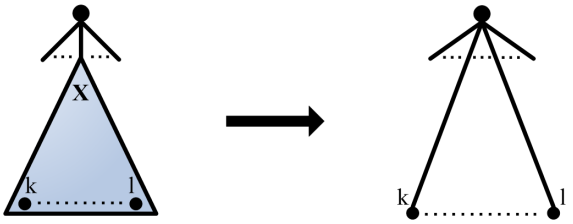

Figure 5.3: Change class *P2L*.

5.3.1.2 Change class *P2L*

The class name stands for "a ***P***hrase was transformed to ***L***eaves which exclude the multiword expression". Similarly to change class *P2LMw*, this class contains sentence whose parse is longer after the replacement or consists of more phrases and less leaves, so these sentences contribute negatively to the final result. Before the replacement, some successive tokens excluding the multiword expression itself are assigned to some phrase. After the replacement, the components of that phrase appear as leaves (figure 5.3). For example, suppose the following sentence:

The agreement on the action officer in Armenia was signed on may 11th 2005.

Before and after the replacement, the sentence is parsed as follows:

B 01: [NP (DT the) (NN agreement)]	A 01: [NP (DT the) (NN agreement)]
B 02: [PP (IN on)]	A 02: [PP (IN on)]
B 03: [NP (DT the) (NN **action**) (NN **officer**)]	A 03: [NP (DT the) (NN **action_officer**)]
B 04: [PP (IN in)]	A 04: (IN in)
B 05: [NP (NN Armenia)]	A 05: [NP (NN Armenia)]
⋮	⋮

We observe that the prepositional phrase containing preposition *in* (B04) turned into a leaf (A04).

[4]Three dots in vertical alignment within parse output lists are used to designate that there is more output but it is omitted, since there is no change in the parse output of this part before or after incorporating multiword expression information.

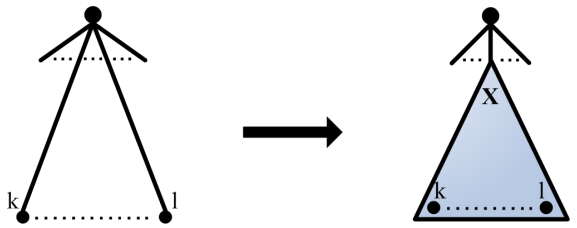

Figure 5.4: Change class *L2PMw*.

5.3.1.3 Change class *L2PMw*

The changes covered by this class are the opposite changes of change class *P2LMw*. The class name stands for "a number of ***L***eaves were transformed to a ***P***hrase which includes the ***M***ulti***w***ord expression". The parse output of the sentences of this class is shorter after the replacement, thus it answers positively to whether multiword expressions improve shallow parsing. Before the replacing the multiword expression sequence with a single token, the multiword expression sequence is not assigned to any phrase possibly among other words. After the replacement, the multiword expression is assigned to a phrase (figure 5.4). Suppose the following phrase:

> "affirmative action officer" AAO refers to the regional affirmative action officer, director, or designee, whichever reference is applicable.

It is shallow parsed before and after the replacement as follows:

B 01: (" ")
B 02: (JJ affirmative)
B 03: (NN action)
B 04: (NN officer)
B 05: (" ")
B 06: [NP (NN aao)]
B 07: [VP (VBZ refers)]
B 08: [PP (TO to)]
B 09: [NP (DT the) (JJ regional) (JJ affirmative) (NN **action**) (NN **officer**)]
⋮

A 01: (" ")
A 02: [NP (JJ affirmative) (NN action_officer)]

A 03: (" ")
A 04: [NP (NN aao)]
A 05: [VP (VBZ refers)]
A 06: [PP (TO to)]
A 07: [NP (DT the) (JJ regional) (JJ affirmative) (NN **action_officer**)]
⋮

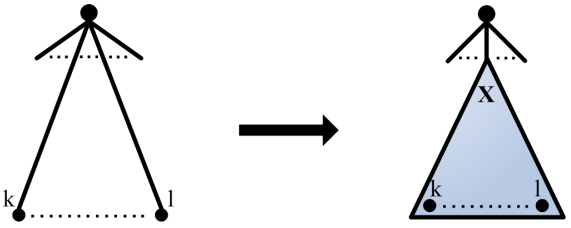

Figure 5.5: Change class *L2P*.

Three leaves including the multiword expression (B02-B04) were after the replacement assigned to a noun phrase (A02).

5.3.1.4 Change class *L2P*

The class name stands for "a number of ***L***eaves were transformed to a ***P***hrase which excludes the ***M***ulti***w***ord expression". Similarly to change class *L2PMw*, before the replacement, one or more successive tokens excluding the multiword expression itself appear as leaves. After the replacement, these tokens are assigned to a phrase (figure 5.5). The class members contribute positively towards the result, because the shallow parse after the replacement is shorter or consists of more phrases and less leaves. The following sentence:

> The action officer usually delivers the execution orders to each of the implementing service operations, whether it be a theatre commander or a task force operator such as the admiral in charge of a carrier battle group.

corresponds to the following shallow parses before and after the replacement:

B 01:	[NP (DT the) (NN **action**) (NN **officer**)]	A 01:	[NP (DT the) (NN **action_officer**)]
B 02:	(RB usually)	A 02:	[ADVP (RB usually)]
B 03:	[VP (VBZ delivers)]	A 03:	[VP (VBZ delivers)]
B 04:	[NP (DT the) (NN execution) (NNS orders)]	A 04:	[NP (DT the) (NN execution) (NNS orders)]
	⋮		⋮

The adjective *usually* which was initially a leaf (B02), was assigned to an adverbial phrase after the replacement (A02).

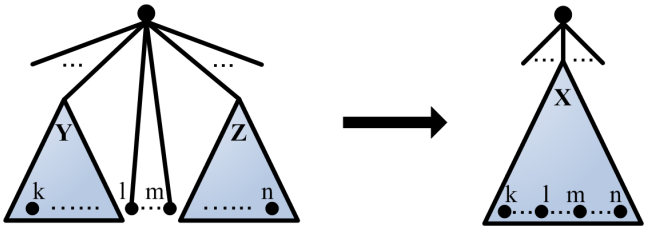

Figure 5.6: Change class *PL2P*.

5.3.1.5 Change class *PL2P*

The class name stands for "a number of ***P***hrases or ***L***eaves were transformed to a single ***P***hrase". After the replacement, the tokens of more than one phrases or leaves are assigned to a single phrase (figure 5.6). Due to the reduction in number of phrases, the sentences of this class contribute positively towards improving shallow parsing accuracy using multiword expression knowledge. Suppose the following phrase:

> The university of Texas-pan American encourages any person who believes that he or she has been subjected to discrimination to immediately report the incident to the equal opportunity and affirmative action officer.

The crucial part at the end of the sentence is parsed as follows before and after the replacement:

⋮	⋮
B 01: [NP (DT the) (NN incident)]	A 01: [NP (DT the) (NN incident)]
B 02: [PP (TO to)]	A 02: [PP (TO to)]
B 03: [NP (DT the) (JJ equal) (NN opportunity)]	A 03: [NP (DT the) (JJ equal) (NN opportunity) (CC and) (JJ affirmative) (NN **action_officer**)]
B 04: (CC and)	
B 05: [NP (JJ affirmative) (NN **action**) (NN **officer**)]	
B 06: (. .)	A 04: (. .)

The constituents of the noun phrases in lines B03 and B05 as well as the leaf in line B04 are correctly assigned to a single noun phrase, in line A03.

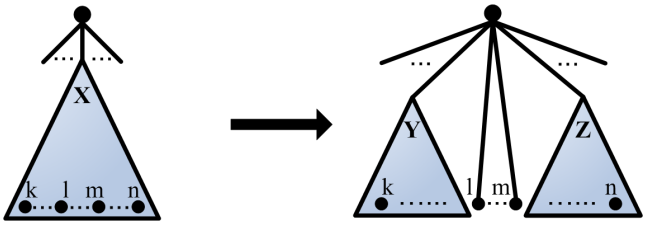

Figure 5.7: Change class *P2PL*.

5.3.1.6 Change class *P2PL*

In contrast to change class *PL2P*, in this class the tokens of one phrase either are assigned to more than one phrases or appear as leaves after the replacement (figure 5.7). The class name stands for "a ***P***hrase was transformed to a number of ***P***hrases or ***L***eaves". The sentences of this class contribute negatively to the overall result, since the number of phrases and leaves in the parse output are more after the replacement. For example, suppose the following sentence:

> The action officer shall prepare and transmit within five working days an evaluation report addressed to the assistant ombudsman for public assistance and corruption prevention office through the bureau of resident ombudsmen for appropriate action.

As shown below, after the replacement, the constituents of the verbal phrase in line B02 are assigned to two verbal phrases in lines A02 and A04 or appear as leaves, in line A03:

B 01:	[NP (DT the) (NN **action**) (NN **officer**)]	A 01:	[NP (DT the) (NN **action_officer**)]
B 02:	[VP (MD shall) (VB prepare) (CC and) (VB transmit)]	A 02:	[VP (MD shall) (VB prepare)]
		A 03:	(CC and)
		A 04:	[VP (VB transmit)]
B 03:	[PP (IN within)]	A 05:	[PP (IN within)]
	⋮		⋮

5.3.1.7 Change class *PN*

This change class does not describe a change in the structure of the parse output but instead a "***P***hrase label ***N***ame change". After replacing the multiword expression sequence with a single token, one phrase appears with a different phrase label, although it retains exactly the same component tokens (figure 5.8). Since this type of change is not covered

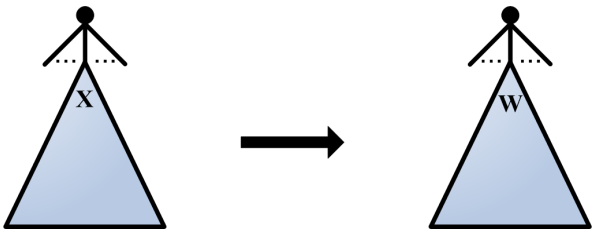

Figure 5.8: Change class *PN*.

by the hypotheses about comparing parse outputs in terms of the length and structure, we leave the sentences of this class out of the final result computation. In a best case scenario, all phrase label changes would be correct while in a worst case scenario all changes would be wrong. Suppose the following sentence:

> Captain Rasmussen, a 1996 graduate of land O'Lakes High School in Land O'Lakes, Florida, is an action officer for the JTF-CS communications directorate, lending his expertise for a February JTF-CS communications exercise with local emergency responders.

The parse outputs before and after the replacement are:

⋮	⋮
B 01: [VP (VBZ is)]	A 01: [VP (VBZ is)]
B 02: [NP (DT an) (NN **action**) (NN **officer**)]	A 02: [NP (DT an) (NN **action_officer**)]
B 03: [PP (IN for)] [NP (DT the) (NNS jtf-cs) (NNS communications) (NN directorate)]	A 03: [SBAR (IN for)] [NP (DT the) (NNS jtf-cs) (NNS communications) (NN directorate)]
B 04: (, ,)	A 04: (, ,)
⋮	⋮

We observe that the prepositional phrase (B03) after the replacement is tagged as a clause introduced by a subordinate conjunction, *SBAR* (A03).

5.3.1.8 Change class *PoS*

This class contains sentences whose shallow parses before and after the replacement are the same in terms of structure. However, after replacing the multiword expression

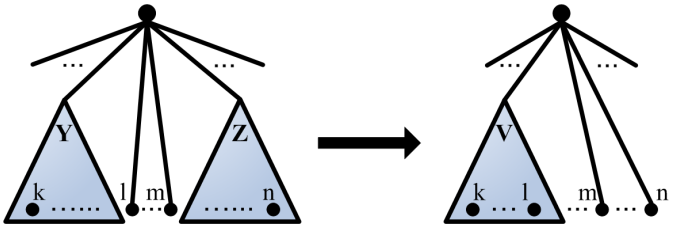

Figure 5.9: Change class *P2P*.

sequence with a single token, one or more tokens appear with a different part of speech. The class name stands for "***P***art ***o***f ***S***peech change". Suppose the following sentence:

> He helps the beaten student get to the school's security officer or "youth action officer".

The parse outputs before and after the replacement for this sentence are:

⋮	⋮
B 01: (CC or)	A 01: (CC or)
B 02: (" ")	A 02: (" ")
B 03: [NP (JJ youth) (NN **action**) (NN **officer**)]	A 03: [NP (NN youth) (NN **action_officer**)]
B 04: (" ")	A 04: (" ")
B 05: (. .)	A 05: (. .)

We observe that before the replacement *youth* is tagged as an adjective, JJ (B03), while after the replacement it is tagged as noun, NN (A03).

This change class accounts for changes that depend entirely on the part of speech tagger. The shallow parser is not affected by these changes. Since the scope of this study is to quantify the improvement in shallow parsing accuracy only, we do not include the changes of this class towards the final result. However, in the results section we show a size estimate of this class.

5.3.1.9 Change class *P2P*

The class name stands for "some ***P***hrases were transformed to less ***P***hrases". After replacing the multiword expression sequence with a single token, the component tokens of

more than one successive phrases α are assigned to a different set of successive phrases β. However, it is always the case that phrases α are less than phrases β ($|\alpha| < |\beta|$) (figure 5.9). Due to this inequality, sentences of this change class are counted positively towards the improvement in shallow parsing after integrating multiword expression knowledge. As an example, suppose the following sentence:

> Rabbi, as a past action officer and command and control and intelligence communications inspector on the joint staff for six years above top secret, I have some critical and timely information for you. The parse outputs before and after replacing the multiword expression tokens are the following:

B 01: [NP (NNS Rabbi)]	A 01: [NP (NNS Rabbi)]
B 02: (, ,)	A 02: (, ,)
B 03: [PP (IN as)]	A 03: [PP (IN as)]
B 04: [NP (DT a) (JJ past) (NN **action**) (NN **officer**)]	A 04: [NP (DT a) (JJ past) (NN **action_officer**) (CC and) (NN command) (CC and) (NN control)]
B 05: (CC and)	
B 06: (NN command)	
B 07: (CC and)	
B 08: (NN control)	
B 09: (CC and)	A 05: (CC and)
B 10: [NP (NN intelligence) (NNS communications) (NN inspector)] [PP (IN on)]	A 06: [NP (NN intelligence) (NNS communications) (NN inspector)] [PP (IN on)]
⋮	⋮

One noun phrase (B04) and 4 successive leaves (B05-B08) in the parse output before the replacement are transformed into one noun phrase (A04).

5.3.1.10 Change class *MwA*

The class name stands for "***M***ulti***w***ord expression ***A***llocation change". Before replacing the multiword expression sequence, the multiword expression constituents are assigned to different phrases (figure 5.10). Clearly, before the replacement the shallow parse output violates the rule that the constituents of the multiword expressions consist a semantic and syntactic unit. Therefore, the sentences of this change class contribute positively towards the final result. Suppose the following sentence:

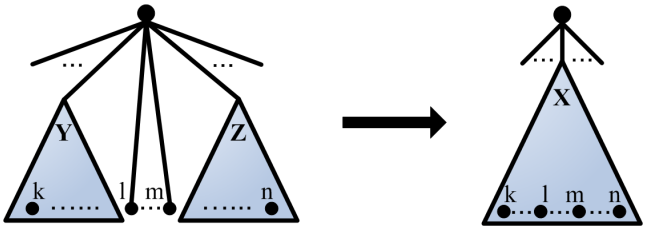

Figure 5.10: Change class *MwA*.

> The campus affirmative action officer serves as the liaison between the Parkersburg campus and West Virginia University, providing campus supervision of hiring processes and providing a central point of contact for claims of discrimination.

The shallow parse outputs before and after replacing the multiword expression tokens with a single one are:

B 01: (DT The)
B 02: (NN campus)
B 03: (JJ affirmative)
B 04: (NN **action**)
B 05: [NP (NN **officer**)]
B 06: [VP (VBZ serves)]
⋮

A 01: [NP (DT The) (NN campus) (JJ affirmative) (NN **action_officer**)]

A 02: [VP (VBZ serves)]
⋮

We observe that 4 leaves (B01-B04) and a noun phrase (B05) in the parse output before the replacement are assigned to a single noun phrase (A01) after the replacement. The multiword expression tokens, *action* and *officer*, were not assigned to the same phrase, initially. ■

To summarise, change classes are introduced to simplify quantifying the improvement that multiword expression integration causes to shallow parsing (table 5.2). According to our hypotheses, change classes with sentences whose parse after the replacement is shorter or contains more phrases and less leaves than before the replacement contribute positively towards the result. These change classes are: *L2PMw*, *L2P*, *PL2P* and *P2P*. In contrast, change classes with sentences whose parse is longer after the replacement or contains more leaves and less phrases contribute negatively. These change

classes are: *P2LMw*, *P2L* and *P2PL*. The instances of change class *PN* can be either correct or wrong after the replacement, since they are not covered by the evaluation hypothesis. They are treated specially in the computation of the result.

5.3.2 Shallow parsing complex change classes

During the inspection of instances where changes occur in the shallow parsing output when replacing the multiword expression with a special token, we came across a number of instances that could be classified in more than one of the classes of the previous subsection. In other words, two or more change classes were happening at the same time. For example, in a number of instances, before the replacement, the multiword expression constituents are assigned to different phrases (change class *MwA*). After the replacement, the tokens of more than one phrases are assigned to a single phrase (change class *PL2P*). These instances consist of new complex changes and are named as the sum of names of the participating classes. The instances of the example above consist the complex change class *PL2P+MwA*. In the result section, we present separately statistics about two complex change classes that occurred in the data, *PL2P+MwA* and *P2P+MwA*. Sentences classified under these classes are considered as positive towards the result, because both the simple change classes that they consist of contribute positively as well.

5.4 Target multiword expressions and corpora collection

We created an experimental set of multiword expressions using *WordNet 3.0* (Miller 1995). *WordNet 3.0* contains $52,217$ multiword entries from which 120 were randomly chosen. Keeping the ones that consist of two tokens resulted in the 116 expressions of tables 5.3 and 5.4. Manually inspecting these multiword expressions proved that they are all *compound nominals*, *proper names* or *adjective-noun constructions*.

