Extraction of Linguistic Resources from Multilingual Corpora and their Exploitation

AHMAD RAZA SHAHID

Ph.D. Thesis

The University of York
Artificial Intelligence Group
Department of Computer Science
United Kingdom

9th February 2012

Abstract

Increasing availability of on-line and off-line multilingual resources along with the developments in the related automatic tools that can process this information, such as GIZA++ (Och & Ney 2003), has made it possible to build new multilingual resources that can be used for NLP/IR tasks.

Lexicon generation is one such task, which if done by hand is quite expensive with human and capital costs involved. Generation of multilingual lexicons can now be automated, as is done in this research work. Wikipedia[1], an on-line multilingual resource was gainfully employed to automatically build multilingual lexicons using simple search strategies.

Europarl parallel corpus (Koehn 2002) was used to create multilingual sets of synonyms, that were later used to carry out the task of Word Sense Disambiguation (WSD) on the original corpus from which they were derived. The theoretical analysis of the methodology validated our approach.

The multilingual sets of synonyms were then used to learn unsupervised mod-

[1] http://www.wikipedia.org/

els of word morphology in the individual languages. The set of experiments we carried out, along with another unsupervised technique, were evaluated against the gold standard. Our results compared very favorably with the other approach. The combination of the two approaches gave even better results.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost I would like to thank my supervisor, Dr Dimitar Kazakov, without whose help and encouragement this thesis would not have been possible. He has been a beacon of light for me throughout the course of this degree.

I would also like to thank my internal assessor, James Cussens, for his rigorous assessment of my work, which helped me in bringing my research to fruition.

I am grateful to my colleagues and friends for their help and support. For that I am indebted to Shengping Xia, Burcu Can, Ioannis Klapaftis, Waleed Alsanie, Tasawer Khan, Juliette Martin, Tom Lampert, Saad S. Khan, Abdul Haseeb Malik, and Muhammad Haseeb.

I owe a great deal to my parents, Khalil Akhtar Shahid and Musarrat Jehan Ara, my sister Fatima Shahid, and especially to my wife, Gulmina Rextina, who provided me with all the moral support needed through the viccissitudes of my PhD.

I would also like to thank the Higher Education Commission, Pakistan, who provided me with all the financial support I needed.

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 .....

Publications:

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

Kazakov, D. and Shahid, A. (2008). Extracting Multilingual Dictionaries for the teaching of CS and AI. In *4th UK Workshop on AI in Education* as part of the annual SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK.

Kazakov, D. and Shahid, A. (2009). Unsupervised Construction of a Multilingual WordNet from Parallel Corpora. In *Workshop on Natural Language Processing methods and Corpora in Translation, Lexicography, and Language Learning (RANLP '09)*, Borovets, Bulgaria.

Shahid, A. and Kazakov, D. (2009). Automatic Multlingual Lexicon Generation using Wikipedia as a resource. In *Proceedings of the International Conference on Agents and Artificial Intelligence, (ICCART '09)*, Porto, Portugal.

Shahid, A. and Kazakov, D. (2010). Retrieving Lexical Semantics from Parallel Corpora. *Polibits, 5*, 25-28.

Shahid, A. and Kazakov, D. (2011). Using Multilingual Corpora to Extract Semantic Information. In *Proceedings of the Symposium on Learning Language Models from Multilingual Corpora, AISB'11 Convention*, York, UK.

I dedicate my thesis to the living memories of my late grandfather, Mirza Manzur-ul-Aziz Baig

CHAPTER 1

Introduction and Motivation

1.1 Initial Motivation

Over the past two decades the web has been transformed from a limited resource available to the lucky few, who could afford it or were working on the technology behind it, to an immense resource rich in all kinds of information. But its size causes its own problems, one of which is the inability to extract useful information from this plethora of written material. The increasingly more advanced hardware is making it possible to process all this information relatively faster. That is where Natural Language Processing (NLP) comes in. It is a technology for automated processing of natural languages, which uses computational devices to extract useful relevant information.

That makes NLP an important task and with increasing information, it is becoming increasingly important. Without useful automated techniques, this vast

pool of information would remain exactly that and its size would lose its meaning.

A lot of work has already gone into NLP but still there are a number of tasks which may take a number of years before they could be considered as resolved, such as Machine Translation (MT), and Word Sense Disambiguation (WSD) to name a few.

Despite the fact that a lot of work has already been done on NLP, the multilingual aspect has still not been fully explored and no truly reliable commercial tools have been developed that could replace the human effort required for such tasks with machines.

The motivation behind this research is to see how a multilingual corpus can be used to:

- extract new lexical resources;
- see how these lexical resources can assist in NLP or Information Retrieval (IR), where documents are retrieved based on a query, tasks in the original multilingual corpus from which they were derived.

1.2 NLP and Information Retrieval (IR)

Though IR is considered to be a different task than the subtasks of NLP, it relies on some of the NLP methods and approaches, such as stemming.

1.2.1 NLP

NLP has a long history with a number of people spending years on building useful real world applications. It has some success stories specially where NLP has been employed in daily tasks, such as spell checker and grammar checker.

NLP is a fairly complex problem and needs background knowledge in linguistics, machine learning and statistics. That requires a long learning curve which makes the talent pool working in it relatively scarce. Also, one important ingredient is missing from most NLP tasks, its interface with speech. Ignoring how humans express themselves and only focussing on language analysis breaks a crucial link in the evolution of NLP technologies, making it a very difficult task to develop machines passing the famous Turing Test that could communicate with humans on equal footing (Turing 1950). Till that is achieved a lot of effort needs to be put into designing and implementing systems that can incrementally increase the performance of NLP tasks.

NLP covers a wide range of subtasks, such as WSD, MT, morphological analysis Question Answering (QA), Sentiment Analysis, and Part-of-Speech (POS) tagging, to name but a few. A number of techniques and models have been used to achieve the above mentioned tasks, such as statistical methods and graphical models.

NLP tasks require rich text resources. Over the years, specially with the advent of the internet, such resources are available in vast numbers and are often free. Corpus based approaches can help in gathering statistics related to use of language constructs in a real world environment.

1.2.2 Corpora based Approaches

Over the years a number of mono- and multi-lingual corpora have emerged. Starting from the Brown Corpus (Francis 1964), an American English corpus that covered 1 million words. Subsequent efforts include the British National Corpus (BNC) (Burnard & Aston 1998), a 100 million word corpus. Such monolingual corpora can obviously help in mono-lingual tasks. Both the mentioned corpora were not annotated with lexical semantic information. An example of an

annotated mono-lingual corpus is Wall Street Journal (WSJ) Treebank (Marcus et al. 1993) which is annotated with syntactic tree structures for all sentences in the corpus.

The annotations provide auxiliary information that is helpful when supervised learning approaches are used for training and testing purposes. It can also act as a gold standard for evaluation purposes. The unannotated corpora lack in auxiliary information and supervised approaches are not suited for such corpora. Unsupervised approaches on the other hand can take into account the statistical information hidden within the corpus and thus need no information provided by experts.