To annotate each multiword expressions for compositionality, we follow the procedure presented in section 4.3. To summarise it briefly, the semantic head of each multiword expression was located. Then the synsets within a neighbourhood of the synsets in which the multiword expression occurs were inspected. Finding the semantic head in the neighbour synsets gives evidence that the expression is most likely compositional. Otherwise, the multiword expressions is tagged as potentially non-compositional. Finally, the above annotations are manually checked. Tables 5.3 and 5.4 show the chosen

Compositional Multiword expressions (Noun - Noun sequences)		
action officer (3119)	beach towel (1937)	bile duct (21649)
car battery (3726)	cartridge brass (479)	checker board (1280)
corn whiskey (1862)	corner kick (2882)	cream sauce (1569)
field mushroom (789)	flight simulator (5955)	honey cake (843)
jazz band (6845)	jet plane (1466)	key word (3131)
king snake (2002)	labor camp (3275)	laser beam (16716)
lemon tree (3805)	life form (5301)	love letter (3265)
luggage van (964)	memory device (4230)	potato bean (265)
power cord (5553)	prison guard (4801)	sausage pizza (598)
savoy cabbage (1320)	surface fire (2607)	tea table (62)
telephone service (9771)	torrey tree (10)	touch screen (9654)
upland cotton (3235)	water snake (2649)	water tank (5158)
wood aster (456)		
Compositional Multiword expressions (Adjective - Noun sequences)		
basic color (2453)	cardiac muscle (6472)	closed chain (1422)
common iguana (668)	cubic meter (4746)	eastern pipistrel (128)
graphic designer (8228)	hard candy (2357)	ill health (2055)
kinetic theory (2934)	male parent (1729)	medical report (3178)
musical harmony (1109)	mythical monster (770)	red fox (10587)
relational adjective (279)	parking brake (7199)	petit juror (991)
taxonomic category (1277)	thick skin (1338)	toxic waste (7220)
universal donor (1454)	parenthesis-free notation (113)	

Table 5.3: 60 compositional multiword expressions randomly chosen from *WordNet*; 37 Noun - Noun sequences and 23 Adjective Noun sequences. The size of the respective corpus in sentences appears within parentheses.

multiword expressions together with information about their compositionality and the parts of speech of their components.

For each multiword expression we created a different corpus by downloading sentences from the web. Each corpus consists of webtext snippets of length 15 to 200 tokens in which the multiword expression appears. Snippets were collected following the process presented in 4.2.1. In brief, given a multiword expression, a set of queries is created: All synonyms of the multiword expression extracted from WordNet are collected. For example, the synonyms of “red carpet” are “rug”, “carpet” and “carpeting”. The multi-

Non-Compositional Multiword expressions (Noun - Noun sequences)		
agony aunt (751)	air conditioner (24202)	band aid (773)
fire brigade (5005)	fire wheel (480)	fish finger (1423)
lip service (3388)	monkey puzzle (1780)	motor pool (3184)
oyster bed (1728)	pack rat (3443)	palm reading (4428)
paper chase (1115)	paper gold (1297)	paper tiger (1694)
picture palace (2231)	pill pusher (924)	pine knot (1026)
powder monkey (1438)	prince Albert (2019)	rat race (2556)
road agent (1281)	sea lion (9113)	spin doctor (1267)
vegetable sponge (806)	winter sweet (460)	
Non-Compositional Multiword expressions (Adjective - Noun sequences)		
black maria (930)	dead end (5256)	dutch oven (4582)
golden trumpet (607)	green light (5960)	high jump (4455)
holding pattern (3622)	joint chiefs (2865)	living rock (985)
magnetic head (2457)	missing link (5314)	personal equation (873)
personal magnetism (2869)	petit four (1506)	pink lady (1707)
pink shower (351)	poor devil (1594)	public eye (3231)
quick time (2323)	red devil (2043)	red dwarf (6526)
red tape (2024)	round window (1380)	silent butler (332)
small beer (2302)	small voice (4313)	stocking stuffer (7486)
sweet bay (1367)	teddy boy (2413)	think tank (4586)

Table 5.4: 56 non-compositional multiword expressions randomly chosen from *WordNet*; 26 Noun - Noun sequences and 30 Adjective Noun sequences. The size of the respective corpus in sentences appears within parentheses.

word expression is paired with each synonym to create a set of queries. For each query, snippets are collected by parsing the web-pages returned by *Yahoo!*. The union of all snippets produces the multiword expression corpus. In Tables 5.3 and 5.4, the number of collected corpus sentences for each multiword expression are shown within parentheses. *GENIA* tagger (Tsuruoka et al. 2005) was used as part of speech tagger.

The state-of-the-art *SNoW-based Shallow Parser* of Munoz et al. (1999) was used for shallow parsing. Sparse Network of Winnows (SNoW) is a learning architecture based on a sparse network of linear functions over a pre-defined or incrementally learnt feature space. It is particularly useful for learning in domains with sparsity; where there is a vast number of features but only a small number of them is active in each instance. This is

because the number of examples required to learn a linear function grows linearly with the number of relevant features and logarithmically with the number of total features. The SNoW network consists of predictors, i.e. computations of simple relations over the input sentence, that can be chained and combined together to produce an estimate of the function of interest. Learning is online and mistake-driven.

The shallow parser uses as features the surface forms of the words in the input sentence and their parts of speech. The predictors are used to decide for each word whether it belongs to a phrase or not (inside/outside predictors). This approach to shallow parsing is reported to compete favourably with other approaches in the literature. Error analysis showed that its accuracy decreases while the length of phrases increases. Moreover, the authors argue that most mistaken predictions happen in cases of conjunctions, gerunds, adverbial noun phrases and some punctuation marks.

Our experiments of integrating multiword expressions into shallow parsing are potentially compatible with any shallow parsers available. However, since the internal structure of a shallow parser affects the quality of its results, using a poor-performing parser can lead to different results in terms of numbers.

5.5 Experimental results and discussion

The corpora collecting procedure of section 5.4 resulted in a corpus of $376,007$ sentences, each one containing one or more multiword expressions. In $85,073$ sentences (22.75%), the shallow parsing output before the replacement is different to the shallow parsing output after the replacement. The corresponding percentage is larger for non-compositional than compositional multiword expressions; 25.33% and 18.17%, respectively. Likewise, more changes happen for adjective-noun than for noun-noun multiword expressions; 28.76% and 16.06%, respectively. All change percentages are presented in the third and fourth column of table 5.5.

7.20% of all $85,073$ change instances are due to one or more parts of speech changes, and are classified to change class *PoS*. In other words, in 7.20% of cases where there is a difference between the shallow parses before and after replacing the multiword expression tokens there is one or more tokens that were assigned a different part of speech. However, excluding parts of speech from the comparison, there is no other difference between the two parses. The focus of this study is to quantify the effect of unifying multiword expression tokens in shallow parsing. Part of speech tagging is a component

of our approach and parts of speech are not necessarily parts of the parsing output. We included them in the examples for presentational reasons, mostly. For this reason, we chose to ignore part of speech changes, the changes of change class *PoS*. Below, we present our results for all other change classes.

Let $\|X\|$ be the function that returns the number of instances assigned to change class X. With respect to the discussion of subsection 5.3.1 about how the instances of each class should be counted towards the final results, the number of sentences whose parsing was corrected after the replacement is the sum of sentences of the change classes that contribute positively, according to the evaluation hypotheses:

$$\begin{aligned} positive \quad = \quad & \|L2PMw\|+\|L2P\|+\|PL2P\|+\|P2P\|+ \\ & \|PL2P+MwA\|+\|P2P+MwA\| \end{aligned} \tag{5.1}$$

In contrast, the sum of sentences of change classes that contribute negatively towards to result according to the evaluation hypotheses is:

$$negative = \|P2LMw\|+\|P2L\|+\|P2PL\|$$

To compute the minimum improvement in shallow parsing, positively contributions classes are computed positively, negatively contributing classes are computed negatively and the undecidable class *PM* negatively as well:

$$min = positive - negative - \|PN\| \tag{5.2}$$

For the maximum improvement, the undecidable class *PM* is computed as contributing positively:

$$max = positive - negative + \|PN\| \tag{5.3}$$

Table 5.5 summarises experimental results for adjective-noun and noun-noun sequences, either compositional or non-compositional. The table also presents average statistics for multiword expressions of the same parts of speech independently of their compositionality and of the same compositionality independently of parts of speech. The third column shows the number of sentences of each class in the corpus. For each one

Multiword expressions				Shallow Parsing improvement	
class	*PoS*	sentences	changed	minimum	maximum
Compositional	N N	143,229	15.74%	4.48%	5.47%
Non-Compositional	N N	77,812	16.67%	4.39%	5.84%
Compositional	J N	68,707	23.24%	7.34%	9.21%
Non-Compositional	J N	86,259	33.15%	15.32%	19.67%
Any	N N	221,041	16.06%	4.45%	5.60%
Any	J N	154,966	28.76%	11.78%	15.03%
Compositional	Any	211,936	18.17%	5.41%	6.68%
Non-Compositional	Any	164,071	25.33%	10.14%	13.11%
Any	Any	376,007	21.30%	7.47%	9.49%

Table 5.5: Summary of experimental results. "PoS" stands for parts of speech, "N N" for noun noun sequences and "J N" for adjective noun sequences.

of the classes of table 5.5, the fifth and sixth columns show the minimum and maximum improvement in shallow parsing, respectively, caused by unifying multiword expression tokens. It should be noted that this improvement are computed on corpora whose all sentences contain at least one known multiword expression. To project this improvement on any general text, one needs to know the percentage of sentences that contain known multiword expressions. Then the projected improvement can be computed by multiplying these two percentages.

On average of all multiword expressions, unifying multiword expression tokens contributes from 7.47% to 9.49% in shallow parsing accuracy. Both for noun-noun and adjective-noun multiword expressions, non-compositional ones improve accuracy more than compositional ones do, due to the idiosyncratic nature of non-compositional multiword expressions. However, the improvement is much larger for non-compositional adjective-noun multiword expressions.

Recall that the shallow parser accuracy is reported to decrease as the length of phrases increases. For this reason, we expect some increase in parsing accuracy due to the fact that we reduce by one the length of the phrases which contain the multiword expressions. To assess the level of accuracy improvement due to this phrase length decrease, we introduce the following baseline. Using the corpora already collected, we

Random sequences with same PoS			Shallow Parsing improvement	
PoS	sentences	changed	minimum	maximum
N N	221,041	10.83%	3.21%	3.96%
J N	154,966	16.49%	5.26%	5.76%
Any	376,007	13.17%	4.05%	4.70%

Table 5.6: Summary of experimental results - Baseline of random sequences. "PoS" stands for parts of speech, "N N" for noun-noun sequences and "J N" for adjective-noun sequences.

assess the improvement in shallow parsing when we replace a random two-word sequence of specific parts of speech with a special made-up token. In noun-noun multiword expressions corpora we replace a random noun-noun sequence in each sentence. In adjective-noun corpora we replace a random adjective-noun sequence in each sentence.

Table 5.6 presents the results of the baseline. We observe that the improvements are as expected smaller than the improvements when multiword expressions are replaced. The differences among the improvements for random noun-noun sequences, compositional noun-noun multiword expressions and non-compositional noun-noun expressions are not large, thus integrating noun-noun multiword expressions does not improve shallow parsing significantly. The difference is also small between random adjective-noun sequences and compositional adjective-noun multiword expressions. In contrast, there is large increase in accuracy between compositional and non-compositional adjective-noun multiword expressions, thus adjective-noun non-compositional multiword expressions are clearly worth considering.

Figure 5.11 shows the percentage of each change class over the sum of sentences whose parse output before unifying multiword expression tokens is different for the parsing output after the replacement. The ravdogram shows percentages for compositional and non-compositional noun-noun sequences, adjective-noun sequences and on average. The most common change class for all multiword expression categories is *PL2P*, accounting for 34.03% of the changes. It contains instances where after replacing the multiword expression with a single made-up token a number of phrases and leaves are assigned to a single phrase. This class contributes positively to the result according to the evaluation hypothesis. Classes of medium frequency, around 10%, are *P2LMw*, *P2PL*,

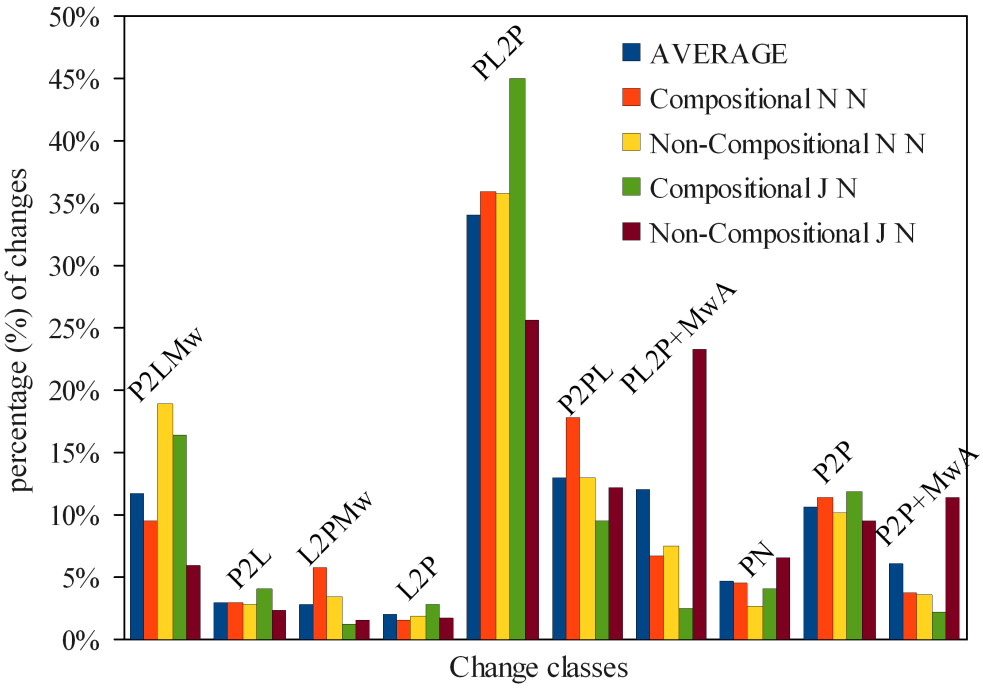

Figure 5.11: Change percentages per change class on average and per multiword expressions category. "N N" stands for noun-noun sequences and "J N" for adjective-noun sequences.

PL2P+MwA and *P2P*. In contrast, *P2L*, *L2PMw L2P*, *PN* and *P2P+MwA* are infrequent, around 5%.

In figure 5.11 we observe that compositional and non-compositional noun-noun sequences do not differ much in any change class. This explains why there is no significant increase on the overall improvement between these multiword expressions. The distribution of changes is very different for adjective-noun sequences; *P2LMw* and *PL2P* account for 10% and 20% more changes for compositional against non-compositional adjective-noun multiword expressions, respectively. In opposition, *PL2P+MwA* and *P2P+MwA* account for 20% and 10% more changes for non-compositional against compositional adjective-noun multiword expressions, respectively. This shows that non-compositional adjective-noun multiword expressions behave differently than compositional ones.

Since there are 10% more changes happening for non-compositional adjective-noun multiword expressions (table 5.3), the change distribution differences can explain why this class of multiword expressions leads to the largest improvement. The components of non-compositional adjective-noun multiword expressions are commonly assigned into different phrases (change class MWA), and this mistake is corrected after the replacement of the multiword expression with a single token.

Recall that Munoz et al. (1999) reported that the shallow parser is commonly mistaken when the phrases contain conjunctions, gerunds and adverbial noun phrases. Our results agree since there is one gerund among compositional adjective-noun multiword expressions, i.e. *parking brake* against 4 among non-compositional adjective-noun multiword expressions, i.e. *holding pattern*, *living rock*, *missing link* and *stocking stuffer*. Moreover, error analysis revealed that conjunctions between two noun phrases are very common among instances of change classes *PL2P* and *PL2P+MwA*.

5.6 Chapter Summary

In this chapter, we presented an experimental study attempting to estimate the contribution of information about multiword expressions into shallow parsing. The integration was realised by replacing the multiword expression components with a single made-up token, unknown to the part of speech tagger and the shallow parser.

The evaluation is done based on 116 multiword expressions extracted from *WordNet 3.0* that consist of two successive components and are *compound nominals*, *proper names* or *adjective-noun constructions*. In particular, the above multiword expressions set consists of 26 non-compositional and 37 compositional noun-noun sequences and 30 non-compositional and 23 compositional adjective-noun sequences. For each multiword expression we collected sentences that contain it from the web. These corpora were part of speech tagged and shallow parsed before and after integrating multiword expression information.

The outputs were compared employing a detailed classification of changes. Based on two intuitionally emerging hypotheses, whose validity is manually checked, each change class is considered as correcting erroneous parses or as making correct parses incorrect. This classification of changes in the parse output contributes in two ways: (a) it allows overcoming human annotation of parse outputs but still being able to quantify if multiword expression information improve shallow parsing accuracy. (b) it provides

with statistics of changes useful to trace where the increase or decrease in accuracy comes from.

We presented experimental results about how often instances of each change class occur on average and for each multiword expression category, separately. Change class frequency counts are combined to estimate the overall improvement in shallow parsing accuracy with respect to compositionality and the parts of speech of multiword expression components. Integrating multiword expression information leads to an increase of between 7.5% and 9.5% in shallow parsing accuracy of sentences that contain these multiword expressions. Increase percentages for compositional noun-noun sequences are between 4.5% and 5.5%, for non-compositional noun-noun sequences between 4.4% and 5.8%, for compositional adjective-noun sequences between 7.3% and 9.2% and for non-compositional adjective-noun sequences between 15.3% and 19.7%.

It is reported that the decrease in phrase length can by itself increase shallow parsing accuracy (Munoz et al. 1999). Therefore, this increase should not be considered as a contribution of multiword expressions. To estimate the size of this increase we introduced a baseline: In each sentence we replaced a random noun-noun or adjective-noun sequence and estimated the increase in shallow parsing accuracy as previously. The increase for random noun-noun sequences was computed between 3.2% and 4.0% and for random adjective-noun sequences between 5.3% and 5.8%. Comparing the baseline improvements to the multiword expression ones reveals that the increase is statistically significant for adjective-noun multiword expressions, only, according to McNemar's test and Fisher's exact test.

In our analysis, we excluded the contribution of part of speech tagging in the final result. In other words, our analysis does not conclude what part of the increase is due to improvements in parts of speech tagging and what part is due to parsing improvement. In some cases, the parse output might improve because some words were assigned a different part of speech during part of speech tagging. It would be interesting to divide the increase into separate part of speech tagging and shallow parsing contributions.

Another extension could be to perform the same experiments using a different shallow parser. This would help to assess the extent to which the outcomes of this research generalise and how much they depend on the chosen shallow parser. However, the fact that the shallow parser in use is among the state-of-the-art guarantees that the results will not be detrimentally different. Similarly, it would be interesting to adapt the proposed

evaluation procedure to deep parsing.

As far as multiword expressions are concerned, we could use our evaluation method to assess the contribution of longer multiword expressions into shallow parsing. One would expect that due to their size, a wrong interpretation of their structure would affect the shallow parsing output more than it does for multiword expressions consisting of two words. Thus, unifying multiword expressions longer than two words would potentially contribute more to shallow parsing accuracy.

CHAPTER 6

Distributional Semantics Composition

Executive Summary

In distributional semantics studies, there is growing attention in determining the meaning of word sequences. The distributional meaning of word sequences is obtained by combining the distributional meanings of component words. State-of-the art models for compositional distributional semantics (CDS) are evaluated on word sequence similarity tests, such as (Kintsch 2001), and lexical substitution tasks. CDS models are reported to show low correlation with human annotated data, for which inter-annotation agreement is also low. This fact reveals that there is need to define a new task for evaluating CDS models.

In this chapter, we propose a novel framework for investigating compositional distributional semantics (CDS). In this framework, a well performing CDS model is expected to compose distributional vectors of multiword expressions that: (a) are similar to the distributional vector of the multiword expression occurring as a whole, given that the multiword expression is compositional; and (b) are dissimilar to the distributional vector of the multiword expression occurring as a whole, given that the multiword expression is non-compositional.

Evaluating existing CDS models on this framework show that it suffers from sparsity; i.e. the chosen multiword expressions occur as a whole very rarely to extract reliable

distributional vectors. To address this problem we propose a new method for extracting evaluation instances. Each instance consists of a word and a sequence. For positive examples, the compositional meaning of the sequence is expected to be similar to the meaning of the word. For negative examples, the compositional meaning of the sequence is expected to be dissimilar to the meaning of the word. Instances were extracted from suitable dictionary definitions.

A large number of state-of-the-art CDS models are evaluated under the proposed evaluation framework. Results indicate that simple addition of the distributional vectors that correspond to the component words of a sequence shows potential. However, there is still space for improvement since the accuracy of existing CDS models highly depends on parameter settings.

In succession we propose an estimation method for the parameter of the basic additive CDS model, based on regression models for multiple dependent variables. The equation system is solved using approximated solutions based on the Moore-Penrose pseudo-inverse. Experiments demonstrate that the CDS model whose parameters are estimated according to the proposed approach outperforms existing CDS models.

6.1 Introduction

In section 2.3 of the literature survey, we presented an introduction to representing context distributions of words. Then, in section 2.3.2, we presented the distributional hypothesis of Harris (1954) which can be operationally defined as: "similar words share similar contexts". The distributional hypothesis allows considering that the distribution of the context of a word in a large corpus is closely related to its semantics. Semantics of the context words within the distribution can securely describe the semantics of the target word. In section 2.3.3, we described various measures that can be employed to compare context distributions and compute whether and how much semantically related the corresponding words are.

Lexical distributional semantics has been largely used to model word meaning in many fields as computational linguistics (McCarthy & Carroll 2003; Manning et al. 2008), linguistics (Harris 1964), corpus linguistics (Firth 1957), and cognitive research (Miller & Charles 1991). Recently, this hypothesis has been operationally defined in many ways in the fields of physicology, computational linguistics, and information retrieval (e.g. Li et al. (2000); Pado & Lapata (2007); Deerwester et al. (1990)).

In section 2.5 of the literature survey, we discussed two ways to extend distributional semantics for words to cover word sequences: (1) via the extension of the distributional hypothesis for specific word sequences (Lin & Pantel 2001); and (2) via the definition of compositional distributional semantics models (Mitchell & Lapata 2008; Jones & Mewhort 2007).

Lin & Pantel (2001) propose the *pattern distributional hypothesis* that extends the distributional hypothesis for specific patterns, i.e. word sequences representing partial verb phrases. The distributional meaning for these patterns is derived directly from looking at their occurrences in a corpus. Due to data sparsity, patterns of different length appear with very different frequencies in the corpus, affecting their statistics detrimentally. Subsection 2.5.1 discussed a variety of issues posed by the generalisation of the distributional hypothesis to cover sequences of words.

To overcome the data sparsity problem caused by the generalised distributional hypothesis, compositional distributional semantics models have been proposed (Mitchell & Lapata 2008; Jones & Mewhort 2007). Compositional distributional semantics (CDS) argue that the distributional meaning for sequences of any length can be obtained by composing the context distributions of the single words in the sequences. Context distributions are generally realised as vectors.

A general framework for compositional distributional semantics models proposed by Mitchell & Lapata (2008) was discussed in subsection 2.5.2 of the literature survey. In succession, subsections 2.5.2.1, 2.5.2.2 and 2.5.2.3 discussed in detail the CDS models proposed in Mitchell & Lapata (2008), Erk & Padó (2008), and Jones & Mewhort (2007), respectively: (a) a model building on basic vector operations, i.e. weighted addition and multiplication; (b) a model based on selectional preferences of the components of the sequence; and (c) a model that projects all vectors into the same dimensional space, called BEAGLE.

There are several problems mostly related to the evaluation of CDS models. In Mitchell & Lapata (2008), evaluation is based on a word sequence similarity test (Kintsch 2001): The proposed CDS models are employed to compose vectors for given verb-noun pairs, in which we know that the verb is ambiguous. However, the noun disambiguates the use of the verb in the pair. There are also two other verbs available for each pair, one of which matches the disambiguated meaning of the pair. Then, evaluation decides whether the composed vector is closer to the disambiguating verb than the other option.

For example, “run” means “gallop” if its subject is “horse”, while it means “dissolve” if its object is “colour”. Given the pair (“horse”,“run”), CDS models compose a vector for it and then the similarity between this and “gallop” is expected to be higher than its similarity to “dissolve”. The proposed CDS methods are reported to unsatisfactorily correlate with human annotated data mainly because the human inter-annotation agreement is low. As a result, it is not clear from the evaluation whether or not the resulting vectors for word sequences successfully represent their distributional semantics.

Moreover, despite the fact that Mitchell & Lapata (2008) propose a general CDS model, they only evaluate experimentally a couple of simplistic parametrisations. The proposed CDS models contain a large number of parameters, however there are no methods to estimate them. Most of the CDS models of the literature are not evaluated comparably, so it is not clear which model performs best for a potential application.

In this chapter, we propose a novel framework for investigating compositional distributional semantics (CDS). The proposed framework uses the notions of compositionality and non-compositionality that characterise multiword expressions to define a new task for evaluating CDS models. The basic idea consists of the following two parts:

1: Given a *compositional* multiword expression, a CDS model is expected to compose the context distributions of the multiword expression components into a distribution *identical or at least similar* to the context distributions of the multiword expression.

2: Given a *non-compositional* multiword expression, a CDS model is expected to compose the context distributions of the multiword expression components into a distribution *dissimilar* to the context distributions of the multiword expression.

We experiment with existing CDS models (Mitchell & Lapata 2008; Erk & Padó 2008; Jones & Mewhort 2007) with respect to this new evaluation framework.

Giesbrecht (2009) and Guevara (2010) have proposed similar evaluation tasks for assessing CDS models. Giesbrecht (2009) employed a variety of CDS models to estimate the meaning of 19 non-compositional German verb-noun combinations. Then, the author compared this composed vector with the vector of the expression as it is used on the whole. Giesbrecht (2009) conclude firstly that tensor products lead to better results than simple additive models as described in Mitchell & Lapata (2008), and secondly that further exploration of CDS models is needed. The framework presented in this chapter is different since CDS models are considered good not only if they perform well for

non-compositional instances but also for compositional ones.

Guevara (2010) presents a new CDS model employing Partial Least Squares Regression. The method is compared against existing CDS models on the task of composing meaning of compositional English adjective-noun sequences extracted from the British National Corpus (BNC). Since the proposed CDS model is a trainable one, 1000 sequences were used for training and 380 for testing. The authors report that the proposed method performs better than the simple additive and simple multiplicative models. A second evaluation setting is proposed; it compares against a gold standard created from the 10-nearest neighbours of each adjective-noun instance of the test set. In this setting the basic additive model is reported to perform better than the proposed trainable CDS model. The framework presented in this chapter is different since it also takes into account non-compositional instances.

In addition, we propose an estimation model that exploits compositional and non-compositional multiword expressions examples, to estimate parameters for additive compositional distributional semantics models. The model determines an equation system that represents a regression problem with multiple dependent variables. A solution to this equation system is estimated using the Moore-Penrose pseudo-inverse matrices (Penrose 1955). The estimation model and its solution were realised in close cooperation with the authors of Fallucchi & Zanzotto (2009). We experiment with this model to assess if more complicated parameter estimation can improve over simplistic parametrised additive models.

The results show that the multiword expressions used for experimentation do not appear frequently enough as sequences within the employed corpus; the British National Corpus BNC. To overcome this data sparsity problem, we propose a new method for extracting compositional distributional semantics examples and counter-examples. We used the same idea of compositionality but this time between single words and their definitions in dictionaries, instead of multiword expressions. Dictionary definitions are word sequences expressing the meaning of a word. Then, the word represents the compositional meaning of the definition; e.g. contact $\approx$ close interaction. Thus, dictionary definitions can make up the positive examples for evaluating CDS models.

In contrast, for creating negative, "non-compositional" examples, we use composed words that are etymologically derived from very old or ancient words, and over the years their meaning has deviated from the combination of the meaning of their constituents.

For example, the word *philosophy* derives from two Greek words *philos* (beloved) and *sophia* (wisdom) but it now means something different to the compositional meaning. Pairs of old composed words and their etymological constituents can be extracted from special etymological lexica such as *Wiktionary*.

Two different sets of positive and negative examples were collected. The first set consists of examples whose definitions or etymological constituents are adjective-noun or noun-noun sequences, while the second set of examples whose definitions or etymological constituents are verb-noun sequences. The results have shown that simplistic parametrisations of known CDS models do not prove to address this experimental setting successfully. In opposition, the parameter estimation method for the additive model performs significantly better and its accuracy increases as the number of singular values taken into account increases.

The rest of the chapter is organised as follows: In section 6.2 we describe our method for estimating parameters of CDS models. Section 6.3 describes in detail the construction of our two experimental datasets, one consisting of multiword expressions (subsection 6.3.1) and one consisting of single words (subsection 6.3.2). In section 6.4, we evaluate a large number of state-of-the-art CDS models on the proposed evaluation framework. Finally, section 6.5 summarises the chapter.

6.2 Estimating Additive CDS Models from Data

In this section, we propose a model that estimates the parameters of additive compositional distributional semantics (CDS) models shown in equation 2.92. This is a supervised method, since it uses training instances whose compositionality is previously known. As will be discussed in section 6.3, in the first set of experiments the model was trained on multiword expressions examples and in the second set on pairs of single words and their definitions. The estimation model and its solution were realised in close cooperation with the authors of Fallucchi & Zanzotto (2009).

The generic additive model composes a new vector $\vec{z}$ as a function of the component vectors $\vec{x}$ and $\vec{y}$. It is described in the following equation:

$$\odot(\mathsf{s}) = \vec{z} = A\vec{x} + B\vec{y} \tag{6.1}$$

A and B are not vectors but instead square matrices, allowing each dimension of the

	vector dimensions				
	between	gap	process	social	two
contact	$< 11,$	$0,$	$3,$	$0,$	$11 >$
x: *close*	$< 27,$	$3,$	$2,$	$5,$	$24 >$
y: *interaction*	$< 23,$	$0,$	$3,$	$8,$	$4 >$

Table 6.1: Example context vectors for the words of the definition: "*contact* $\equiv$ *close interaction*"

resulting vector $\vec{z}$ to depend upon all dimensions of $\vec{x}$ and $\vec{y}$. In this specialisation of equation 2.91, matrices A and B are used to capture the relation R and the background knowledge K.

Matrices A and B are very large to compute and store, posing restrictions that affect the applicability of the generic additive model. Estimating these matrices is neither a simple classification learning problem nor a simple regression problem. It is a regression problem with multiple dependent variables. Fallucchi & Zanzotto (2009) propose a method to estimate these matrices A and B using singular value decomposition, Moore-Penrose pseudo-inverse and a set of training examples. In this section this estimation method is briefly presented and in succession used in the experimental section 6.4.

Let s be a sequence of two words, x and y; i.e. s = x y. Suppose that $\vec{s}$, $\vec{x}$ and $\vec{y}$ are the distributional vectors corresponding to words s, x and y, respectively. Let e be the triple of vectors $(\vec{s}, \vec{x}, \vec{y})$, and E be a set of training examples each of which is a triple of vectors, e.

Note that an ideal perfectly performing compositional distributional semantics model $\odot$ is expected to output $\vec{s}$ as the composition of $\vec{x}$ and $\vec{y}$; i.e. $\vec{s} = \odot(\mathsf{xy})$. However, in general the expected vector $\vec{s}$ is not guaranteed to be equal to the composed one, $\vec{z} = \odot(\mathsf{xy})$. Table 6.1 shows an example triple $t = (\vec{contact}, \vec{close}, \vec{interaction})$, together with the corresponding distributional vectors in an example feature space, consisting of 5 context features: *between*, *gap*, *process*, *social*, and *two*. Feature values are occurrence counts within some corpus.

Using table 6.1 as a running example, subsection 6.2.1 formulates the problem as a system of linear equations. Subsection 6.2.2 presents the solution of Fallucchi & Zanzotto (2009).

6.2.1 Setting the linear equation system

This subsection describes how the regression problem with multiple dependent variables described above can be solved with a linear equation system. In the experimental section, we refer to our model as the estimated additive model (EAM).

The matrices A and B of equation 6.1 can be joined in a single matrix:

$$\vec{z} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} \tag{6.2}$$

For the example triple t of table 6.1, equation 6.2 is:

$$\vec{contact} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} \vec{close} \\ \vec{interaction} \end{pmatrix} \tag{6.3}$$

and it can be rewritten as:

$$\begin{pmatrix} 11 \\ 0 \\ 3 \\ 0 \\ 11 \end{pmatrix} = \begin{pmatrix} A_{5\times5} & B_{5\times5} \end{pmatrix} \begin{pmatrix} 27 \\ 3 \\ 2 \\ 5 \\ 24 \\ 23 \\ 0 \\ 3 \\ 8 \\ 4 \end{pmatrix} \tag{6.4}$$

The matrices in equation 6.2, can be transposed as follows:

$$\begin{aligned} \vec{z}^T &= \left(\begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} \right)^T \\ &= \begin{pmatrix} \vec{x}^T & \vec{y}^T \end{pmatrix} \begin{pmatrix} A^T \\ B^T \end{pmatrix} \end{aligned} \tag{6.5}$$

In the current setting, matrix $\begin{pmatrix} \vec{x}^T & \vec{y}^T \end{pmatrix}$ is a known quantity and matrix $\begin{pmatrix} A^T \\ B^T \end{pmatrix}$ is to be estimated. For our running example t, this equation is:

$$\begin{pmatrix} 11 & 0 & 3 & 0 & 11 \end{pmatrix} = \begin{pmatrix} 27 & 3 & 2 & 5 & 24 & 23 & 0 & 3 & 8 & 4 \end{pmatrix} \begin{pmatrix} A^T_{5\times 5} \\ B^T_{5\times 5} \end{pmatrix} \tag{6.6}$$

Equation 6.5 serves as prototype for the final equation system. The larger the matrix $\begin{pmatrix} AB \end{pmatrix}$ to be estimated, the more equations like 6.5 are needed. Given that the training examples set E contains n triples $(\vec{s}, \vec{x}, \vec{y})$, we can write the following system of equations:

$$\begin{pmatrix} \vec{z}_1^T \\ \vec{z}_2^T \\ \vdots \\ \vec{z}_n^T \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} \vec{x}_1^T & \vec{y}_1^T \end{pmatrix} \\ \begin{pmatrix} \vec{x}_2^T & \vec{y}_2^T \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \vec{x}_n^T & \vec{y}_n^T \end{pmatrix} \end{pmatrix} \begin{pmatrix} A^T \\ B^T \end{pmatrix} \tag{6.7}$$

The vectors derived from the triples can be seen as two matrices of n rows, Z and $\begin{pmatrix} XY \end{pmatrix}$ related to $\vec{z}_i^T$ and $\begin{pmatrix} \vec{x}_i^T & \vec{y}_i^T \end{pmatrix}$, respectively. The overall equation system is then the following:

$$Z = \begin{pmatrix} X & Y \end{pmatrix} \begin{pmatrix} A^T \\ B^T \end{pmatrix} \tag{6.8}$$

This equation system represents the constraints that matrices A and B have to satisfy in order to be a possible linear compositional distributional semantics model that can at least describe seen examples. We will hereafter call $\Lambda = \begin{pmatrix} A & B \end{pmatrix}$ and $Q = \begin{pmatrix} X & Y \end{pmatrix}$. The system in equation 6.8 can be simplified as:

$$Z = Q\Lambda^T \tag{6.9}$$

As Q is a rectangular and singular matrix, it is not invertible and the system in equation 6.8 has no solutions. It is possible to use the principle of Least Square Estimation for computing an approximation solution. The idea is to compute the solution $\widehat{\Lambda}$ that

minimises the residual norm, i.e.:

$$\widehat{\Lambda}^T = \arg\left[\min_{\Lambda^T} \|Q\Lambda^T - Z\|^2\right] \tag{6.10}$$

One solution for this problem is the *Moore-Penrose pseudo-inverse* (Penrose 1955). In linear algebra, the pseudo-inverse A^+ of a matrix A is a generalisation of the inverse matrix and is primarily applied to compute a least squares solution estimate of systems that lack a unique solution. The pseudo-inverse can be defined and is unique for all matrices whose items are real or complex numbers. It can be computed using singular value decomposition. The pseudo-inverse A^+ of an $m \times n$ matrix A is defined as the unique $m \times n$ matrix satisfying all of the following four criteria:

1: $A\ A^+$ maps all column vectors of A to themselves; i.e. $AA^+A = A$
2: $A+$ is a weak inverse for the multiplicative semi-group; i.e. $A^+AA^+ = A^+$
3: AA^+ is Hermitian; i.e. $(AA^+)^* = AA^+$; and
4: A^+A is also Hermitian; i.e. $(A^+A)^* = A^+A$

In the above descriptions, A^* is the Hermitian transpose of a matrix A. For matrices M whose elements are real numbers the Hermitian transpose equals the matrix transpose, $M^* = M^T$.