Un-annotated corpora can also be developed more cheaply since no expert knowledge is required to provide annotations. Annotated corpora on the other hand do not come cheap. A certain level of consensus is required among experts before the annotated corpus could be used as a gold standard. That puts an extra cost on the development of such corpora.

Apart from the above mentioned mono-lingual corpora some multi-lingual corpora are also freely available. Normally speeches made in multi-lingual fora are a good source of such corpora. One example is the *Hansard* which are the parliamentary proceedings in the Westminster style of governments. The Canadian Hansard[1] is bilingual in English and French. Another example from more recent history is the Europarl (Koehn 2002) which is a very comprehensive account of European Parliamentary proceedings updated regularly and is freely available in a refined form ready to be used for NLP tasks[2] in 11 European languages, or a subset of them.

Europarl is immensely useful as an un-annotated corpus and is well suited to unsupervised learning. Statistical approaches are well suited to process it and

[1] http://www.parl.gc.ca/housechamberbusiness/chambersittings.aspx
[2] http://www.statmt.org/europarl/

extract useful information, for instance, as to how the words are aligned between any two languages (Brown et al. 1993). Since such corpora are lacking in any annotated information, the algorithms have to themselves build statistical tables based on the frequency of words. GIZA++ (Och & Ney 2003) is now considered to be a standard tool in bilingual word alignment. It has been used in for the purposes of this research work as well.

Supervised or un-supervised, corpora based approaches can be used to automatically create resources such as WordNet, rather than doing it manually as for the original Princeton WordNet (PWN) (Fellbaum 1998), or EuroWordNet (Vossen 1998), which took years to be built.

Though, hand-crafted resources are generally more comprehensive and fine grained, such as WordNet, the automatically generated ones take less time and fewer number of human resources. But such processes are prone to errors, since normally statistical approaches are employed and they are far from perfect, inducing their own errors that may be multiplied over multiple languages. They also normally still require pre-processing to be done before any useful tool could be employed to extract any useful information.

1.2.3 Information Retrieval (IR)

Lancaster (1968) defines IR as:

An information retrieval system does not inform (i.e. change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request.

van Rijsbergen (1979) gives a detailed account of automatic IR in his book which is freely available on the internet[3].

IR pertains to searching documents of interest. The brute force approach

[3]http://www.dcs.gla.ac.uk/Keith/Preface.html

would be to look at each available document and see if it is of interest to the user by comparing it with the query given by the user. A more refined approach would be to *cluster* documents which are closer to each other based on a certain metric such as the Euclidean distance in the Vector Space Model (Salton et al. 1975). The closer the documents, the greater the chances that they belong to the same cluster. Thus clusters of documents are formed with a class assigned to each one of them. When a query is given by the user the whole set of documents in the cluster closest to the query is returned.

In this work we have treated each speech delivered in the European Parliament as a document and performed clustering on them with and without the sense tags in order to evaluate the multilingual synsets that were generated. This step essentially is IR.

1.2.4 Multilingual NLP and IR

While monolingual NLP is generally concerned with using NLP techniques from the perspective of one language at a time, for instance, finding English synonyms using the Distributional Similarity measure (van der Plas & Tiedemann 2006), where there is no need to compare the context in one language to the context in another. For such tasks a corpus in one language suffices.

Yet, the full potential of NLP techniques can not be realized unless they are set in the environment where multiple languages are considered. Using more than one language may increase the overhead of dealing with more than one language at a time but could be very useful for certain tasks, for instance, trying to reduce polysemous ambiguity in the language of interest. A polysemous word may be translated into different word forms in another language, indicating different senses in which it might be used. That helps in narrowing down the number of senses in one of the languages, in which that word could be used in that particular

context. It is essentially the WSD task, but rather than relying on one language we take cues from other languages.

Multilingual corpora are also useful in some other tasks, such as Machine Translation, where models can be trained to learn word/phrase pairs in the languages of interest. The larger the size of the corpora the greater the accuracy of such alignment. They can also be useful for creating multilingual lexicons, a task hitherto done by teams of human experts which is quite time consuming and is also costly. Automating such tasks can save a lot human effort and can also be less costly.

IR can also be done in the multilingual context. One of the earliest experiments were conducted by Salton (Salton 1970) for English and German. Other methods that have been used include similarity thesauri (Schäuble 1997) (Sheridan & Ballerini 1996) and Latent Semantic Indexing (LSI) (Landauer & Littman 1990) have also been used for the said purpose. Comparable Corpora have also been used for Multilingual IR (Talvensaari et al. 2007), and (Braschler & Schäuble 1998). McEnery (2003) defines comparable corpus as the one that is collected using "the *same proportions* of the texts of the *same genres* in the *same domains* in a range of *different languages* in the *same sampling period*.".

1.3 Multilingual Resources

The need of international organizations, such as the United Nations (UN)[4] or the European Union (EU)[5] together with the advancements in Computer technology made it relatively easier to exploit them.

Different resources vary widely on what they offer in terms of the topics and languages. The UN defines six languages viz. Arabic, Chinese (Mandarin),

[4]http://www.un.org/en/
[5]http://europa.eu/

English, French, Russian and Spanish (Castilian) as official languages. EU on the other hand defines 23 different languages as official. Being official grants a special status to that language and all publications are done in all the official languages. These provide an immense resource of multilingual parallel corpora, where the translations of the documents are available. Southeast European Times (SETimes)[6] is the manifestation of parallel corpora available online consisting of the news items.

Apart from the above mentioned corpora Wikipedia[7] is a rich resource available freely online in 282 languages. It makes it easy for the users to edit it and contribute articles on any conceivable topic under the sun. The articles may or may not be translated in different languages depending on their interest and are not translations of each other. Thus, they are not parallel in nature, yet expanding on the same topic but in different context and perspective.

1.4 Problem Statements

The thesis has three main aims, two of which (building multilingual lexicons and generating multilingual proto-synsets) are independent while the third (morphological analysis of the proto-synsets) is dependent on the second.

1.4.1 Building Multilingual Lexicons

Premise A: Wikipedia is a freely available online resource which can be seen as a multilingual comparable corpus.

Premise B: The title(s) of each Wikipedia article across languages are faithful translations of the same concept.

[6] http://www.setimes.com/cocoon/setimes/xhtml/en_GB/homepage/

[7] http://www.wikipedia.org/

Premise C: A crawler can be used to mine Wikipedia and extract the titles of the articles on the same topic in languages of interest.

Conjecture A: The fact that article titles are translations of each other across languages, can be used to generate multilingual lexicons.

Conjecture B: The Wikipedia categories can be used to select terms from a particular domain.

Our first aim is expressed in the following research questions:

Is it possible to use an online freely available multilingual resource, such as Wikipedia, to create a multilingual lexicon? Can it be done to create a general as well as domain-specific dictionaries?