Using the Moore-Penrose pseudo-inverse, Q^+, as a solution for equation 6.10 results in the following final equation:

$$\widehat{\Lambda}^T = Q^+Z \tag{6.11}$$

The pseudo-inverse matrix, Q^+, can provide an approximated solution even if equation 6.10 system has no solutions.

6.2.2 Computing Moore-Penrose pseudo-inverse

In this subsection we discuss how the Moore-Penrose pseudo-inverse, Q^+ in equation 6.11, can be computed using singular value decomposition (SVD), following (Fallucchi & Zanzotto 2009). Singular value decomposition (SVD) is widely used in computational linguistics and information retrieval for reducing dimensions of feature spaces, or in other words reduce the number of features (Deerwester et al. 1990).

Moore-Penrose pseudo-inverse (Penrose 1955) is computed as follows: Let the original (input) matrix Q of dimensions $n \times m$ be of rank r. The singular value decomposition of the original matrix Q is:

$$Q = U\Sigma V^T \tag{6.12}$$

Σ is a square diagonal matrix of dimension r. Then, the pseudo-inverse matrix that minimises equation 6.10 is:

$$Q^+ = V\Sigma^+ U^T \tag{6.13}$$

The square diagonal matrix Σ^+ of dimension r is the transposed matrix of Σ having as diagonal elements the reciprocals of the singular values of Σ, i.e. $\frac{1}{\delta_1}, \frac{1}{\delta_2}, ..., \frac{1}{\delta_r}$. Singular values are defined as the absolute values of the eigenvalues.

Using SVD to compute the pseudo-inverse matrix allows for different approximations (Fallucchi & Zanzotto 2009). The algorithm for computing the singular value decomposition is iterative and able to stop at a given k less than the real rank r (Golub & Kahan 1965). The property of singular values, i.e., $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$, guarantees that initially derived dimensions have higher singular values, i.e. the first k are bigger than the discarded $r - k$. High singular values correspond to dimensions of the new space where examples have more variability whereas low singular values determine dimensions where examples have smaller variability (Liu 2007). Thus, higher dimensions k are more informative than low dimension $k' > k$, and discarding a number of lowest dimensions can potentially increase discrimination ability. We can consider different values for k to obtain different SVD for the approximations Q_k^+ of the original matrix Q^+ in equation 6.13), i.e.:

$$Q_k^+ = V_{n\times k}\Sigma^+_{k\times k}U^T_{k\times m} \tag{6.14}$$

where Q_k^+ is a matrix n by m obtained considering the first k singular values.

6.3 Building positive and negative examples

As explained in the previous section, a set of training data is essential to estimate the parameters of additive CDS models. Similarly, a set of test instances is needed to

evaluate all models that were presented in this section. Obviously, the choice of these data is very crucial, since the data sets the task on which the models will be evaluated, i.e. defines what distinguishes a well performing CDS model from an unsuccessful one. In the introduction of the present chapter, section 6.1, and in the summarising discussion of the literature survey, section 2.5.3, we identified that the tasks, on which CDS models were evaluated, do not clarify how well the models perform. In this section we propose a different paradigm for evaluating CDS models.

Training or testing instances can be presented in the format of triples. Supposing that s is a sequence of two words or a single word; and x, y single words, then a triple e consists of the vectors that represent their context distributions in some corpus, i.e. $e \equiv (\vec{s}, \vec{x}, \vec{y})$ Data triples can either encode multiword expression instances or single word instances. In the former case, s is a sequence of two words, the multiword expression itself. x and y are its first and second component word, respectively. In the case that a data triples encodes a single word, its first part, s, encodes the single word itself while x and y encode a sequence of two words related to the single word s. The example of table 6.1 corresponds to triple, (*contact*, *close*, *interaction*) The definition of the general distributional hypothesis, described in section 2.5.1, allows comparing word sequences of different length.

Both the multiword expression dataset and the single word dataset contain positive and negative instances. For positive instances, $(\vec{s}_{pos}, \vec{x}_{pos}, \vec{y}_{pos})$, a good CDS model is expected to compose the second and third part, i.e. $\vec{x}_{pos}$ and $\vec{y}_{pos}$, into a new vector $z = \odot(\mathsf{s})$ similar to $\vec{s}_{pos}$. An ideal, perfectly performing CDS model would compose a vector, $\vec{z}$, identical to $\vec{s}_{pos}$, i.e. $\vec{z} \equiv \vec{s}_{pos}$. In contrast, for negative instances, $(\vec{s}_{neg}, \vec{x}_{neg}, \vec{y}_{neg})$, the composition $z = \odot(\mathsf{s})$ of the second and third part of the triple, i.e. $\vec{x}_{neg}$ and $\vec{y}_{neg}$, is expected to be significantly different that $\vec{s}_{neg}$. An ideal, perfectly performing CDS model would compose a vector, $\vec{z}$, totally dissimilar to $\vec{s}_{neg}$, i.e. $\vec{z} \neq \vec{s}_{pos}$. The positive and negative instances can be employed to determine whether a CDS model is good or not and also to compare different CDS models.

The dataset of Mitchell & Lapata (2008) consists of instances $(\mathsf{v}, \mathsf{n}_1, \mathsf{v}_1, \mathsf{n}_2, \mathsf{v}_2)$. v is an ambiguous verb. When combined with n_1, v is disambiguated in one of its senses, similar to v_1. When v is combined with n_2, it is disambiguated in a different sense, similar to v_2. For example, *(ran*, *horse*, *gallop*, *colour*, *dissolve)*. The verb *ran* means *gallop* if its subject is *horse*, while it means *dissolve* if its subject is *colour*.

This dataset poses a different task, closer to disambiguation than to distributional semantic composition. Moreover, it contains 60 tuples, very few to split in testing and training part for our experiments. To the best of our knowledge, there are no other suitable sets available.

We induced two datasets, one containing multiword expressions and one single words. The latter was created after experiments on the former finished. The results showed that the multiword expressions of the dataset were very rare in the corpus that we employed to extract context distributions; the British National Corpus (BNC). The set containing single words was developed so as to address this data sparsity problem.

6.3.1 Dataset containing multiword expressions

In the previous section, we discussed a novel evaluation framework to assess and compare compositional distributional semantics (CDS) models. In this subsection, we will describe how the general dataset definition of the previous section can be specified to contain multiword expressions that consist of two words. As explained, each data triple, $(\vec{s}, \vec{x}, \vec{y})$, encodes the context distributional vector of a multiword expression, `s`, and the the context distributional vector of the multiword expression components, `x` and `y`.

The multiword expressions dataset needs to be divided into two parts: one containing positive instances and one containing negative ones. The notion of compositionality suits perfectly to be used as a distinguishing factor.

Positive instances of the dataset are the instances that encode compositional multiword expressions. According to the definition of compositionality, the meaning of compositional multiword expressions can be computed as a combination of the meanings of its component words. This is exactly what it is expected from the positive instances of the dataset of the proposed evaluation framework. For example, the compositional multiword expression *tea table* is encoded to the following triple within the positive part of the dataset: $(\vec{\text{tea\_table}}, \vec{\text{tea}}, \vec{\text{table}})$.

As far as negative instances are concerned, the restriction forced by the proposed framework is that the context distribution representing the meaning of the multiword expression should be significantly different to combinations of the context distributions of the component words. It would be possible to create negative instances by replacing the component words of positive examples with other random words, i.e to take a positive triple, $(\vec{s}_{pos}, \vec{x}_{pos}, \vec{y}_{pos})$, created from a compositional multiword expression and

replace $\vec{x}_{pos}$ and $\vec{y}_{pos}$ with random words $\vec{x}_{random}$ and $\vec{y}_{random}$. Due to data sparseness, the probability that the combination of the meaning of random words, $\vec{x}_{random}$ and $\vec{y}_{random}$, is totally unrelated to the meaning of the multiword expression, $\vec{s}_{pos}$, is very high. However, this method to construct negative instances will result in instances that are too generic to be interesting cases. In particular, the component words, $\vec{x}_{random}$ and $\vec{y}_{random}$, are expected to be too loosely related to the multiword expression, $\vec{s}_{pos}$.

Instead we prefer to encode non-compositional multiword expressions as the negative instances of the dataset. This way of constructing negative instances is directly comparable to the way that positive instances were constructed. For example, the non-compositional multiword expression *fish finger* can be encoded to the following triple within the negative part of the dataset: $(\vec{\text{fish\_finger}}, \vec{\text{fish}}, \vec{\text{finger}})$.

6.3.2 Dataset containing single words

As will be discussed in detail in the experiments section, experiments on the multiword expressions dataset showed that neither standard CDS models nor the trainable CDS model can perform significantly better than the random baseline. Inspecting the dataset showed the most possible reason for this failure. Multiword expressions occur rarely in the chosen corpus, so that no reliable context distributions can be extracted. The lengthier a sequence of words is, the rarer it becomes and the lower the reliability of its distributional vector is. A dataset of single words instead of multiword expressions seems ideal to address the sparsity problem.

Positive instances, $(\vec{s}_{pos}, \vec{x}_{pos}, \vec{y}_{pos})$, should consist of the distributional vectors of a single word, `s`, and two other words, `x` and `y`, whose meaning in combination should be similar to the meaning of `s`. Such equivalences can be found in dictionaries, natural repositories of equivalent expressions. Words that are defined in dictionaries are declared to be semantically similar to their definition sequences. This holds for at least some sense of the defined words. For example, consider the dictionary definition *contact* $\equiv$ *close interaction*. Since a single word, *contact*, is declared to be semantically similar to a two-word expression, *close interaction*, it can be used as a positive instance: $(\vec{\text{contact}}, \vec{\text{close}}, \vec{\text{interaction}})$ Such dictionary definitions can be extracted from any dictionary, e.g. *WordNet*.

Negative instances, $(\vec{s}_{neg}, \vec{x}_{neg}, \vec{y}_{neg})$, should consist of the distributional vectors of a single word, `s`, and two other words , `x` and `y`, whose combination of meanings is

dissimilar to the meaning of s. As discussed for the negative instances of the multiword expressions dataset, single word negative instances could be created in a random way. However we noticed that randomly created negative instances can be too generic and too loosely related to be interesting cases.

Instead, we use the following idea: Many composed words are etymologically derived from very old or ancient words. These words consist of components which are in general not related to their meaning. For example, the word *philosophy* derives from two Greek words *philos* and *sophia*, which mean *beloved* and *wisdom*, respectively. However, the use of the word *philosophy* in not related to the uses of *beloved* and *wisdom*. The word has lost its original compositional meaning. Other examples are *municipal* whose Latin components translate into *receive duty*, and *octopus* whose Greek components translate into *eight foot*. As the above examples suggest, composed words consisting of old or ancient Latin or Greek components are mostly non-compositional and can be used to create negative instances for the single words dataset. The above examples can be encoded into the following negative instances: $(\text{philo}\vec{\text{sophy}}, \text{bel}\vec{\text{oved}}, \text{wis}\vec{\text{dom}})$, $(\text{muni}\vec{\text{cipal}}, \text{rec}\vec{\text{eive}}, \text{d}\vec{\text{uty}})$, and $(\text{octo}\vec{\text{pus}}, \text{ei}\vec{\text{ght}}, \text{f}\vec{\text{oot}})$. Negative instances can be extracted from dictionaries containing etymological information such as *Wiktionary*[1].

The proposed way of extracting positive and negative instances has the following desirable properties:

Property 1: Since the dataset contains definitions and etymological relations of single words, we can extract stable and meaningful distributional vectors for them. Then, these vectors can be compared to the distributional vectors obtained using the CDS model under evaluation. This new dataset successfully tackles the sparsity problem of the dataset containing multiword expressions.

Property 2: The second and third parts of data instances, x and y following the previous notation, exhibit a wide variety of different syntactic structures, e.g. adjective-noun, noun-noun and verb-noun. This allows training and testing CDS models that take into account syntax, such as the model of Erk and Pado discussed in section 2.5.2.2. Table 6.3.2 presents the distribution of the most frequent syntactic structures in the definitions of *WordNet* 3.0 (Miller 1995). Definitions were parsed using the Charniak parser (Charniak 2000).

[1] http://www.wiktionary.org

Frequency	*Structure*
2635	(FRAG (PP (IN) (NP (DT) (JJ) (NN))))
833	(NP (DT) (JJ) (NN))
811	(NP (NNS))
645	(NP (NNP))
623	(S (VP (VB) (ADVP (RB))))
610	(NP (JJ) (NN))
595	(NP (NP (DT) (NN)) (PP (IN) (NP (NN))))
478	(NP (NP (DT) (NN)) (PP (IN) (NP (NNP))))
451	(FRAG (PP (IN) (NP (NN))))
419	(FRAG (RB) (ADJP (JJ)))
375	(S (VP (VB) (PP (IN) (NP (DT) (NN)))))
363	(S (VP (VB) (PP (IN) (NP (NN)))))
342	(NP (NP (DT) (NN)) (PP (IN) (NP (DT) (NN))))
341	(NP (DT) (JJ) (JJ) (NN))
330	(ADJP (RB) (JJ))
307	(NP (JJ) (NNS))
244	(NP (DT) (NN) (NN))
241	(S (NP (NN)) (NP (NP (NNS)) (PP (IN) (NP (DT) (NNP)))))
239	(NP (NP (DT) (JJ) (NN)) (PP (IN) (NP (DT) (NN))))

Table 6.2: Top 20 syntactic structures of *WordNet* definitions

6.4 Experiments

This section presents results of experimentation with the proposed novel framework to evaluate compositional distributional semantics (CDS) models. The aim is firstly to determine if existing CDS models can discriminate between the positive and negative instances of a given dataset, as described in section 6.3. CDS models can be compared with respect to their ability of detecting a statistically significant difference between positive and negative instances of a dataset.

Specifically, we evaluate the following existing CDS models:

1: basic additive model (BAM)

2: basic multiplicative model (BMM)

3: circular convolution multiplicative model (CCMM)

4: basic additive model with environmental vectors (BAM-E)

5: basic multiplicative model with environmental vectors (BMM-E)

6: circular convolution multiplicative model with environmental vectors (CCMM-E)

7: basic additive model with selectional preferences (BAM-SP)

8: basic multiplicative model with selectional preferences (BMM-SP)

9: circular convolution multiplicative model with selectional preferences (CCMM-SP)

10: basic additive model environmental vectors and selectional preferences (BAM-E-SP)

11: basic multiplicative model with environmental vectors and selectional preferences (BMM-E-SP)

12: circular convolution multiplicative model with environmental vectors and selectional preferences (CCMM-E-SP)

Most of these models were discussed in subsection 2.5.2 of the literature survey. The remaining models are combinations of those of subsection 2.5.2 and are explained below:

The circular convolution multiplicative model (CCMM) is inspired from BEAGLE (Jones & Mewhort 2007), which was described in section 2.5.2.3. It uses circular convolution to compose vectors $\vec{x}$ and $\vec{y}$:

$$\begin{aligned} z_i &= \sum_{j=1}^{n} x_j y_{\lceil i-j \rceil} \\ \lceil i-j \rceil &= \begin{cases} i-j, & \text{if } i > j \\ n+i-j, & \text{if } i \leq j \end{cases} \end{aligned} \tag{6.15}$$

The circular convolution multiplicative model with environmental vectors (CCMM-E) is the original BEAGLE model. The basic additive model and the basic multiplicative model with environmental vectors (BAM-E and BMM-E) compose environmental vectors by addition and multiplication, respectively. BAM-E can be expressed as follows:

$$\odot(\mathsf{s}) = \vec{z} = \alpha\vec{\overline{x}} + \beta\vec{\overline{y}} \tag{6.16}$$

where α and β are two scalar parameters. BMM-E can be represented as:

$$\odot(\mathsf{s}) = \vec{z} = \vec{\overline{x}}^{T}\vec{\overline{y}} \tag{6.17}$$

Vectors $\vec{\overline{x}}$ and $\vec{\overline{y}}$ are assumed to be row vectors. The dimension of enviromental vectors,

D, was set to 1024 for the experiments of this chapter.

The basic additive, multiplicative and circular convolution multiplicative models with selectional preferences (BAM-SP, BMM-SP and CCMM-SP) are different implementations of selectional preferences, described in equations 2.106 and 2.107, following addition, multiplication and circular convolution, respectively.

Models combining environmental vectors and selectional preferences are a further combination of the environmental vectors of BEAGLE and the model of Erk and Pado. These models work exactly as the selectional preferences models of equations 2.106 and 2.107 but the all vectors are beforehand mapped in the environmental feature space.

Recall equation 2.93 which describes the basic additive model (BAM):

$$\odot\,(\mathsf{s}) = \vec{z} = \alpha\vec{x} + \beta\vec{y}$$

All the above models based on BAM, i.e. BAM, BAM-E, BAM-SP and BAM-E-SP, are evaluated for unary values of the scalar parameters: $\alpha = \beta = 1$; and also for increments of 0.1 of scalar α and at the same time $\beta = 1 - \alpha$.

The second target of this experiments is to investigate whether and to what extent estimating the parameters of additive CDS models from data is helpful towards this task, i.e. whether the additive CDS model with estimated parameters (EAM) performs better than existing CDS models. In particular, EAM, which employs the parameter estimation that was presented in section 6.2, is directly comparable only to BAM, since EAM estimates the parameters of it.