1.4.2 Creating and using Multilingual Synsets

Premise A: Monolingual, PWN, (Miller et al. 1990) and Multilingual, EuroWordNet, (Vossen 1998) WordNets already exits. Where a WordNet is a lexical semantic resource in which the semantics of a word is defined by the list of all words sharing that meaning. Such lists are referred to as synsets. The original PWN is for English, while EuroWordNet is for various languages of the European Union, which are linked to the PWN through Inter-Lingual Indexes.

Premise B: Standard word alignment tools, such as GIZA++ (Och & Ney 2003), exist that take in a bilingual parallel corpus, and word aligns it, mapping a word in one language onto one or more words representing its probable translations in the other, using the contexts of the two words in their respective texts.

Conjecture 1: Word-aligning a multilingual parallel corpus would produce a set of words/phrases containing synonymous expressions for all languages. These can be used as a kind of multilingual synsets annotating the words and phrases in the corpus from which they have been derived with a lexical semantic tag. The result can be used in unsupervised approaches to NLP/IR as no additional human

annotation of the parallel corpus is required.

Conjecture 2: The notion of multilingual synsets can be employed with the ultimate aim of being able to disambiguate between the meanings of words and phrases in any given language represented in the corpus.

That raises the following research questions:

Can the existing word alignment tools, such as GIZA++ (Och & Ney 2003) be used to word align parallel corpora across different languages? Can the word alignments, thus created, be used to merge the aligned words and create phrases? Can the sum of multilingual phrases be used as multilingual synsets to disambiguate the word meanings in the corpus from which they have been derived? Can we use them in general purpose tools beyond the parallel corpus from which they have been derived? Can meaningful evaluation be done in the absence of any gold standard?

1.4.3 Morphological Analysis of Multilingual Synsets

Premise A: There are a number of approaches for the unsupervised learning of word morphology that can be used to map word forms onto their base forms (i.e., lexical entries) (Snyder & Barzilay 2008), (Kazakov & Manandhar 2001), and (Goldsmith 2001).

Premise B: Any approach using multilingual synsets would benefit from a tool mapping word forms onto lexical entries in order to avoid spurious variations among these synsets.

Premise C: The multilingual synsets provide additional context to the word forms for any given language that could be used with benefit when applying unsupervised learning of word morphology.

Conjecture 1: We can use this data to produce a word morphology model in an unsupervised way.

Conjecture 2: The result can be used to improve the quality of the multilingual synsets.

That raises the following research questions:

Can we test Conjecture 1 and evaluate it by comparing against a gold standard (van den Bosch et al. 1996) or with other unsupervised techniques?

1.4.4 Evaluation and its Challenges

Creating the multilingual lexicons and a WordNet like resource posed their own challenges but their evaluation proved to be a really hard nut to crack. In the absence of multilingual gold standard corpora, evaluating our own algorithms was tricky.

The way the multilingual lexicons could be evaluated were through building the languages' family tree structure and comparing it with the real world family tree structures for the languages considered. The sparsity of the lexicon provided clues to which languages shared common set of articles and were thus considered to be closer since the people who had written in those languages seemed to be interested in similar topics probably due to the fact that they belonged to the same geographical region or shared cultural, political or religious leanings.

Evaluation of multilingual synsets proved to be even harder in the absence of any gold standard. Thus we assumed the original English corpus to be disambiguated, as the gold standard and the results of disambiguation were evaluated against it after clustering both the original and the disambiguated corpus.

1.5 Thesis Outline

The thesis is structured as follows. Chapter 2 gives a literature review on using Wikipedia to create multilingual resources, aligning parallel corpora, morphol-

ogy, WSD, evaluation, and IR. Chapter 3 gives a detailed explanation of how the multilingual lexicons were generated from Wikipedia. Chapter 4 expands on the extraction of multilingual proto-synsets from the aligned corpora, and their evaluation. Chapter 5 explains how the multilingual proto-synsets were used to do morphological analysis of the languages at hand. Chapter 6 discusses the conclusions and future work.

1.6 Note on Terminology

One aim of the thesis is to create multilingual proto-synsets that could become the basis of an automatically created fully-fledged multilingual WordNet with all the nuances of semantic relationships as defined in the PWN, such as hyponymy, herpnymy, synonymy, and meronymy. But before we embark upon such an endeavor it would be appropriate to define what a multilingual proto-synset really is.

A multilingual proto-synset, as the name implies, would be multilingual in nature. The term synset, as originally used for the PWN is a set of synonyms. But in our case we are putting together words and phrases in different languages, their alignments put together constitute the synset. For instance, ⟨resumption of, wiederaufnahme, reprise de, $\epsilon\pi\alpha\nu\acute{\alpha}\lambda\eta\psi\eta\ \tau\eta\varsigma$⟩ is one such multilingual synset. We add the notion of *proto* to convey the message that these synsets are still in their raw form and will need a lot of processing to make them into refined set of synsets, for instance, merging synsets that are syntactic inflections of each other. The synset describing the concept *dog* and the *dogs* are basically the same and are just inflectional variation of each other.

CHAPTER 2

Literature Review

This work falls under the larger umbrella of Artificial Intelligence (AI) and covers a variety of sub-domains, such as Natural Language Processing (NLP) and Information Retrieval (IR), Machine Learning (ML), Computational Lexicography, and using search to build lexicons and using statistical methods combined with a deterministic algorithm to build a multilingual resource.

Since, it covers all these sub-domains of AI, it would be appropriate to shed light on what they are and what contributions have been made by other people relevant to our research.

The rest of the chapter is organized as follows: Section 2.1 discusses various mono- and multilingual resources available that could be used to carry out NLP/IR tasks; Section 2.2 discusses various tasks that fall under the category of NLP/IR, such as Word Sense Disambiguation (WSD), Morphology, and IR itself; Section 2.3 expands upon ML approaches relevant to our work; and fi-

nally, Section 2.4 discusses various approaches adopted to build resources from corpora.

2.1 Resources

Over the years many online and offline resources have been built that can be used by applying AI or ML techniques to either build new resources or to carry out other useful tasks, such as WSD and morphological analysis.

A lot of human effort has gone into building these resources, with or without the modern technology available in the form of micro-processors. The available resources are both mono- and multi-lingual in nature, and are either machine readable or can be converted into one.

We have used a few of these resources in our research. The rest of this section covers a few of the available resources.

2.1.1 Wikipedia

Beginning in 2001, Wikipedia[1] has emerged as one of the largest online sources of multilingual information, "attracting 400 million unique visitors monthly as of March 2011 according to ComScore"[2]. "There are more than 91,000 active contributors working on more than 17,000,000 articles in more than 270 languages." (September 20, 2011) [3] With a very high flexibility for editing, virtually anyone can add pages in Wikipedia.

It is a freely available multilingual encyclopaedia which can be edited by anyone with access to the internet. To ensure the veracity of information avail-

[1] http://www.wikipedia.org/
[2] http://stats.wikimedia.org/reportcard/
[3] http://en.wikipedia.org/wiki/Wikipedia:About

able there, administrators[4] are promoted through consensus among the Wikipedia community. One of their tasks is to ensure that articles are properly referenced. It caters to multitude of writing systems and covers every conceivable topic in the world that attracts enough attention that people want to write about it. The objectivity and quality of articles on Wikipedia may be brought to scrutiny but Giles (2005) showed that their quality is comparable to Encyclopedia Brittanica.

It is based on *wiki* which is a collaborative tool that allows users to edit online. Ward Cunningham[5] the first prototype in 1995. Wikipedia is based on it and the word "Wikipedia" is a portmanteau of the words *wiki* and *encyclopedia*[6].

2.1.2 Parallel Corpora

Parallel corpora are translations of a text in different languages. The languages covered in any particular corpus is dependent on the purpose for which it is created. For instance, the Canadian parliamentary proceedings, known as the Canadian Hansard[7], covers English and French.

The quality of parallel corpora is dependent on the quality of translators. They may or may not be tagged. Tags can be syntactic or semantic in nature. An example of a lexical semantic resource is Princeton WordNet (PWN) (Miller et al. 1990), which is discussed later in section 2.1.3. Tagged corpora can be used both for supervised and unsupervised learning approaches. While the untagged corpora can only be used for unsupervised learning approaches unless they are tagged first.

Parallel corpora can be used to carry out certain NLP/IR tasks, such as Word Alignment (Och & Ney 2000), and Word Sense Disambiguation (WSD) (Tufis

[4]http://en.wikipedia.org/wiki/Wikipedia:Administrators
[5]http://en.wikipedia.org/wiki/Ward_Cunningham
[6]http://en.wikipedia.org/wiki/Wikipedia:About
[7]http://www.parl.gc.ca/ParlBusiness.aspx?Language=E

et al. (2004), Ng et al. (2007)).

There are a number of parallel corpora available for NLP/IR tasks. Examples include Europarl (Koehn 2002) and Canadian Hansard [8] etc.

Europarl

Europarl (Koehn 2002) provides the translated proceedings of the the European parliament freely available for carrying out NLP/IR tasks. It is currently available in 11 different languages[9], covering a variety of language families and writing styles. The aim of the project is to create sentence aligned corpora. Earlier versions were less refined and hence needed a lot of pre-processing. Later versions are relatively easier to process with less pre-processing required and are already sentence aligned.

OPUS: the open parallel corpus

OPUS (Tiedemann 2004) is a project that aims to provide a wide range of parallel corpora to the research community. The corpora are taken from several online resources, sentence aligned and converted into a uniform XML format. It covers more than a 100 languages and is thus rich in linguistic diversity. It uses a number of sources to build these parallel lexical resources, such as Europarl (Koehn 2002), European Central Bank (ECB) corpus[10], and Southeast European Times[11].

Canadian Hansard

Canadian Hansard is the record of Canadian Parliamentary Proceedings[12]. They are available in both English and French and is tagged with information related to the speakers and the language used by them.

[8] http://www.parl.gc.ca/housechamberbusiness/chambersittings.aspx
[9] http://www.statmt.org/europarl/
[10] http://www.ecb.int/pub/html/index.en.html
[11] http://www.setimes.com/
[12] http://www.parl.gc.ca/ParlBusiness.aspx?Language=E

2.1.3 WordNet: A Lexical Semantic Resource

The notion of synset, or set of synonyms, comes from a project at Princeton, guided by George Miller (Miller et al. (1990), Fellbaum (1998)). He started work on a lexical database, as opposed to an alphabetical dictionary, known as PWN.

Conventional dictionaries put everything in alphabetical order, which seems to be the most natural way of storing such information. Yet it has proved to be highly in-efficient in terms of finding synonyms, antonyms and other such semantic information, which might be of great use to the user.

PWN divided the lexicon into three different categories: nouns, verbs, adjectives, and adverbs. It provides a mapping between word forms and word meanings by building a lexical matrix, with word forms being the headings of the columns and word meanings being the headings of the rows. Any entry in this matrix builds a relationship between the form and the meaning. If there are two entries in a row, the words are synonymous, and if there are two entries in the same column, the words are polysemous. Where synonyms are words with the same meaning and polysemous are the words with multiple meanings. PWN defines other semantic relationships as well, such as antonymy, hyponymy, and meronymy.

Synonymy

Synonymy defines a relation between any two word forms which share a common meaning. Thus, two words are synonymous if substituting one for the other in a sentence, does not change its truth value. This is a very strict definition of synonym, and such synonyms are rare, if they exist at all. A more weaker version of the definition relates to the context in which the word forms are used. So two word forms are synonymous if substituting one for the other in a linguistic context does not change its truth value. For instance, in the *carpentary* context,

substituting *plank* for *wood* will not change its truth value, hence they are synonyms. Such words can be combined in the form of sets, known as the *synsets*. Thus the synset in this case will be {plank, board}.

Antonymy

Antonymy is a relation between words that carry meaning opposite to each other. It also has to do with word forms and not just the meaning. It would be a mistake to assume that *not-x* would be an antonym of the word x. For instance, *rich* and *poor* are antonyms but to assume that *not-rich* is antonym for *rich* would be a folly, since not being rich does not necessarily mean being poor.

Similarly *rise* and *fall* are antonyms, and so are *ascend* and *descend*. But *rise* and *descend* are not. Thus, word form is also important, and not just meaning, in deciding whether two words are antonyms of each other or not.

Hyponymy/Hypernymy

It defines the *IS_A* relation between word meanings. In other words it defines the subordinate/superordinate relationships, where hyponymy corresponds to subordination and hypernymy corresponds to superordination. For instance a tiger IS_A cat or the hypernym for tiger is a cat and the hyponym of a cat is a tiger. It helps building the inheritance systems, which may be used for IR.

Meronymy/Holonymy

It defines the HAS_A relation between word meanings. For instance a car has_a tyre, which is holonymy relation. Meronymy is the opposite relation, e.g., that tyre is a part of a car.

Synset is the basic unit of information that PWN deals with. Figure 2.1 gives a snapshot of the synset in WordNet[13]. Words at the same level form a synset, which in this case is for the concept *car*. In the WordNet version 3.0, there are a total of 117,659 synsets. Every synset in WordNet has a unique ID and is also

[13]http://wordnetweb.princeton.edu/perl/webwn?s=car&sub=Search+WordNet

assigned a POS tag. For instance, for the word *Actifed*, the ID is 02680086 and the POS tag is *n*, which means that it is a noun.

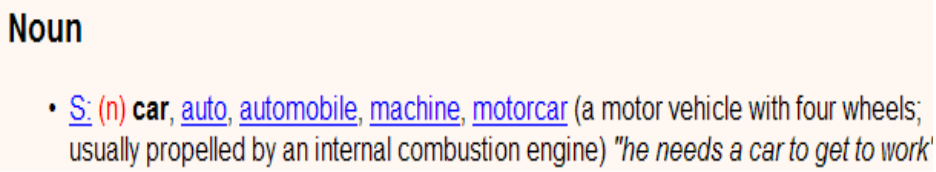


Figure 2.1: A snapshot of PWN showing a synset.