The whole datasets of multiword expressions or single words were used for testing purposes for comparison among existing CDS models, since none of this models needs any kind of training. In contrast, to compare between BAM and EAM, the datasets were divided into a training and a test part, 50% each.

In subsection 6.4.1 we present the setting of conducted experiments. We discuss the adopted distributional similarity measure and evaluation measure and we provide construction details of the multiword expressions dataset and the single words dataset. In subsections 6.4.2 and 6.4.3 we present and discuss the evaluation results on the multiword expressions dataset and the single words dataset, respectively.

6.4.1 Experimental setting

In this section, we describe the chosen measure of distributional similarity, a measure to assess whether a given CDS model can differentiate between positive and negative instances, and finally details about dividing the multiword expressions and single words dataset in a training and a testing part. Cosine similarity was employed as a measure to compute distributional similarity. Let s be a word or a sequence of words and x, y two words which consist a sequence related to s. For positive instances, s is compositional and the context distribution of xy is expected to be identical to the context distribution of s. For negative instances, s is non-compositional and the context distribution of xy although related is expected to be significantly different from the context distribution of s. Suppose that $\vec{s}$, $\vec{x}$ and $\vec{y}$ are the distributional vectors corresponding to words s, x and y, respectively. Cosine similarity compares the context vector $\vec{s}$ to the composed vector $\vec{z} = \odot(\mathsf{xy})$. As discussed in subsection 2.3.3, the cosine similarity $cos(\vec{v}, \vec{u})$ of two n-dimensional vectors $\vec{v}$ and $\vec{u}$ is defined as:

$$cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \, \|\vec{v}\|}$$

$\cdot$ is the dot product and $\|\vec{a}\|$ is the magnitude of vector $\vec{a}$ computed using the Euclidean norm.

Cosine was the best performing distributional similarity measure for this task among a large variety of measures that have been tested: Hamming distance, Euclidean distance, Jaccard coefficient, Dice coefficient and cosine similarity. Binary and weighted versions of these measures were implemented. The former assume unary values for all non-zero vector dimensions, while the latter take the actual vector values into account.

We have specified how to compute the semantic similarity of the composed vector to the target word vector. In succession, we need to assess if the similarity values of positive instances are significantly different from the similarity values of negative instances. For a good CDS model, the distribution of similarities $sim(\vec{s}, \odot(\mathsf{xy}))$ over all positive instances should be significantly different from the same distribution of similarities over all negative instances.

For this purpose we used Student's t-test for two independent samples of different sizes. Student's t-test hypothesises that the two samples are normally distributed. The null hypothesis states that the means of the two samples are equal, $\mu_1 = \mu_2$. Student's

t-test takes into account the sizes N_1 and N_2, means M_1 and M_2, and variances s_1^2 and s_1^2 of the two samples to compute the following value:

$$\begin{aligned} t &= \frac{M_1 - M_2}{\sqrt{\frac{2(s_1^2 + s_2^2)}{df \times N_h}}} \\ \text{where: } df &= N_1 + N_2 - 2 \\ \text{and: } N_h &= 2 \times \frac{N_1 \times N_2}{N_1 + N_2} \end{aligned} \tag{6.18}$$

df stands for the degrees of freedom and N_h is the harmonic mean of the sample sizes.

Given the statistic t and the degrees of freedom df, we can compute the probability that the two samples derive from the same distribution. This probability value is referred to as p-value and is formally defined as the probability of obtaining a t statistic value at least as extreme as the one that was actually observed, assuming that the null hypothesis holds. The null hypothesis can be rejected if p-value is below the chosen threshold for statistical significance (usually 0.1, 0.05 or 0.01), otherwise it is accepted. In our case, rejecting the null hypothesis means that the similarity values of positive instances are significantly different from the similarity values of negative instances and the corresponding CDS model performs well. Accepting the hypothesis denotes insignificant differences and the corresponding CDS model is unable to differentiate between positive and negative instances. As a result, p-value can be used as a performance ranking function for CDS models.

The dataset consisting of multiword expressions was described in subsection 6.3.1. The procedure described in section 4.3 is able to derive evidence about the compositionality of a given multiword expression that occurs in *WordNet* (Miller 1995). *WordNet 3.0* contains $52,217$ multiword expressions of which $6,287$ multiword expression were judged as non-compositional. Multiword expressions that contain prepositions or consist of more than two components were filtered out. Only adjective-noun or noun-noun multiword expression were kept. This filtering procedure resulted in 5000 non-compositional multiword expressions which consist the negative instances of the multiword expression dataset. The positive instances are 5000 multiword expressions randomly chosen from the pool of multiword expressions that were judged as compositional by the procedure in subsection 4.3.

The dataset consisting of single words was introduced in subsection 6.3.2. Posit-

ive instances were extracted from *WordNet* and negative instances from *Wiktionary*. We chose to collect two different categories of single target word instances: (a) instances containing adjective-noun or noun-noun sequences (*NN* single word set); and (b) instances containing verb-noun sequences (*VN* single word set).

This distinction is expected to aid in exploiting models with selectional preferences, i.e. BAM-SP, BMM-SP and CCMM-SP. Capturing different syntactic relations, the two sets can support that our results are independent from the syntactic relation between the words of each sequence. While implementing these models we used as semantic heads the second word of each sequence of *NN* single word instances, and the first word of *VN* single word instances. The corresponding models of implementing selectional preferences were described in equations 2.106 and 2.107, respectively. For *NN* single word instances, we considered the syntactic relation adjectival_modifier. In contrast, for *VN* single word instances, we manually tagged each sequence with either relation noun_subject or noun_object depending on the semantics of the sequence.

The *NN* single word set contains 1065 positive and 377 negative instances, while the *VN* single word set contains 161 positive and 111 negative instances. As already mentioned, positive instances were extracted for *WordNet* and negative ones from *Wiktionary*. The size of the *VN* single word set is small due to the fact that verb-noun sequences are quite rare in *WordNet* and *Wiktionary*. Frequency vectors for all multiword expressions and single words occurring in both the multiword expressions dataset and the single words datasets were constructed from the British National Corpus using sentences as contextual windows and words as features.

6.4.2 Results on the dataset of multiword expressions

In this subsection, we present the results of all experiments on the multiword expressions dataset as described previously. The first experiment compares existing models for Compositional Distributional Semantics (CDS) on the whole dataset. The second experiment compares the basic additive model (BAM) with the estimated additive model (EAM) that was proposed in section 6.2. Since EAM needs training data for the estimation of matrices, positive and negative instances were split into two halves to consist the training and testing parts. In both experiments, CDS models are judged according to their ability to distinguish between positive and negative instances of the dataset. This ability is measured by the probability of confusing positive and negative instances; i.e. Stu-

CDS model	*MWE dataset*
BAM (α=β=1)	0.92915
BMM	0.54831
CCMM	0.76071
BAM-SP (α=β=1)	0.37726
BMM-SP	0.27555
CCMM-SP	0.07760

CDS model	*MWE dataset*
BAM-E (α=β=1)	0.26178
BMM-E	0.28529
CCMM-E	0.62764
BAM-E-SP (α=β=1)	0.62304
BMM-E-SP	0.44186
CCMM-E-SP	0.90381

Table 6.3: Probability of confusing positive and negative instances of the multiword expressions dataset when composing with existing CDS models

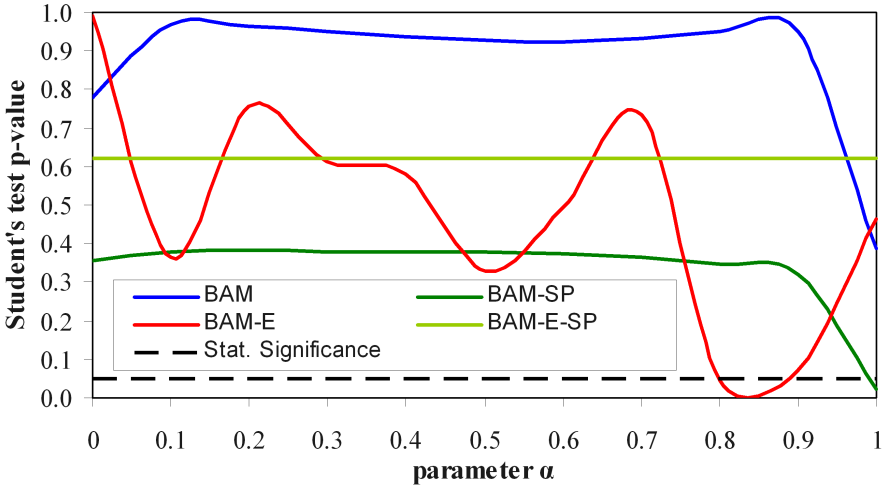

Figure 6.1: Probability of confusing positive and negative instances of the multiword expressions dataset when composing with existing CDS models

dent's test p-value. As discussed in subsection 6.4.1, the level of statistical significance is usually set to 0.1%, 0.05% or 0.01%. We adopt 0.05% for this discussion.

Table 6.3 presents the results of the first experiment. We observe that no method successfully distinguishes between positive and negative instances. The best performing model is the circular convolution multiplicative model with selectional preferences (CCMM-SP), however its probability of confusing positive and negative instances is slightly larger than the statistical significance threshold 0.05%.

In table 6.3 BAM-based models are evaluated for unary values of parameters α and

CDS model	*MWE dataset*
BAM (α=β=1)	0.12891
EAM (k=1)	0.61469
EAM (k=10)	0.58602
EAM (k=20)	0.66433

Table 6.4: Probability of confusing positive and negative instances of the multiword expressions dataset when composing with BAM and EAM

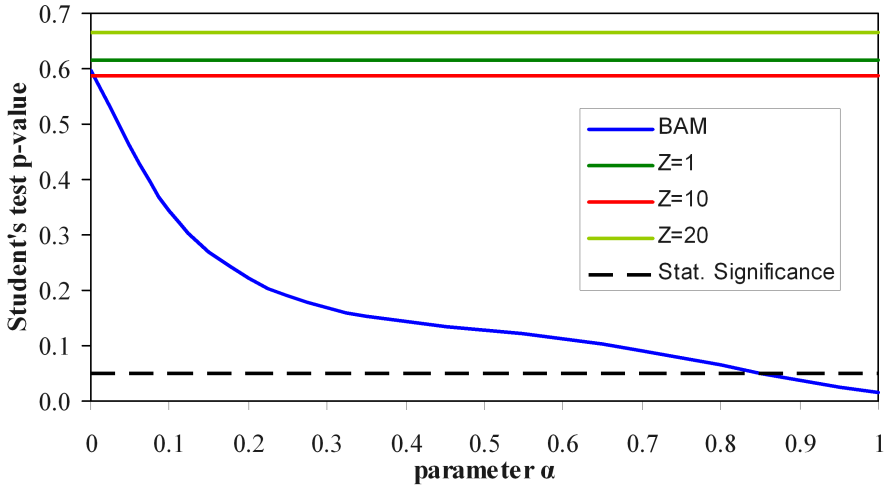

Figure 6.2: Probability of confusing positive and negative instances of the multiword expressions dataset for BAM and EAM and for various values for parameter α (where $\beta = 1 - \alpha$)

β, only. Figure 6.1 investigates further the parametrisation of BAM-based models. The y-axis represents Student's test p-value for all values of α in 0.1 increments; at the same time $\beta = 1 - \alpha$. The basic observation remains the same; no CDS model can distinguish between positive and negative instances. The basic additive CDS model with environmental vectors (BAM-E) appears to be the most dependent upon the values of parameters α and β.

BAM-based models depend on the two parameters α and β (equations 2.93 and 6.16). Table 6.4 presents the results of the second experiment. It shows the p-values of BAM and three versions of EAM, each of which takes different number of eigenvalues (k) into account to compute the Moore-Penrose pseudo-inverse, as explained in sub-

section 6.2.2. No CDS model successfully distinguishes between positive and negative examples.

The picture is the same in figure 6.2 where the same EAM models are compared against a different parametrisation for BAM. Again, the p-values achieved by BAM for all values of α in 0.1 increments and $\beta = 1 - \alpha$ are plotted. The representation focuses on the performance of BAM with respect to different α values. The performance of EAM for different k values is represented with horizontal lines. Probabilities of different models are directly comparable. The dashed line represents the threshold of statistical significance; the value below which the detected difference between the positive and negative instances becomes statistically significant.

BAM performs well only when $\alpha = 0.9$ or $\alpha = 1$, i.e. when the CDS composed vector is equal to the context vector of the first component of each multiword expression. This result motivated us to inspect the data to explain why. The majority of multiword expressions do not occur in the BNC or occur once. Usually, the first component word is also very infrequent in the BNC, so its context is very similar to the context of the multiword expression, and this is captured by BAM. As a result, experiments on the dataset of multiword expressions indicate that no CDS model can successfully accomplish the task. This is basically because the multiword expressions occur rarely in the chosen corpus, and thus the dataset should be changed to consist of more frequent word sequences.

6.4.3 Results on the dataset of single words

The dataset of single words was induced to reduce the data sparsity that the multiword expressions dataset suffered. As discussed in subsection 6.4.1, the single words dataset was divided into two sets: an *NN* set, whose sequences consist of two nouns, and *VN* set, whose sequences consist of a verb and a noun. Table 6.5 reports the results of evaluating existing CDS models on the *NN* and *VN* single words dataset. CDS models are scored by the probability p that positive and negative instances of the dataset are similarly distributed. In this first experiment the whole dataset is used for testing.

In general, we observe that it is rather difficult to find a good CDS model in both the *NN* and *VN* single words dataset. There are 4 CDS models that distinguish well between positive and negative instances of the *NN* dataset, but not for the *VN* dataset. For example, BAM seems to be a good candidate as the null hypothesis can be rejected for *NN* experiments, but it fails in *VN* experiments. There is only one method that

	Single words datasets	
CDS model	*NN*	*VN*
BAM (α=β=1)	**0.02585**	0.80129
BMM	0.10808	0.95079
CCMM	**0.01428**	0.71929
BAM-SP (α=β=1)	0.13857	0.19061
BMM-SP	**<1.00E-10**	**0.00050**
CCMM-SP	**<1.00E-10**	0.18177
BAM-E (α=β=1)	0.94686	0.37125
BMM-E	0.30468	0.31580
CCMM-E	0.40191	0.27970
BAM-E-SP (α=β=1)	0.63673	0.90745
BMM-E-SP	0.80022	0.41503
CCMM-E-SP	0.17627	0.25803

Table 6.5: Probability of confusing positive and negative instances of single words datasets *NN* and *VN* when composing with existing CDS models

performs well on both datasets: BMM-SP, and thus seems to be a good candidate for a general CDS model. In contrast, CDS models for which the null hypothesis cannot be rejected cannot be considered as good models.

In succession, we attempt to explore whether the parameters of BAM-based CDS models can play an important role. Figure 6.3 reports Student's t-test p-values with respect to α for 0.1 increments. To reduce two parameters to one β was set to $1 - \alpha$.

A first observation for the *NN* dataset is that BAM-based models with selectional preferences can discriminate between positive and negative instances when $\alpha = 0$. This is not evidence that the CDS models perform well because when $\alpha = 0$ there is no real composition happening; we are considering only the second component word of the sequence. Whenever the vectors of the two words are really composed ($alpha < 1$ and $beta < 1$), the values of the probability are far from being satisfactory. Yet, especially for models employing environmental vectors the p-values change dramatically for different values of parameter α, for both datasets.

For the *NN* dataset, BAM seems to be able to successfully distinguish between positive and negative instances for $\alpha > 0$. In contrast, for the *VN* dataset, all CDS model tend to confuse positive and negative instances much more than tolerable. Even if these

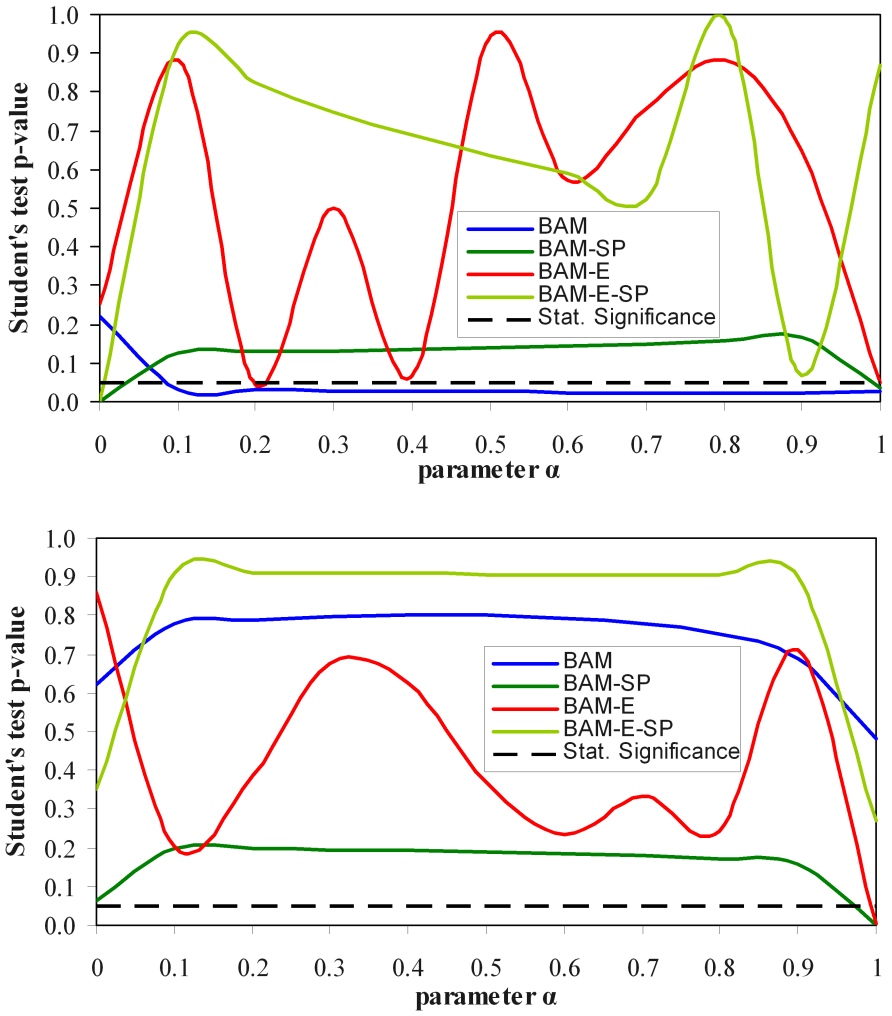

Figure 6.3: Probability of confusing positive and negative instances of *NN* (up) and *VN* (down) single words datasets when composing with existing CDS models and for various values for parameter α (where $\beta = 1 - \alpha$)

plots do not give the possibility to select the correct α, they suggest that although the additive CDS models can be useful, there is also need to scale to more complex additive models.