2.2 NLP and IR

NLP and IR are by now established areas of research in the realm of Computer Science. NLP is the branch of Computer Science (CS) which deals with interaction between computers and natural languages used by humans. It falls under the broader category of Artificial Intelligence (AI). NLP has both the computational and the linguistics aspects, since the knowledge of both is required in order to build effective NLP systems.

IR is the science of searching relevant documents, the information within documents, or meta data related to documents based on a search query given the by the user.

NLP has a number of subtasks, of which Word Sense Disambiguation (WSD) and Morphological Analysis are discussed here.

2.2.1 Word Sense Disambiguation

Ambiguity is natural in any natural language. The task of disambiguation refers to the process by which the software narrows down the meaning of a target word. It can be performed in the mono-lingual or multi-lingual context, based on the availability of resources.

A common example of ambiguity is the word *bank* has two common interpretations: the bank of a river and the financial institution. The task of disambiguation is to estimate in which sense it is used in a given context. It is known as *polysemy* in linguistics.

Chomsky (1965) gives many examples of ambiguities that exist in English. For instance the sentence "flying planes can be dangerous" can be interpreted in two different ways: "flying planes are dangerous"; or "flying planes is dangerous".

Another example quoted by Chomsky is "I had a book stolen". It can be interpreted in three different ways: "someone stole my book"; "I had someone steal a book"; or "I had almost succeeded in stealing a book".

Polysemy occurs quite often in any natural language. For English, the Princeton WordNet (PWN) (Miller et al. 1990) gives an average polysemy of 2.79 for nouns, 3.57 for verbs, 2.71 for adjectives, and 2.50 for adverbs[14].

Polysemy is just one kind of ambiguity inherent in a natural language. Prepositional phrase (PP) attachment ambiguity and ambiguity in tagging are also examples of ambiguities in a language.

PP attachment ambiguity refers to the problem of deciding whether the PP attaches with a noun or a verb. More light is shed on it in section 2.2.1.3.

Ambiguity in *tagging* refers to what part of speech (POS) tag should be assigned to a word. For instance the word *butter* could either be a noun or a verb. Using a word as a verb rather than as a noun might totally change the meaning of the word and thus could be viewed as a WSD problem. In order to disambiguate such ambiguities, nearby structural cues might help such as the use of a determiner before the word.

Both supervised and unsupervised learning approaches could be used for re-

[14]http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html

solving ambiguity in the text. The difference between the two is that in *Supervised Learning* we know the classification of each example but in *Unsupervised Learning* the classification of training data is unknown in advance. Thus unsupervised learning can be seen as clustering while supervised learning can be seen as a classification task, or as a curve-fitting task.

2.2.1.1 Supervised Disambiguation

In supervised disambiguation the task is to train the algorithm based on labeled examples and then to generalize it in order to classify the hitherto unseen examples. It can not work without the availability of annotated data, which is expensive to create. An example of this approach is the Bayes Classifier.

Bayes Classifier

The Bayesian approach looks at words surrounding the target word in the text, making its context. Words in a context carry useful information about the target word and help in narrowing down its sense. The Bayes classifier uses the *Bayes decision rule* to decide the class of the target word. Its aim is to minimize error in classification (Duda & Hart 1973). The rule can be described mathematically as in equation 2.1.

$$Decide\ s'\ if\ P(s'|c)\ >\ P(s_k|c)\ for\ s_k \neq s' \tag{2.1}$$

Here s' and s_k are two different senses of the target word, and c is the set of words in its immediate context.

The probability $P(s_k|c)$ is usually not known but can be estimated using the Bayes rule as given in equation 2.2.

$$P(s_k|c) = \frac{P(c|s_k)}{P(c)}P(s_k) \tag{2.2}$$

In equation 2.2 $P(s_k)$ is the *prior probability* of sense s_k, without any information about the context in which the word has occurred. It is updated with a factor that incorporates the context into its calculations. $P(s_k|c)$ is the *posterior probability*. However, $P(c)$ is independent of the sense and does not help in disambiguating the word sense and hence can be removed from the equation to give: $P(s_k|c) \propto P(c|s_k)P(s_k)$. The classification process is then reduced to maximizing the posterior probability (Equation 2.3).

$$\begin{aligned} s' &\propto \arg\max_{s_k} P(s_k|c) \\ &\propto \arg\max_{s_k} \frac{P(c|s_k)}{P(c)} P(s_k) \\ &\propto \arg\max_{s_k} P(c|s_k)P(s_k) \\ &\propto \arg\max_{s_k} [\log P(c|s_k) + \log P(s_k)] \end{aligned} \tag{2.3}$$

Gale et al. (1992) describe a particular kind of Bayes classifier, known as the *Naive Bayes Classifier*.

Naive Bayes Classifier

Naive Bayes Classifier essentially sees the words in a text as conditionally independent. It removes the structure from the text, and is referred to as the *bag of words* model. In the bag of words model, words are not dependent on each other so that their order does not matter any more. For instance, the word *professor* is more likely to occur in the context of a *university* and not a trade union. But that relationship is lost in the bag of words model. Mathematically the Naive Bayes assumption can be expressed as in equation 2.4.

$$P(c|s_k) = \prod_{v_j \in c} P(v_j|s_k) \tag{2.4}$$

Equation 2.5 redefines the decision rule in equation 2.1 in the light of bag of words model.

$$\text{Decide } s' \text{ if } s' = argmax_{s_k}[\log P(s_k) + \sum_{v_j \in c} \log P(v_j|s_k)] \qquad (2.5)$$

Dictionary-Based Disambiguation

In order to disambiguate a word in one language we can take clues from its translation in another (Dagan et al. (1991), Dagan & Itai (1994)). The reason being that two different translations may be used in another language for two different senses of a word in the original language.

Manning & Schütze (1999) gives an example of the English word *interest*, which has two different meanings and are translated as two different word forms in German. One is *Beteiligung*, meaning the *legal share*, and the other is *Interesse*, meaning attention or concern. The first meaning can be used as "somebody has more than 50% interest in the company". The second meaning can be used as "she has shown interest in Computer Science".

We look for occurrences of the word *interest* in the English text and see if it is used in a particular sense in the translated contexts in German. If that is always the case, then our job is done and we assign that particular sense to the target word.

If that is not the case then the context of the target word needs to be looked more carefully for other clues. For instance, let's suppose the word *interest* is used in the sense of *to show interest*. The translation of show in German is *zeigen*, and it will occur in the context of *Interesse*, since legal shares are not shown. Thus, we can conclude that in the phrase *to show interest*, the word interest is used in the second sense.