The second experiment compares BAM with three versions of EAM, each one considering different number of eigenvalues for computing the Moore-Penrose pseudo-inverse

	Single words datasets	
CDS model	*NN*	*VN*
BAM (α=β=1)	0.05690	0.50753
EAM (k=1)	0.11735	0.22306
EAM (k=10)	$<$**1.00E-10**	0.16452
EAM (k=20)	**0.00431**	**0.00453**

Table 6.6: Probability of confusing positive and negative instances of single word datasets *NN* and *VN* for BAM and EAM.

matrix. Results are shown in table 6.6. We observe that in contrast to BAM, some EAM models succeed in separating positive from negative examples for both sets. For the *NN* single words dataset this happens for $k = 10$ and $k = 20$, while for the *VN* single words dataset for $k = 20$, only.

Figure 6.4 shows the results of investigating whether simple parameter adjustment of BAM can perform better than EAM. Plots show the basic additive model (BAM) with different values for parameter α, where $\beta = 1 - \alpha$, and EAM computed for different approximations of the Moore-Penrose pseudo-inverse matrix, i.e. with different values of k.

Experimental results show some interesting facts: While BAM for $\alpha > 0$ perform better than EAM computed with $k = 1$ in the *NN* set, it does not perform better in the *VN* set. EAM with $k = 1$ has 1 degree of freedom corresponding to 1 parameter, the same as BAM. The parameter of EAM is tuned on the training set, in contrast to α, the parameter of BAM. Increasing the number of considered dimensions k of EAM, estimated models outperform BAM for all values of parameter α. Moreover, EAM detect a statistically significant difference between the positive and negative instances for $k = 10$ and $k = 20$ for the *NN* set and the *VN* set, respectively. Simple parametrisations of a BAM do not outperform the proposed estimated additive model.

6.5 Summary

We proposed a novel framework to investigate compositional distributional semantics (CDS) models, since existing CDS models have been investigated only with respect to word sequence similarity tests and lexical substitution tasks. The framework defines a

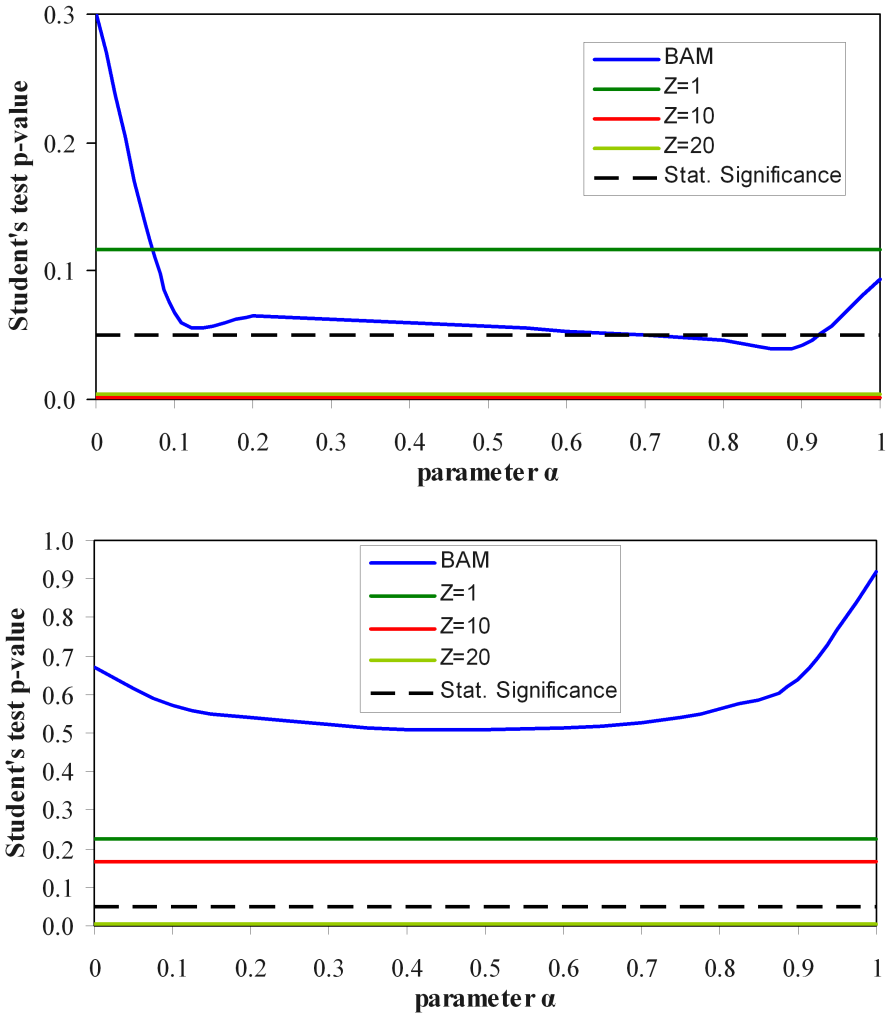

Figure 6.4: Probability of confusing positive and negative instances of *NN* (up) and *VN* (down) single words datasets when composing with BAM and EAM and for various values for parameter α (where $\beta = 1 - \alpha$)

new task to evaluate CDS models. According to this task, a good CDS model is expected to compose the context distributions of the components of a compositional sequence to produce a distribution similar to the context distribution derived from the occurrences of the sequence as a whole. At the same time, for non-compositional sequences, a good CDS model is expected to construct distributions significantly different from the context

distribution of the occurrences of the sequence as a whole.

The results of evaluating a number of state-of-the-art CDS models showed that none can perform well; but this was found to be on account of the fact that sequences appear very infrequently as a whole. Thus, no reliable context distributions can be derived. To tackle this data sparsity, we created new positive and negative examples from suitable dictionary entries. These instances work exactly as the multiword sequences, since their functionality is still based on the notion of compositionality. The only difference is that instead of comparing the composition result against the context distribution of the occurrences of a sequence, it is compared towards the context distribution of the occurrences of a single word.

Experimenting with this new evaluation set showed that several existing CDS models perform well on this new task: a basic model that employs selectional preferences and multiplies the distributional vectors of sequence constituents and more evidently a model that simply adds the distributional vectors of sequences constituents. The performance of CDS models is shown to depend highly on parameter values. In succession a new method for estimating the parameters of the basic additive model is proposed. The estimation problem is formed as a regression problem with multiple dependent variables. An approximate solution is computed using singular value decomposition and Moore-Penrose pseudo-inverse. Experiments showed that the CDS model that uses this parameter estimation method is highly competitive with respect to state-of-the-art models for compositional distributional semantics.

CHAPTER 7

Conclusion and Future Work

Executive Summary

This final chapter summarises the research work that was accomplished in this thesis. It discusses whether and to what extent the research hypothesis was fulfilled. Moreover, it presents a handful of open issues relevant to the subjects of the previous four chapters and proposes several directions for future work.

7.1 Thesis Summary

In this section, the basic research directions that were identified in the literature survey are summarised. In succession, we briefly describe the ways that these directions were explored in our research and the corresponding research outcomes and experimental results.

7.1.1 Literature Summary

In the literature review of this thesis, we performed a detailed survey of the research work in four major issues concerning the phenomenon of multiword expressions in natural language processing: (a) multiword expression and term recognition, (b) compositionality, (c) direct application to shallow parsing, and (d) indirect application to compositional

distributional semantics as a source of instances with desirable properties to build an evaluation framework.

Methods for extracting multiword expressions and terms, i.e. domain specific multiword expressions, were classified into linguistic, statistical and hybrid ones; the latter combine linguistic and statistical components and possibly other tools. Statistical methods quantify the properties of multiword expression candidates and their context and score each candidate separately. They are classified as unithood-based or termhood-based. The former assess the attachment strength of the candidate constituents, and the latter assess the degree that a candidate multiword expression refers to a specific concept. The state-of-the-art lacks an evaluation framework that will allow comparing between different methods. Although many methods have explored a variety of information sources, evaluation is usually done separately or against a few other methods found in the literature using different corpora and incompatible evaluation settings.

We then reviewed the literature that addresses the issue of deciding the compositionality of a given multiword expression. Methods were classified into those that compare the context distributions of the multiword expression and its components and to others based on substitutions. The latter are fewer than the former and address the issue indirectly: hypothesising that non-compositional multiword expressions are more rigid than compositional ones. All but two approaches, Fazly & Stevenson (2007) and Fazly et al. (2009), do not address the fact that the same multiword expression might have non-compositional and compositional uses. Fazly & Stevenson (2007) and Fazly et al. (2009) judge each instance of a multiword expression as compositional or not by comparing its context distribution to the context distribution of all instances together. Most methods found in the literature are evaluated on light verb constructions, verb-particle constructions, or verb-noun phrases. In contrast, multiword expressions consisting of adjectives and nouns are much less exploited.

The next objective of this thesis is to investigate how multiword expressions can be integrated into shallow parsing and whether and to what extent this integration contributes to shallow parsing accuracy. There is very limited work published on this issue, since most parsers ignore multiword expressions. There are two approaches of integrating multiword expressions in deep parsing. Zhang et al. (2006) adopted a "word with spaces" model (Sag et al. 2002), represented each multiword expressions as a new lexicon entry and showed a significant increase in coverage. Villavicencio et al. (2007)

argued that although the "word with spaces" approach of Zhang et al. (2006) enhances parser coverage, it affects the quality of the parse output detrimentally. Instead, they propose to integrate non-compositional multiword expressions, only, and achieved an increase in coverage similar to the one in Zhang et al. (2006) but added significantly less entries.

We located several disadvantages of the above approaches: Firstly, multiword expressions that were used in these experiments were judged as compositional or not using measures for extracting collocations. These measures correlate well but imperfectly with compositionality. Secondly, evaluations are small in size due to the cost of manually annotating parse output. Thirdly, there is no distinction among different types of multiword expressions, thus it is unclear which types affect parsing accuracy and coverage.

The last area of literature investigation in this thesis is distributional semantics composition. Several models addressing this task have been reviewed and three major issues were identified as potential fields of further research. Firstly, model parameters have not been exploited extensively. For example, the additive model in Mitchell & Lapata (2008) is evaluated hypothesising that always all features of a component word of a sequence contribute equally to the composed context distribution of the sequence. Secondly, state-of-the art models of the literature are evaluated on word sequence similarity tests, such as (Kintsch 2001), and lexical substitution tasks. They show low correlation with human annotated data, for which inter-annotation agreement is also low. These facts reveal that there is need to define a new task for evaluating CDS models. Thirdly, the models of the literature consist of various components and require a number of choices when implemented, such as the feature space of the context distribution. However, the models of the literature are usually evaluated following one of these options. It would be interesting to see if others can perform better. For example, the basic additive model of Mitchell & Lapata (2008) could be used to compose the environmental vectors of BEAGLE.

7.1.2 Research Summary

To address the limitations of the literature relevant to extracting multiword expressions and multiword terms, in chapter 3 we proposed an evaluation framework that allows comparing extraction methods under common settings. The framework consists of two evaluation corpora of the biomedical domain: *GENIA* and *PennBioIE*, evaluation metrics and a way to visualise the results.

Using this framework, several methods for linguistic filtering and for extracting collocations and multiword terms were evaluated. *Termhood-based* methods were shown to outperform *unithood-based* ones. This result should be assessed considering the fact that the corpora were domain-specific. In succession, *termhood-based* methods were analysed into components, each of which takes into account a source of useful information for term extraction. These components were evaluated separately to assess their contribution. Marginal frequency, i.e. the count of independent occurrences of a term, is shown to perform better than other more sophisticated *termhood-based* methods, under the current evaluation framework. In addition, marginal frequency is shown to improve performance of C-Value and Statistical barrier when integrated in the corresponding algorithms.

In chapter 4, we proposed a new method to address the problem of identifying whether a multiword expression is compositional or not. We focused on developing a method that takes into account that a multiword expression might have idiomatic (non-compositional) and non-idiomatic (compositional) uses. The method employs graph-based word sense induction to induce the senses of the multiword expression and its semantic head. Comparing the major senses of the multiword expression and its semantic head is adopted as an indicator of compositionality. We hypothesised that the major sense of a word or expression captures the sense that one thinks of when encountering it without context.

The proposed method was evaluated on English adjective-noun constructions, compound nominals and proper names, extracted from *WordNet*. We proposed a semi-supervised approach for resolving compositionality of multiword expressions occurring in *WordNet* that minimises human effort. It is shown that when the parameters of the system are estimated manually, sense induction can assist in identifying compositional multiword expressions. In succession, we propose an unsupervised scheme for estimating the parameters of graph-based sense induction systems. The scheme employs graph connectivity measures to score the clustering output in which a given parameter setting results. Using this scheme to estimate the parameters of the proposed system achieved comparable accuracy to selecting parameters manually.

In chapter 5, we investigated whether knowing which sequences are multiword expressions and of which type can contribute to shallow parsing. We adopted a "word with spaces" approach to integrate 116 multiword expressions extracted from *WordNet 3.0*; in

particular *compound nominals*, *proper names* or *adjective-noun constructions*. For each multiword expression we collected sentences from the web. In succession, they were part of speech tagged and shallow parsed.

The parse outputs before and after integrating multiword expression information were compared employing a detailed classification of changes. Every change that occurred in the parse output was assigned to a class, and for each class it was known beforehand whether the corresponding changes improve or deteriorate the parse output. The results showed that integrating non-compositional multiword expression information improves the shallow parse output more than integrating compositional ones. In particular, the contribution is shown to be significant for non-compositional adjective-noun multiword expressions.

In chapter 6, we proposed a novel framework to investigate compositional distributional semantics (CDS) models. The framework defines a new task, according to which a CDS model performs well if: (a) for each compositional sequence, it composes a distribution similar to the distribution derived from the occurrences of the sequence as a whole; and (b) for each non-compositional sequence, it composes a distribution different to the distribution of the occurrences of the sequence as a whole. The framework was tested on two datasets: the first one contains compositional and non-compositional multiword expressions and the second contains compositional and non-compositional equivalences between single words and sequences, extracted from suitable entries of *WordNet* and *Wiktionary*. The advantage of the latter dataset is that the original distribution, which is compared with the composed distribution, represents a single word and thus suffers less data sparsity.

Several state-of-the-art CDS models were evaluated under the proposed framework. On the dataset containing multiword expressions no CDS model performed well due to data sparsity. On the dataset containing equivalences of single word and sequences several existing CDS models perform well: a basic model that employs selectional preferences and multiplies the distributional vectors of sequence constituents and more evidently a model that simply adds the distributional vectors of sequence constituents. However, performance highly depends on parameter values. For this reason, a new supervised method of estimating the parameters of the basic additive model based on singular value decomposition and Moore-Penrose pseudo-inverse is proposed. Experiments showed that the resulting CDS model is very competitive with respect to existing CDS models.

7.2 Contributions

In the introduction of this thesis we stated the following hypothesis:

> *The tasks of recognising multiword expressions and deciding for their compositionality can be addressed in unsupervised manners, based on cooccurrence statistics and distributional semantics. Further, multiword expressions are beneficial for other fundamental applications of Natural Language Processing either by direct integration or as an evaluation tool.*

The hypothesis is stated so as to capture all four aspects that correspond to four important directions of this research relevant to multiword expressions. We believe that applications are equally important to the original tasks of multiword expressions, recognition and compositionality, because applications highlight the importance of the linguistic phenomenon to the other sub-fields of Natural Language Processing (NLP). Certainly, the evaluation results, that were summarised in the previous section, show that the hypothesis holds, i.e. that there are unsupervised ways of taking cooccurrence, nestedness, context and other types of information into account to recognise multiword expressions and decide for their compositionality and that multiword expressions can be beneficial for other NLP applications.

In the context of recognising domain-specific multiword expressions, we have shown that unithood-based measures, that use statistic tests and are originally developed to extract collocations, perform under the proposed evaluation framework worse than termhood-based measures, which are specially developed for domain-specific multiword expressions. Thus, extracting terms is significantly different to extracting collocations and requires special methods that carefully consider a wider range of information sources than statistical tests and cooccurrence frequencies. The proposed evaluation framework is designed to accommodate as many of the methods found in the literature as possible, even more than those that have already been evaluated. The contribution of this part of our work is mainly the conclusion that marginal frequency is a very efficient heuristic for extracting domain-specific multiword expressions, despite its simplicity.

Secondly, we have shown that there is much space for further work in the field of multiword terminology extraction, since methods found in the literature do not outperform marginal frequency. State-of the art methods have not yet succeeded in taking advantage other sources of information, such as context. NC-Value, the algorithm that

inputs the C-Value ranked list of candidate multiword terms and re-ranks it according to context information, did not achieve significantly different results than C-Value for both evaluation corpora. Moreover, the NC-Value output does not seem to be much affected from changing its parameter values. As a result, the method does not verify experimentally the theoretical intuitions on which it is based. This is a conclusion that holds for the proposed evaluation framework, and is a matter of further experimentation to investigate whether it generalises to other corpora and evaluation settings.

As far as the problem of deciding the compositionality of multiword expressions is concerned, we have shown that taking into account the different senses of multiword expressions and their components achieves an improvement over the standard approach of directly comparing the corresponding context distributions. The contribution is inline with a broad variety of research works which argue that partitioning the context distribution into separate distributions for each sense of the target word or sequence leads to a more accurate representation of context and better accuracy in various NLP tasks; e.g. Klapaftis (2008).

A separate contribution to research relevant to multiword expressions is the algorithm that estimates the compositionality of multiword expressions in *WordNet*. It can serve as a tool to minimise human effort in compositionality annotation, it is very easy to implement and also applicable to any synset hierarchy similar to *WordNet*, such as *WordNets* for languages other than English. Thus, it can be employed by researchers to develop their own datasets containing multiword expressions of any desirable parts of speech, given that they occur in some synset hierarchy; such as noun phrases containing prepositions, e.g. *Commission on Human Rights*.

Our attempt to integrate knowledge about multiword expressions in shallow parsing led to several contributions. As expected, we showed that shallow parsing can improve knowing about non-compositional multiword expressions while this is not the case for compositional ones. More evidently, non-compositional adjective-noun sequences are shown to contribute more than noun-noun sequences. These results are useful to the research community, especially to research relevant to parsing.

A second contribution from our experimentation with shallow parsing is the definition of change classes, which can provide statistical evidence useful for any further error analysis. Except for the application of change classes to evaluate the contribution of multiword expression in shallow parsing, they can serve to evaluate any other change

in the input or the implementation of the parser. Consequently, change classes can be seen as a more general tool for comparing the output of a shallow parser before and after some kind of change. This is straightforward, since change classes, as implemented and used in chapter 5, are agnostic of the event that occurred between the two shallow parses under comparison.

The contribution of our research to Compositional Distributional Semantics (CDS) is two-fold. Firstly, we proposed a new task for evaluating CDS models specially designed for this purpose, in contrast to previously employed evaluation tasks. This new task provides an excellent opportunity to facilitate the connection between theoretical CDS models and experimental evaluations. Despite the fact that the evaluation presented in this thesis considers sequences of two words only, the task is straightforwardly extensible to longer sequences. Of course, this extension requires new datasets of compositional and non-compositional sequences of the chosen length.

Secondly, we evaluated several state-of-the-art models of the literature and shown that the basic additive model and the basic multiplicative model with selectional preferences exhibit interesting potential. We have also shown that there is evident need to abandon simplistic parameterisations and instead estimate the parameters of CDS models. In specific we proposed a trainable CDS model, which is shown to perform better than existing CDS models. In general, we have structured the task of CDS modelling and opened a new direction in developing more efficient CDS models.

7.3 Future Work

In this section, we propose several directions for future work for each of the four major parts of this thesis. Directions mainly emerge from inspecting the contributions of the previous section and searching for ways to strengthen them.

7.3.1 Multiword expression Recognition

As discussed in the previous section, our evaluation on multiword term recognition showed that state-of-the-art methods do not successfully integrate context information. Thus, there is much space for developing new methods towards this direction. In addition, since domain-specific term extraction is proven to be a task significantly different to general-domain multiword expression recognition, it might be interesting to attempt

to integrate other sources of information in the recognition process, such as resolving abbreviations.

The proposed evaluation framework, although designed to accommodate as many approaches in the literature as possible, is still unable to accommodate some of them. For example, it hypothesises that candidate multiword expressions are generated by applying some linguistic filter on raw text. However, there are methods that skip linguistic filtering and apply directly on text. It would be interesting to investigate ways that these methods could also be comparably evaluated under the framework.

7.3.2 Compositionality analysis

The main disadvantage of the proposed system for resolving compositionality of a given multiword expression is its time deficiency. For each multiword expression, the system collects two corpora from the web: for the expression itself and for its semantic head. In a latter stage, it queries a web search engine to collect cooccurrence statistics useful to construct the graph. Both these processes are extremely time-consuming mainly due to network transfer delays.

Evidently, there is need to exploit clever ways to avoid collecting large web corpora and retrieving cooccurrence counts from web search engines. Potential success would allow speeding up the unsupervised process of deciding compositionality of a given multiword expression. Then, the system would be efficient enough to comprise a component in other Natural Language Processing tools, e.g. in a shallow parser with parallel non-compositional multiword expression recognition.

Another direction for future work emerges from the fact that the proposed system only uses the major senses of the input multiword expression. It would be interesting to investigate the other senses as well and output a list of the senses together with indications about the compositionality of each one.

7.3.3 Multiword expressions and shallow parsing

In our experimentation with integrating multiword expressions in shallow parsing, we followed a simple approach: we replaced the tokens of each multiword expression with a special made-up token consisting of the same tokens joined with underscores. It would be interesting to relax this hard-wiring heuristic. For example, one could inspect the possibility of assigning a probability that a sequence is a multiword expression and then

letting the shallow parser decide whether the most probable parse should include the sequence as a multiword expression or not. This approach to the problem is very relevant to the discussion in the previous subsection about integrating a multiword expression compositionality component in a shallow parser. If the component can be improved in terms of time-efficiency, multiword expression candidates of various lengths can be assessed for compositionality, and the distributional similarity value can be used as a compositionality estimate.

The extent to which our results generalise could be investigated further. A similar analysis could be performed using other shallow or possibly deep parsers. In the case of deep parsers the contribution of knowledge about multiword expressions is expected to be larger, since deep parsing is a more complicated task than shallow parsing.

7.3.4 Compositional distributional semantics models

In chapter 6, we proposed a novel method to estimate the parameters of the additive Compositional Distributional Semantics (CDS) model. Evaluation results for the CDS model whose parameters are estimated using the proposed method are very encouraging. Thus, it would be very interesting to exploit similar trainable parameter estimation models for the basic multiplicative CDS model and CDS models with selectional preferences. For these models, evaluation with simple parameterisations showed promising results.

Finally, a direction for extending the proposed evaluation framework is to investigate automatic methods for creating counter instances. In our framework, the positive instances of the multiword expression dataset consist of compositional multiword expressions while the positive instances of the single words dataset consist of equivalences of single words and compositional sequences. Negative examples consist of non-compositional multiword expressions and equivalences of single words and non-compositional sequences, respectively. The bottleneck in constructing the datasets is that negative instances are rare. However, choosing random equivalences is not suitable, since the sequences are expected to be totally unrelated to the single word, due to sparsity. It would be interesting to exploit other ways of creating negative instances in order to overcome the data sparsity problem.

References

Agirre, E., Ansa, O., Hovy, E., & Martínez, D. (2001). Enriching WordNet concepts with topic signatures. In *proceedings of the NAACL workshop on WordNet and Other lexical Resources: Applications, Extensions and Customizations*, Pittsburg, PA, USA.

Agirre, E., Baldwin, T., & Martínez, D. (2008). Improving parsing and PP attachment performance with sense information. In *proceedings of ACL-08: HLT*, (pp. 317–325)., Columbus, Ohio, USA. Association for Computational Linguistics.

Agirre, E., Martínez, D., de Lacalle, O. L., & Soroa, A. (2006). Two graph-based algorithms for state-of-the-art WSD. In *proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, (pp. 585–593)., Sydney, Australia. Association for Computational Linguistics.

Agirre, E. & Soroa, A. (2007a). Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, (pp. 7–12)., Prague, Czech Republic. Association for Computational Linguistics.

Agirre, E. & Soroa, A. (2007b). Ubc-as: A graph based unsupervised system for induction and classification. In *proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, (pp. 346–349)., Prague, Czech Republic. Association for Computational Linguistics.

Ananiadou, S. (2001). Automatic term recognition in biology. presentation slides:

personalpages.manchester.ac.uk/staff/sophia.ananiadou/PSBtut1.ppt, last visited on 2/08/2010.

Anastasiou, D., Hashimoto, C., Nakov, P., & Kim, S. N. (Eds.). (2009). *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*. Singapore: Association for Computational Linguistics.

Baldwin, T. (2006). Compositionality and multiword expressions: Six of one, half a dozen of the other? In *proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, (pp.1̃)., Sydney, Australia. Association for Computational Linguistics.

Baldwin, T., Bannard, C., Tanaka, T., & Widdows, D. (2003). An empirical model of multiword expression decomposability. In *proceedings of the ACL 2003 workshop on Multiword expressions*, (pp. 89–96)., Morristown, NJ, USA. Association for Computational Linguistics.

Baldwin, T., Bender, E. M., Flickinger, D., Kim, A., & Oepen, S. (2004). Road-testing the english resource grammar over the british national corpus. In *proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004*, (pp. 2047–2050).

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the web. In *proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Bannard, C. (2007). A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In *proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, (pp. 1–8)., Prague, Czech Republic. Association for Computational Linguistics.

Bannard, C., Baldwin, T., & Lascarides, A. (2003). A statistical approach to the semantics of verb-particles. In *proceedings of the ACL 2003 workshop on Multiword expressions*, (pp. 65–72)., Morristown, NJ, USA. Association for Computational Linguistics.

Baroni, M. & Lenci, A. (2009). One distributional memory, many semantic spaces. In *proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, (pp. 1–8)., Athens, Greece. Association for Computational Linguistics.

Biemann, C. (2006). Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *proceedings of TextGraphs: the Second Workshop on Graph Based Methods for Natural Language Processing*, (pp. 73–80)., New York City, NY, USA. Association for Computational Linguistics.

Bies, A., Ferguson, M., Katz, K., & MacIntyre, R. (1995). *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. Linguistic Data Consortium.

Bikel, D. M. (2000). A statistical model for parsing and word-sense disambiguation. In *proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, (pp. 155–163)., Morristown, NJ, USA. Association for Computational Linguistics.

Bille, P. (2005). A survey on tree edit distance and related problems. *Theoretical Computer Science*, *337*(1-3), 217–239.

Bourigault, D. (1992). Surface grammatical analysis for the extraction of terminological noun phrases. In *proceedings of the 14th conference on Computational linguistics*, (pp. 977–981)., Morristown, NJ, USA. Association for Computational Linguistics.

Brants, T. & Franz, A. (2006). Web 1t 5-gram corpus version 1. Technical report, Google Research.

Brill, E. (1992). A simple rule-based part of speech tagger. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, (pp. 112–116)., Morristown, NJ, USA. Association for Computational Linguistics.

Briscoe, T., Carroll, J., & Watson, R. (2006). The second release of the rasp system. In *proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia. Association for Computational Linguistics.

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., & Roossin, P. (1988). A statistical approach to language translation. In *proceedings of the 12th conference on Computational linguistics*, (pp. 71–76)., Morristown, NJ, USA. Association for Computational Linguistics.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *proceedings of the 1st NAACL*, (pp. 132–139)., Seattle, Washington, USA. Association for Computational Linguistics.

Choueka, Y. (1988). Looking for needles in a haystack or locating interesting collocational expressions in large textual databases. In *RIAO*, (pp. 609–624).

Church, K. W. & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, *16*(1), 22–29.

Clark, S. & Pulman, S. (2007). Combining symbolic and distributional models of meaning. In *proceedings of the AAAI Spring Symposium on Quantum Interaction*, (pp. 52–55).

Collins, M. J. (1999). *Head-driven statistical models for natural language parsing*. PhD thesis, Philadelphia, PA, USA.

Cook, P., Fazly, A., & Stevenson, S. (2007). Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, (pp. 41–48)., Prague, Czech Republic. Association for Computational Linguistics.

Cook, P., Fazly, A., & Stevenson, S. (2008). The VNC-Tokens Dataset. In *proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, Marrakech, Morocco.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory* (99th ed.). Wiley-Interscience.

Dagan, I., Lee, L., & Pereira, F. C. N. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning*, *34*(1-3), 43–69.

Daille, B., Gaussier, E., & Lange, J. (1994). Towards automatic extraction of monolingual and bilingual terminology.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*(6), 391–407.

Dias, G., Kaalep, H., & Muischnek, K. (2001). Automatic extraction of verb phrases from annotated corpora: A linguistic evaluation for estonian. In *Workshop on Collocation of the joint 39th Annual Meeting of the Association of Computational Linguistics and 10th Conference of the European Chapter of the Association of Computational Linguistics (EACL/ACL 2001)*, Toulouse, France. Association for Computational Lin-

guistics.

Dorow, B. & Widdows, D. (2003). Discovering corpusspecific word senses. In *proceedings 10th conference of the European chapter of the ACL*, (pp. 79–82)., Budapest, Hungary. Association for Computational Linguistics.

Dowding, J., Moore, R., Andryt, F., & Moran, D. (1994). Interleaving syntax and semantics in an efficient bottom-up parser. In *proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, (pp. 110–116)., Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Dunning, T. E. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, *19*(1), 61–74.

Erk, K. (2007). A simple, similarity-based model for selectional preferences. In *proceedings of 45th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Erk, K. & Padó, S. (2008). A structured vector space model for word meaning in context. In *proceedings of EMNLP*, Honolulu, HI. Association for Computational Linguistics.

Essen, U. & Steinbiss, V. (1992). Cooccurrence smoothing for stochastic language modeling. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, (pp. 161–164 vol.1).

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Danielweld, & Yates, A. (2004). Methods for domain-independent information extraction from the web: An experimental comparison. In *The 19th National Conference on Artificial Intelligence (AAAI-04)*, (pp. 391–398)., San Jose, California.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., & Yates, A. (2005). Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, *165*(1), 91–134.

Evert, S. & Krenn, B. (2001). Methods for the qualitative evaluation of lexical association measures. In *proceedings of 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, (pp. 188–195)., Morristown, NJ, USA. Association for Computational Linguistics.

Fallucchi, F. & Zanzotto, F. M. (2009). SVD feature selection for probabilistic taxonomy

learning. In *proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, (pp. 66–73)., Athens, Greece. Association for Computational Linguistics.

Fazly, A., Cook, P., & Stevenson, S. (2009). Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, *35*(1), 61–103.

Fazly, A. & Stevenson, S. (2006). Automatically constructing a lexicon of verb phrase idiomatic combinations. In *proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, (pp. 337–344)., Trento Italy. Association for Computational Linguistics.

Fazly, A. & Stevenson, S. (2007). Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures. In *proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, (pp. 9–16)., Prague, Czech Republic. Association for Computational Linguistics.

Feldman, R., Fresko, M., Kinar, Y., Lindell, Y., Liphstat, O., Rajman, M., Schler, Y., & Zamir, O. (1998). Text mining at the term level. In *principles of Data Mining and Knowledge Discovery*, (pp. 65–73).

Fellbaum, C. (Ed.). (2006). *Collocations and Idioms: Linguistic, lexicographic, and computational aspects*. Berlin, Germany: London: Continuum Press.

Firth, J. R. (1957.). *Papers in Linguistics.* London: Oxford University Press.

Frantzi, K. T., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms: the c-value/nc-value method. *International Journal on Digital Libraries*, *3*(2), 115–130.

Fujita, S., Bond, F., Oepen, S., & Tanaka, T. (2007). Exploiting semantic information for hpsg parse selection. In *DeepLP '07: Proceedings of the Workshop on Deep Linguistic Processing*, (pp. 25–32)., Morristown, NJ, USA. Association for Computational Linguistics.

Giesbrecht, E. (2009). In search of semantic compositionality in vector spaces. In *ICCS '09: Proceedings of the 17th International Conference on Conceptual Structures*, (pp. 173–184)., Berlin, Heidelberg. Springer-Verlag.

Golub, G. & Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B:*

Numerical Analysis, *2*(2), 205–224.

Grégoire, N., Evert, S., & Kim, S. N. (Eds.). (2007). *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*. Prague, Czech Republic: Association for Computational Linguistics.

Grégoire, N., Evert, S., & Krenn, B. (Eds.). (2008). *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions*. Marrakesh, Morocco: European Language Resources Association (ELRA).

Gu, B. (2006). Recognizing nested named entities in genia corpus. In *proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, (pp. 112–113)., New York, New York. Association for Computational Linguistics.

Guevara, E. (2010). A regression model of adjective-noun compositionality in distributional semantics. In *proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, (pp. 33–37)., Uppsala, Sweden. Association for Computational Linguistics.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*, 146–162.

Harris, Z. S. (1964). Distributional structure. In Katz, J. J. & Fodor, J. A. (Eds.), *The Philosophy of Linguistics*, New York. Oxford University Press.

Hatzivassiloglou, V. (1994). Do we need linguistics when we have statistics? a comparative analysis. In Klavans, J. L. & Resnik, P. (Eds.), *the Balancing Act*, volume 4, (pp. 67–94).

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. Technical Report S2K-92-09.

Hearst, M. A. (1998). Automated discovery of wordnet relations. In *C. Fellbaum, WordNet: An Electronic Lexical Database* (pp. 131–153). MIT Press.

Heid, U. (1999). A linguistic bootstrapping approach to the extraction of term candidates from german text.

Hektoen, E. (1997). Probabilistic parse selection based on semantic cooccurrences. In *5th International workshop on parsing technologies (IWPT-97)*, (pp. 113–122).

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). Ontonotes:

The 90% solution. In *proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, (pp. 57–60)., New York City, NY, USA. Association for Computational Linguistics.

Ingram, L. & Curran, J. R. (2007). Distributional similarity of multi-word expressions. In *Australasian Language Technology Workshop (ALTW)*, (pp. 146–148).

Jackendoff, R. S. (1997). *The Architecture of the Language Faculty.* MIT Press.

Jones, M. N. & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*(1), 1–37.

Justeson, J. & Katz, S. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 9–27.

Kageura, K. & Umino, B. (1996). Methods of automatic term recognition: a review. *Terminology*, *3*(2), 259–289.

Karakos, D., Eisner, J., Khudanpur, S., & Priebe, C. (2007). Cross-instance tuning of unsupervised document clustering algorithms. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, (pp. 252–259)., Rochester, New York. Association for Computational Linguistics.

Karov, Y. & Edelman, S. (1998). Similarity-based word sense disambiguation. *Computational Linguistics*, *24*(1), 41–59.

Katz, G. & Giesbrecht, E. (2006). Automatic identification of non-compositional multiword expressions using latent semantic analysis. In *proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, (pp. 12–19)., Sydney, Australia. Association for Computational Linguistics.

Kintsch, W. (2001). Predication. *Cognitive Science*, *25*(2), 173–202.

Klapaftis, I. P. (2008). *Unsupervised Concept Hierarchy Induction: Learning The Semantics Of Words*. PhD thesis, Department of Computer Science, University of York, York, UK.

Klapaftis, I. P. & Manandhar, S. (2008). Word sense induction using graphs of collocations. In *proceedings of the 18th European Conference on Artificial Intelligence, (ECAI-2008)*, Patras, Greece.

Klein, D. & Manning, C. D. (2003). Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, (pp. 423–430)., Morristown, NJ, USA. Association for Computational Linguistics.

Korhonen, A., Bond, F., McCarthy, D., & Villavicencio, A. (Eds.). (2003). *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. Sapporo, Japan: Association for Computational Linguistics.