The goal is to disambiguate the target word in a particular context. Let us define a relationship $R(w, v)$ as the 'object-of' relationship, or we can say word

w is an object of word v. For the case of *interest*, one such relationship is $R(interest, show)$. Given the two translations of the word *interest* in German, and one translation of the word *show*, we count the number of times *beteiligung* occurs as the object of *zeigen*, and also the number of times *Interesse* occurs as the object of *zeigen*. Or in other words we take counts of R(beteiligung, zeigen) and R(Interesse, zeigen). The count of R(Interesse, zeigen) would be higher so we can conclude that it is used in the second sense of the word.

Let $R(w, v)$ be the 'is-object-of' relation, S be the second-language corpus, $T(s_k)$ be the set of possible translations of sense s_k, and $T(v)$ be the set of possible translations of v. Then,

comment: Given: a context c in which w occurs in relation $R(w, v)$

for all senses s_k of w **do**

$$\text{score}(s_k) = |\{c' \in S | \exists w' \in T(s_k), v' \in T(v) : R(w', v') \in c'\}|$$

end

choose $s' = \text{argmax}_{s_k} \ \text{score}(s_k)$

In some cases using dictionaries for disambiguation may not make sense, specially for closely related languages. For instance, the word *interest* in English and its French equivalent *intérêt* are ambiguous in both languages in more or less the same ways. In such cases bilingual dictionaries might not be of much help in resolving ambiguity. It makes sense to make use of dictionaries when they make sense and to make use of other alternatives when it does not (Gale et al. 1992).

Information Retrieval (IR) Approach to Sense Disambiguation

Gale et al. (1992) treated contexts just as documents are treated in a probabilistic information retrieval (IR) model (Salton 1989), (van Rijsbergen 1979). Let the tokens be represented by t, the relevant and irrelevant documents by r and r' respectively, and the two senses by s_1 and s_2 respectively, then the IR model sorts documents by:

$$\text{score}(d) \;=\; \prod_{t \in d} \frac{P(t|r)}{P(t|r')} \tag{2.6}$$

for WSD contexts c would be sorted by:

$$\text{score}(c) \;=\; \prod_{t \in c} \frac{P(t|s_1)}{P(t|s_2)} \tag{2.7}$$

where $P(t|s_i)$ denotes the estimate of the probability that whether the token appears in the context of s_1 or s_2.

They defined the context as a window of 50 words to the left and also to the right of the ambiguous word. Other studies have chosen to keep the context to the words that are quite nearby. An approach that finds its basis on Kaplan's (Kaplan 1950) observation "a context consisting of one or two words has an effectiveness not markedly different from that of the whole sentence." Yet they figured that in the Hansards (official records of the Canadian Parliament), context was found to be relevant to noun disambiguation up to ten thousand words away. Yet information at some remote distance from the ambiguous word may just duplicate the information available at some nearer point. They also showed that not many examples were needed for training to achieve good accuracy. In their experiments three examples gave 75% accuracy and ten gave 80%. Thus the marginal utility of extra examples was not very high. Thus useful systems could be built for senses not occurring too many times in the corpus.

2.2.1.2 Unsupervised Disambiguation

Unsupervised methods, as opposed to supervised methods do not require annotated corpus to carry out any useful tasks. They use contextual information to describe the properties of the target words, phrase, sentences, and documents.

Sentence Level Translation

Brown et al. (1990) chose as the translation of the French sentence F that sentence in English E for which $P(E|F)$ is greatest. It is defined by the Bayes'rule as:

$$P(E|F) = \frac{P(E)P(F|E)}{P(F)} \tag{2.8}$$

Since the denominator is independent of E, the equation reduces to maximizing $P(E)P(F|E)$. The first factor corresponds to the statistical characterization of the English language, and the latter corresponds to the statistical characterization of the process of translation from English to French. Different models can be employed to estimate the values of the probabilities.

Translation Model

The approach used the concept of *alignment* in which each English word, independent of other words, produced 0 or more words in French. If A denotes a typical alignment then the probability of translation from English to French can be expressed as a sum over all possible alignments.

$$P(F|E) = \sum_{A} P(F, A|E) \tag{2.9}$$

The number of possible alignments increase rapidly with the size of the sentences in the two languages. Yet not all of them contribute equally to the sum. The one that contributes the most is called the *Viterbi Alignment* between the two languages. The words thus aligned are known as connections. They obtained over 12 million connections from the Canadian Hansard (Brown et al. 1990).

They defined as $p(e, f)$ as the probability that the connection chosen at random from the set of connections would connect the English word e to the French word f. It could be used to compute the mutual information between a French

word and its English mate in a connection. Mutual information estimates give us the relationship between the two variables and gives us the information that each one of them shares. It gives us a measure of how much uncertainty is removed about one if we have information about the other.

Brown, Pietra, Pietra & Mercer (Brown et al.) described a method for labeling a word with the sense depending on the context in which it appears, so as to increase the mutual information between the words in a connection. In the French sentence *Je vais prendre ma propre décision*, the word *prendre* should be translated as *make* since its object is *décision*. If *décision* is replaced by *voiture*, meaning car in English, it should be translated as *take* to yield *I will take my own car*. Thus the sense assigned to *prendre* depends on the first noun to the right, which they called the *informant* for *prendre*.

They defined seven informants for French: the word to the left; the word to the right; the first noun to the left; the first noun to the right; the first verb to the left; the first verb to the right; and the tense of either the current word, if it is a verb, or of the first verb to the left of the current word. For English they only considered the previous two words.

For the French word *prendre*, the noun to the right yielded the most information, 0.381 bits, about the English translation of the word. The nouns which appear most frequently on the right of *prendre* were identified, with the probability of occurrence greater than one part in fifty. They were divided into two groups depending on the sense they translate the French word *prendre* into. The word is assigned the sense depending on the group to which the word on its right belongs to. They discovered that if the noun on the right of *prendre* was *décision*, the probability of its translation as *to make* was 3 times higher than its translation as *to take*.

Yarowsky (1992) used Roget's Categories to disambiguate words in English.

Roget's categories tend to correspond to sense distinctions. Thus finding the Roget category for a word is akin to discriminating between different senses of the word. The most probable category given the context was selected. There are a total of 1,043 such categories. Each word may belong to one or more categories, identifying different senses in which it could be used.

For each category in the Roget Categories, they first collected contexts representative of the category. The goal of this step is to collect a set of words that are typically found in the context of category. In order to achieve it they collected contexts of 100 surrounding words for each occurrence of each member of the category in the corpus (June 1991 electronic version of Grolier's Encyclopedia).

Then in each collective context, they extracted the words that would give significantly more information about the meaning of the target word. They called them the salient words. In order to find the salient words they used a mutual information like estimate: $\frac{P(w|C)}{P(w)}$, where $P(w/C)$ is the probability of a word that it appears in the context of the category, and $P(w)$ is its overall probability in the whole of the text. The higher the estimate, the higher the probability that the word is a salient word. Log of salience gives the weight of the word.