Korkontzelos, I., Klapaftis, I., & Manandhar, S. (2009). Graph connectivity measures for unsupervised parameter tuning of graph-based sense induction systems. In *proceedings of the NAACL-2009 Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, Boulder, Colorado, USA.

Krenn, B. (2000). *The Usual Suspects: Data-Oriented Models for the Identification and Representation of Lexical Collocations.* PhD thesis, DFKI & Universität des Saarlandes, Saarbrücken.

Kulick, S., Bies, A., Liberman, M., Mandel, M., Mcdonald, R., Palmer, M., Schein, A., Ungar, L., Winters, S., & White, P. (2004). Integrated annotation for biomedical information extraction. In Hirschman, L. & Pustejovsky, J. (Eds.), *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, (pp. 61–68)., Boston, Massachusetts, USA. Association for Computational Linguistics.

Landauer, T. K. & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

Laporte, E. & Voyatzi, S. (2008). An Electronic Dictionary of French Multiword Adverbs. In *Language Resources and Evaluation Conference. Workshop Towards a Shared Task on Multiword Expressions*, (pp. 31–34).

Lee, L. (1999). Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, (pp. 25–32).

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, *20/1*, 1–31.

Li, P., Burgess, C., & Lund, K. (2000). The acquisition of word meaning through global

lexical co-occurrences. In *proceedings of the 31st Child Language Research Forum*.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *COLING-ACL*, (pp. 768–774).

Lin, D. (1998b). An information-theoretic definition of similarity. In *proceedings of the 15th International Conference on Machine Learning*, (pp. 296–304).

Lin, D. (1999). Automatic identification of non-compositional phrases. In *proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, (pp. 317–324)., College Park, Maryland, USA. Association for Computational Linguistics.

Lin, D. & Pantel, P. (2001). DIRT-discovery of inference rules from text. In *proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, San Francisco, CA.

Lin, J. (2002). Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, *37*(1), 145–151.

Liu, B. (2007). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer.

Manning, C. & Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*, chapter 5: Collocations. MIT Press.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press.

Mason, Z. J. (2004). Cormet: a computational, corpus-based conventional metaphor extraction system. *Computational Linguistics*, *30*(1), 23–44.

Maynard, D. & Ananiadou, S. (2000a). Identifying terms by their family and friends. In *proceedings of the 18th conference on Computational linguistics*, (pp. 530–536)., Morristown, NJ, USA. Association for Computational Linguistics.

Maynard, D. & Ananiadou, S. (2000b). Trucks: a model for automatic term recognition. *Journal of Natural Language Processing*.

McCarthy, D. (2006). Automatic methods to detect the compositionality of multiwords. presentation slides. url: www.sunum.org/myfiles/B2/McCarthyCollocIdioms06.ppt last accessed: 28/11/2009.

McCarthy, D. & Carroll, J. (2003). Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, *29*(4), 639–654.

McCarthy, D., Keller, B., & Carroll, J. (2003). Detecting a continuum of compositionality in phrasal verbs. In *proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*.

McCarthy, D., Venkatapathy, S., & Joshi, A. (2007). Detecting compositionality of verb-object combinations using selectional preferences. In *proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, (pp. 369–379)., Prague, Czech Republic. Association for Computational Linguistics.

McInnes, B. T. (2004). Extending the log likelihood measure to improve collocation identification. Master's thesis, University of Minnesota.

Mikheev, A., Moens, M., & Grover, C. In *proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Bergen, Norway. Association for Computational Linguistics.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Miller, G. A. & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, *6*(1), 1–28.

Mitchell, J. & Lapata, M. (2008). Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, (pp. 236–244)., Columbus, Ohio, USA. Association for Computational Linguistics.

Moirón, B. n. V., Villavicencio, A., McCarthy, D., Evert, S., & Stevenson, S. (Eds.). (2006). *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. Sydney, Australia: Association for Computational Linguistics.

Moon, R. (1998). *Fixed Expressions and Idioms in English. A Corpus-based Approach.* Oxford: Clarendon Press.

Munoz, M., Punyakanok, V., Roth, D., & Zimak, D. (1999). A learning approach to shal-

low parsing. In *EMNLP-VLC, the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, (pp. 168–178).

Nadeau, D., Turney, P., & Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *19th Canadian Conference on Artificial Intelligence*, Québec City, Québec, Canada.

Nakagawa, H. (2000). Automatic term recognition based on statistics of compound nouns. *Terminology*, *6*(2), 195–210.

Nakagawa, H. & Mori, T. (2002). A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002*, (pp. 1–7)., Morristown, NJ, USA. Association for Computational Linguistics.

Navigli, R. & Lapata, M. (2007). Graph connectivity measures for unsupervised word sense disambiguation. In *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, (pp. 1683–1688)., Hyderabad, India.

Nunberg, G., Wasow, T., & Sag, I. A. (1994). Idioms. *Language*, *70*(3), 491–539.

Ordelman, R. J. F. (2002). Twente nieuws corpus (twnc). Technical report.

Pado, S. & Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, *33*(2), 161–199.

Pecina, P. & Schlesinger, P. (2006). Combining association measures for collocation extraction. In *ACL, 2006*.

Penrose, R. (1955). A generalized inverse for matrices. In *proceedings of Cambridge Philosophical Society*.

Piao, S. S. L., Rayson, P., Mudraya, O., Wilson, A., & Garside, R. (2006). Measuring mwe compositionality using semantic annotation. In *MWE '06: Proceedings of the Workshop on Multiword Expressions*, (pp. 2–11)., Morristown, NJ, USA. Association for Computational Linguistics.

Pollard, C. & Sag, I. (1994). *Head-driven Phrase Structured Grammar*. Stanford: Chicago CSLI.

Radev, D., Teufel, S., Saggion, H., Lam, W., Blitzer, J., Qi, H., Elebi, A., Liu, D., & Drabek, E. (2003). Evaluation challenges in large-scale document summarization.

Rao, C. R. (1982). Diversity: Its measurement, decomposition, apportionment and analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, *44*(1), 1–22.

Rayson, P., Piao, S., Sharoff, S., Evert, S., & Moirón, B. n. V. (2009). Multiword expressions: hard going or plain sailing? *Language Resources and Evaluation*.

Rayson, P., Sharoff, S., & Adolphs, S. (Eds.). (2006). *Proceedings of the EACL Workshop on Multi-word-expressions in a multilingual context*. Trento, Italy: Association for Computational Linguistics.

Resnik, P. (1993). *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania.

Rohde, D. L. T. (2004). *TGrep2 User Manual*. Available at http://tedlab.mit.edu /~dr/Tgrep2.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A. A., & Flickinger, D. (2002). Multiword expressions: A pain in the neck for nlp. In *CICLing*, (pp. 1–15).

Salton, G. & McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Schone, P. & Jurafsky, D. (2001). Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In Lee, L. & Harman, D. (Eds.), *proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, (pp. 100–108). Association for Computational Linguistics.

Schulte im Walde, S. (2003). A collocation database for german verbs and nouns. In *proceedings of the 7th Conference on Computational Lexicography and Text Research*, Budapest, Hungary.

Séaghdha, D. . & Copestake, A. (2007). Co-occurrence contexts for noun compound interpretation. In *proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, (pp. 57–64)., Prague, Czech Republic. Association for Computational Linguistics.

Shimohata, S., Sugio, T., & Nagata, J. (1997). Retrieving collocations by co-occurrences

and word order constraints. In Cohen, P. R. & Wahlster, W. (Eds.), *proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, (pp. 476–481)., Somerset, New Jersey. Association for Computational Linguistics.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*(1-2), 159–216.

Subramaniam, V. L., Mukherjea, S., Kankar, P., Srivastava, B., Batra, V. S., Kamesam, P. V., & Kothari, R. (2003). Information extraction from biomedical literature: methodology, evaluation and an application. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, (pp. 410–417)., New York, NY, USA. ACM Press.

Tanaka, T., Villavicencio, A., Bond, F., & Korhonen, A. (Eds.). (2004). *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*. Barcelona, Spain: Association for Computational Linguistics.

Tapanainen, P., Piitulainen, J., & Järvinen, T. (1998). Idiomatic object usage and support verbs. In *proceedings of the 17th international conference on Computational linguistics*, (pp. 1289–1293)., Morristown, NJ, USA. Association for Computational Linguistics.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., & Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. *Advances in Informatics*, *3746*, 382–392.

Turney, P. D. (2003). Coherent keyphrase extraction via web mining. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, (pp. 434–439)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Van de Cruys, T. & Moirón, B. n. V. (2007). Semantics-based multiword expression extraction. In *proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, (pp. 25–32)., Prague, Czech Republic. Association for Computational Linguistics.

van Noord, G. (2006). At last parsing is now operational. In Mertens, P., Fairon, C., Dister, A., & Watrin, P. (Eds.), *TALN06. Verbum Ex Machina. Actes de la 13e conference*

sur le traitement automatique des langues naturelles, (pp. 20–42)., Leuven, Belgium.

van Rijsbergen, C. J. (1979). *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow.

Venkatapathy, S. & Joshi, A. K. (2005). Measuring the relative compositionality of verb-noun (v-n) collocations by integrating features. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, (pp. 899–906)., Morristown, NJ, USA. Association for Computational Linguistics.

Veronis, J. (2004). Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, *18*(3), 223–252.

Villavicencio, A., Copestake, A., Waldron, B., & Lambeau, F. (2004). Lexical encoding of mwes. In *MWE '04: Proceedings of the Workshop on Multiword Expressions*, (pp. 80–87)., Morristown, NJ, USA. Association for Computational Linguistics.

Villavicencio, A., Kordoni, V., Zhang, Y., Idiart, M., & Ramisch, C. (2007). Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In *proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, (pp. 1034–1043)., Prague, Czech Republic. Association for Computational Linguistics.

Vivaldi, J., Màrquez, L., & Rodríguez, H. (2001). Improving term extraction by system combination using boosting. *Lecture Notes in Computer Science*, *2167*, 515–526.

Wang, G., Zhang, H., Wang, H., & Yu, Y. (2007). Enhancing relation extraction by eliciting selectional constraint features from wikipedia. *Natural Language Processing and Information Systems*, *4592*, 329–340.

Weeds, J. & Weir, D. (2005). Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, *31*(4), 439–475.

Weeds, J., Weir, D., & McCarthy, D. (2004). Characterising measures of lexical distributional similarity. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics.

Wermter, J. & Hahn, U. (2004). Collocation extraction based on modifiability statistics. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Craig (1999). Kea: Practical automatic keyphrase extraction. In *The ACM Digital Library*, (pp. 254–255).

Xiong, D., Li, S., Liu, Q., Lin, S., & Qian, Y. (2005). Parsing the penn chinese treebank with semantic knowledge. In *The 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, (pp. 70–81)., Jeju Island, Korea.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, (pp. 189–196).

Zesch, T. & Gurevych, I. (2007). Analysis of the wikipedia category graph for NLP applications. In *proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*, (pp. 1–8)., Rochester, NY, USA. Association for Computational Linguistics.

Zhang, Y., Kordoni, V., Villavicencio, A., & Idiart, M. (2006). Automated multiword expression prediction for grammar engineering. In *MWE '06: Proceedings of the Workshop on Multiword Expressions*, (pp. 36–44)., Morristown, NJ, USA. Association for Computational Linguistics.

Citation Index

Agirre & Soroa (2007a), 141, 149, 151, 165
Agirre & Soroa (2007b), 141
Agirre et al. (2001), 140
Agirre et al. (2006), 78, 140, 141
Agirre et al. (2008), 170
Ananiadou (2001), 31
Anastasiou et al. (2009), 22
Baldwin et al. (2003), 18, 21, 82, 89, 93–95
Baldwin et al. (2004), 168
Baldwin (2006), 18, 21, 92, 135, 172
Banko et al. (2007), 39
Bannard et al. (2003), 89, 92, 136
Bannard (2007), 63, 69, 75
Baroni & Lenci (2009), 76, 100
Biemann (2006), 144
Bies et al. (1995), 174
Bikel (2000), 169
Bille (2005), 178
Bourigault (1992), 39, 74
Brants & Franz (2006), 143
Brill (1992), 74
Briscoe et al. (2006), 69, 74, 75
Brown et al. (1988), 46
Charniak (2000), 213
Choueka (1988), 30
Church & Hanks (1990), 49, 74, 94, 114
Clark & Pulman (2007), 101
Collins (1999), 75
Cook et al. (2007), 71, 72, 75, 90
Cook et al. (2008), 90
Cover & Thomas (1991), 84
Dagan et al. (1999), 82, 85, 86
Daille et al. (1994), 50, 74
Deerwester et al. (1990), 81, 100, 200, 208
Dias et al. (2001), 114
Dorow & Widdows (2003), 141
Dowding et al. (1994), 170
Dunning (1993), 46, 74, 114, 143
Erk & Padó (2008), 102, 104–106, 109, 201, 202
Erk (2007), 36, 106

Essen & Steinbiss (1992), 85
Etzioni et al. (2004), 39
Etzioni et al. (2005), 39
Evert & Krenn (2001), 51, 114
Fallucchi & Zanzotto (2009), 203–205, 208, 209
Fazly & Stevenson (2006), 66, 67, 72, 75, 90, 97, 136
Fazly & Stevenson (2007), 67–69, 75, 90, 97, 136, 138, 229
Fazly et al. (2009), 36, 38, 75, 90, 97, 136, 138, 229
Feldman et al. (1998), 63
Fellbaum (2006), 22
Firth (1957), 81, 200
Frantzi et al. (2000), 52, 55, 56, 74, 114, 117, 130
Fujita et al. (2007), 170
Giesbrecht (2009), 202
Golub & Kahan (1965), 209
Grégoire et al. (2007), 22
Grégoire et al. (2008), 22
Guevara (2010), 202, 203
Gu (2006), 116, 119
Harris (1954), 81, 98, 99, 200
Harris (1964), 81, 200
Hatzivassiloglou (1994), 85
Hearst (1992), 36
Hearst (1998), 36
Heid (1999), 39
Hektoen (1997), 170
Hovy et al. (2006), 165
Ingram & Curran (2007), 88
Jackendoff (1997), 21
Jones & Mewhort (2007), 98, 102, 107–110, 201, 202, 215
Justeson & Katz (1995), 37, 74, 117, 118, 122
Kageura & Umino (1996), 30, 31, 40, 52, 113, 118
Karakos et al. (2007), 151
Karov & Edelman (1998), 86
Katz & Giesbrecht (2006), 89, 95
Kintsch (2001), 99, 110, 199, 201, 230
Klapaftis & Manandhar (2008), 141, 144, 145, 151, 164, 165
Klapaftis (2008), 234
Klein & Manning (2003), 36, 139
Korhonen et al. (2003), 21
Korkontzelos et al. (2009), 164
Krenn (2000), 95
Kulick et al. (2004), 116, 119
Landauer & Dumais (1997), 110
Laporte & Voyatzi (2008), 92, 172
Lee (1999), 82, 85, 86, 146
Lenci (2008), 99
Li et al. (2000), 81, 100, 200
Lin & Pantel (2001), 98, 100, 201
Lin (1998a), 82, 84–87, 90, 94
Lin (1998b), 84
Lin (1999), 90, 92, 94
Lin (2002), 84
Liu (2007), 209
Manning & Schutze (1999), 3, 17, 21, 30, 45, 74, 125
Manning et al. (2008), 81, 200

Mason (2004), 36
Maynard & Ananiadou (2000a), 52, 56, 64, 66, 74
Maynard & Ananiadou (2000b), 63, 64, 66, 74
McCarthy & Carroll (2003), 36, 81, 200
McCarthy et al. (2003), 82, 89, 90, 93–95
McCarthy et al. (2007), 36, 96
McCarthy (2006), 82
McInnes (2004), 49, 125
Mikheev, Moens & Grover (Mikheev et al.), 39, 120
Miller & Charles (1991), 81, 82, 200
Miller (1995), 27, 75, 147, 188, 213, 218
Mitchell & Lapata (2008), 98–100, 102–107, 109, 110, 201, 202, 210, 230
Moirón et al. (2006), 22
Moon (1998), 18
Munoz et al. (1999), 190, 196, 197
Nadeau et al. (2006), 39
Nakagawa & Mori (2002), 52, 58, 74
Nakagawa (2000), 52, 58, 59, 74, 114, 130
Navigli & Lapata (2007), 151, 155
Nunberg et al. (1994), 20, 21
Ordelman (2002), 75
Pado & Lapata (2007), 76, 81, 100, 200
Pecina & Schlesinger (2006), 51, 74, 86, 114
Penrose (1955), 203, 208, 209
Piao et al. (2006), 95
Pollard & Sag (1994), 105
Radev et al. (2003), 120
Rao (1982), 84
Rayson et al. (2006), 22
Rayson et al. (2009), 17
Resnik (1993), 36, 76, 100
Rohde (2004), 75
Séaghdha & Copestake (2007), 37, 74
Sag et al. (2002), 18, 21, 168, 169, 171, 229
Salton & McGill (1986), 83
Salton et al. (1975), 75, 76
Schone & Jurafsky (2001), 65, 74, 82, 89, 91, 92
Schulte im Walde (2003), 36, 63, 73
Shimohata et al. (1997), 25, 52, 60, 62, 63, 74
Smolensky (1990), 101
Subramaniam et al. (2003), 63
Tanaka et al. (2004), 22
Tapanainen et al. (1998), 89
Tsuruoka et al. (2005), 33, 190
Turney (2003), 31
Van de Cruys & Moirón (2007), 63, 70, 71, 75
Venkatapathy & Joshi (2005), 90, 94–96
Veronis (2004), 140, 141
Villavicencio et al. (2004), 39
Villavicencio et al. (2007), 96, 97, 168–170, 229
Vivaldi et al. (2001), 65, 74
Wang et al. (2007), 36

Weeds & Weir (2005), 86
Weeds et al. (2004), 86, 87
Wermter & Hahn (2004), 116, 124
Witten et al. (1999), 31
Xiong et al. (2005), 170
Yarowsky (1995), 145
Zesch & Gurevych (2007), 151
Zhang et al. (2006), 168, 169, 229, 230
van Noord (2006), 75
van Rijsbergen (1979), 84