Occurrence of a salient word in the context of the target word indicates that the target word belongs to the category for which the salient word is related. Presence of several such words provides further evidence of it.

In order to determine the category of the target word they summed the weights of the salient words in a context and chose the category with the highest sum, using equation 2.10.

$$\arg\max_{C} \sum_{w \text{ in } context} \log\left(\frac{P(w|C) \text{ x } P(C)}{P(w)}\right) \tag{2.10}$$

The algorithm was applied to 12 words: star, mole, galley, cone, bass, bow,

taste, interest, issue, duty, sentence and slug. Accuracy ranged from 100% for some, such as the *Securities* sense of the word *stock* to a low of 25% for the *ornamentation* sense of *ribbon*. For most of the words accuracy remained fairly high.

Properties of a Word in a Document and in its Context

Yarowsky (1995) used the notions of "one sense per discourse" and "one sense per collocation" to do unsupervised learning for disambiguation. Their corpus contained 460 million words containing news articles, scientific abstracts, spoken transcripts, and novels. The notions stated above are defined below as:

One sense per discourse:

The sense of a target word is highly consistent within any given document (Yarowsky 1995).

One sense per collocation:

Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship (Yarowsky 1995).

They first extracted all examples of a given target word (*plant* in this case) from the corpus. They then put together these extracted examples to form the untagged training set. For each sense of the target word, they identified a few *seed collocations*. For instance, they chose the words *life* and *manufacturing* as seed collocations for the senses of the word *plant* relating to trees and industrial plants. They then collected all the training examples containing the seed collocations and tagged them with the appropriate senses. That yielded 82 examples (1%) of the sense of plant being life form, and 106 examples (1%) of the sense of plant being an industrial unit. The rest of the 98% were residual training examples, making a total of 7,350 examples.

The seed collocations were used to identify other collocations for each sense

using the decision list algorithm (Yarowsky 1994). Decision list gives a list of collocations with the corresponding sense ordered by the log-likelihood ratio $Log\left(\frac{P(Sense_A|Collocation_i)}{P(Sense_B|Collocation_i)}\right)$. A new instance would be compared against the decision list and the first collocate would be identified matching it, from the top of the list. The corresponding sense would be assigned to that new instance. A collocate, such as *life* might appear in different collocations, for instance as *life (within ±2-10 words)* or *plant life*.

After applying the "one sense per collocation" principle, they used the "one sense per discourse" principle to tag the training examples. If many instances of the target word were tagged with a particular sense in a discourse, the rest of the instances were also assigned the same sense. The principle can also be used to correct certain misclassifications. If a training instance is classified as something else originally, its sense tag can be changed if the rest of the examples, or most of them, in a discourse share a particular sense.

By applying both the constraints, the original seed sets keep expanding with new examples being added while the residual keeps shrinking till the algorithm converges to a stable residual set.

They showed that their algorithm gave similar performance as the supervised algorithm (decision list algorithm applied to the same data without using any discourse information) given identical training contexts (95.5% vs. 96.1%).

2.2.1.3 PP Attachment Ambiguity

Prepositional Phrase (PP) attachment is an attachment ambiguity problem that has intrigued both linguists and computational linguists for decades. It basically pertains to deciding whether the PP attaches to the noun or the verb. An example could be "He saw a man with the telescope." It is difficult to decide in this case whether the telescope was carried by the person watching the other one, or by

the one being watched.

Collins & Brooks (1995) used the backed-off model to ascertain the probabilities of pp attachment to the noun or to the verb. It uses the 4-tuples comprising the four head words, denoted by ⟨verb,head of noun phrase1,preposition,head of noun phrase2⟩. An attachment decision value was defined, with 1 for noun attachment and 0 for verb attachment, and was denoted by A, to create the quintuples. Since the attachment value of A was dependent on the four head words, conditional probabilities were used. It was assumed that the default was noun attachment ($A = 1$), thus the actual probability of A given the four head words was estimated using:

$$\hat{p}(1|v, n1, p, n2) \tag{2.11}$$

where v is the verb head, $n1$ is the head of the noun phrase 1, p is the prepositional phrase, and $n2$ is the head of the noun phrase 2. The decision could be made based on the test:

$$\hat{p}(1|v, n1, p, n2) \geq 0.5 \tag{2.12}$$

Thus if the above test is true then attachment is assumed to be noun, otherwise verb. The probability estimates were based on frequency counts. The maximum likelihood estimated (MLE) method was used for estimation, which gives the following:

$$\hat{p}(1|v, n1, p, n2) = \frac{f(1, v, n1, p, n2)}{f(v, n1, p, n2)} \tag{2.13}$$

Where f denotes the frequency with which a tuple occurs in the training data. Thus, $f(1, v, n1, p, n2)$ is the frequency with which the tuple *(1,v,n1,p,n2)* occurs with a noun attachment. The denominator does not contain any information re-

garding noun or verb attachment, and thus is just the count of number of times that tuple occurs with any attachment in the training data. Thus if the above ratio between the two frequencies is greater than or equal to 0.5, then it is noun attachment, else it is verb attachment.

The backed-off model (Katz 1987) is based on predicting the probability of the word n, given the $n-1$ preceding words. But since these estimates are based on frequencies of n-grams, the higher the order of these n-grams, lower the frequency. Thus it is quite possible that for any order n, the frequency might be less than a certain threshold, which would give inaccurate estimates. Due to this problem in the backed-off model the order of n-grams is reduced in each iteration, which increases the chances of frequency of such n-grams be greater than the threshold. It is backed off till sufficiently accurate estimates can be made.

Using the backed-off model for PP attachment prediction (Collins & Brooks 1995), the tuples are reduced in size first from four head words to three, and then to two, given that those tuples would have a preposition in them. It yields frequency counts of three different 4-tuples: $f(1, v, n1, p)$, $f(1, v, p, n2)$, and $f(1, n1, p, n2)$. Further reduction would yield three different frequency counts of 3-tuples: $f(1, v, p)$, $f(1, n1, p)$, and $f(1, p, n2)$. Thus first it would try to estimate the probability of PP-attachment for the case when we are taking frequency counts of all the four head words. If it is not greater than 0, then it backs off, reducing the order of n-grams to three, and tries to estimate the probability if the combined frequency of all the 4-tuples are greater than 0. In case it fails, it backs off to the last case where the order of n-grams is reduced to 2, and the estimates are made based on the combined frequencies of the 3-tuples. If all the above cases fail then it gives the default pp-attachment of noun to the phrase.

The study proved more successful than other studies hitherto done on the

Wall Street Journal corpus, yielding an accuracy of 84.5%, which was very near to the human performance of 88%, using four head words alone. One of the important discovery of this study was that "ignoring events which occur less than 5 times in training data reduces performance to 81.6%".

Kazakov et al. (2006) used Inductive Logic Programming (ILP) to learn rules that would help in resolving PP attachment ambiguity. They used the ILP tools Progol and CLOG. They used WordNet to assign all possible semantic tags to label the ⟨Verb, Noun, Prep, Noun⟩ 4-tuples. The 4-tuples were also labeled with the attachment class, 'N' for noun attachment, and 'V' for verb attachment.

Both the tools learned attachment rules. The rules were learnt separately by Progol while CLOG learned them together. Progol learned pure prolog programs with no ordering between them while CLOG learned the rules with an ordering from the most specific to the most general. Only the most specific applicable rule would apply. CLOG rules are intended to learn the most likely rules that explained an attachment given a particular context. CLOG would only learn one rule per example, but Progol might learn many rules for each example.

They adopted both greedy and non-greedy approaches for learning with progol. In the greedy approach once a clause was learnt, all the positive examples covered by it were removed. That speeded up the process by removing a few positive examples at each step but that made evaluation more difficult since knowing how many positive examples are covered by a clause is important. In the non-greedy approach Progol takes one positive example at a time and finds the best clause that covers it. The process of inducing a clause is independent for different examples and hence it can be parallelized. It does not remove any positive examples once a clause has been covered.

Both Progol and CLOG learned a number of rules. For the non-greedy approach Progol learned 1,542 rules for noun-attachment and 1,996 rules for verb

attachment. For the greedy approach it learned 257 rules for noun attachment and 541 rules for verb attachment. CLOG on the other hand learned 338 rules in total.

It could not improve on the Naive Bayesian approach since the original data had a lot of ambiguity.

PP Attachment and WSD

The context in which the word appears plays a great role in disambiguating any ambiguous words. That is where PP Attachment disambiguation comes into the picture. By changing the *noun* or *verb* attachment, the whole meaning of the sentence might change and it might also change the sense in which a particular ambiguous word has been used.

Consider an example of a sentence:

I saw a star in the park with a telescope.

It has different semantic interpretations:

1) *I saw [a star [in the park]] [with a telescope.]*

2) *I saw [a star [in the park [with a telescope]]].*

3) *I saw [a star] [in the park [with a telescope]].*

Depending on whether *with a telescope* attaches with the *star*, in which case it might mean a tv or film star having a telescope, or it was me who had the telescope, in which case star could mean any celestial body. Thus resolving PP attachment ambiguity can help in WSD.

2.2.1.4 WordNet and WSD

The richness of PWN (Miller et al. 1990) as a lexical semantic resource, makes it a good choice for carrying out WSD (Banerjee & Pedersen (2002), Mihalcea & Moldovan (1999), Li et al. (1995), Agirre & Rigau (1995)).

Banerjee & Pedersen (2002) made changes to the basic algorithm defined by Lesk (Lesk 1986) but rather than using Oxford Advanced Learner's Dictionary they used WordNet.

The original Lesk algorithm takes a small context around the target word in the text and looks up into the dictionary for the definitions. For a phrase, such as *coal ash*, they will look for the definitions of both the words, and see if the word *ash* was used in the sense of a color, a tree or the natural resource which started the Industrial Revolution. They will see how many words were in common between different sense of the two words, and the senses for each word that shared the maximum number of words in definitions would be used to sense tag the original words. They discovered that the words that the definitions of *coal* and *ash* shared the most for any of their senses, were *combustible*, *burn*, and *solid*. That coincided with the definition of *coal* that when burnt left *ash*, and that is the abundant natural resource.

Banerjee's use of WordNet rather than a dictionary improved the performance up to 32% accurate as compared to 16-23% for different variations of the Lesk algorithm.

Li et al. (1995) used semantic networks that exist in WordNet to create word/word relationships and later used them for WSD. The semantic network defined by the WordNet has nodes where each node carries a synset. At one node the synset defines the strict synonymy relationships between words. Each sense of a word, as we get as a result of querying the WordNet search engine, has a separate node for it in the semantic network. One level up is the parent synset of a particular synset and one level down is the child synset. Similarly sibling synsets are defined that share a common parent synset. The synonymy relationship only goes one way from the child to the parent, where the parent synset can be taken as the extended synonym of its child synset. They used the notion of semantic distance which is

proportional to the shortest distance between any two synsets in the network.

They only looked for noun objects of verbs in the given text (the Canadian Income Tax Guide) but they reckon it could be extended to cater for noun subjects as well. They used verbs as the context of the noun objects, essentially creating verb-noun word pairs and looked for semantic similarity between different nouns and verbs using the WordNet semantic network. They used different heuristic rules that were based on the idea that noun objects that shared same or similar verbs were similar. They found their method to be 72% accurate and only 4% of the results were wrong for noun object sense disambiguation. The rest were judged to be partially correct.

2.2.1.5 Multilingual Disambiguation

Multilingual resources come in handy when it comes to word sense disambiguation since a polysemous word in one language may be translated into distinct words in another. Distinct words in the other language might be due to the bias of the translator or the context of that word. Such clues are important if one wants to ascertain the true sense in which the original word is used.

Diab (2000), Diab & Resnik (2002) reported some initial investigation into using word alignment and creating sets in the target language, English in this case, for each word *F* in the source language, French in this case.

They identified a few words in French to be disambiguated, for instance *catastrophe*. After word alignment they took all the words in the English corpus aligned with the target word in French to form the target sets. For the word *catastrophe*, the target set was *disaster*, *tragedy* and *situation*.

Then within the target set they considered all possible subsets of pairs of words and looked at their senses in WordNet and estimated which sense of a word got support from which sense of another word in the sense. For instance,

the word *tragedy* might mean a kind of drama, as opposed to say comedy. But that would get little support from senses of the word *disaster*. The *calamity* sense of *tragedy* gets more support from senses of other words in the set. That helps in narrowing down the senses of the words in the target set.

Since we know the instances in English that correspond to the target set *disaster*, *tragedy* and *situation*, we assign them the sense tag *calamity*. This sense tag is later propagated to instances in French, and the instances where *catastrophe* are aligned with the target set *disaster*, *tragedy* and *situation*, are all assigned the sense tag *calamity*. That is how WSD is performed.

Another example is that of the word *bank*, which can be used either in the sense of *shore*, or in the sense of a *financial institution*. Using PWN (Miller et al. 1990) taxonomies, distances were measured between different senses of all the words in the target set. The French word *rive* would translate into *bank* and *shore* in the parallel English corpus. Bank has 10 different senses defined in the WordNet 1.6, with only the second sense corresponding to the river bank. Shore has two senses defined with the first one a more appropriate translation of *rive*. Thus distance between the senses of *bank* and *shore* related to bank of a body of water would be expected to be lower than say the distance between *shore* and the financial institution sense of *bank*. Propagating the assigned sense to the tokens in the original corpora essentially formed the WSD step. They evaluated their algorithm on an artificially created corpora and found the accuracy to be up to 79%.

2.2.1.6 Disambiguation in Wikipedia

Wikipedia, owing to its vastness of information, diversity of topics covered, and the number of languages represented, is a useful resource that can be put to the task of WSD. It already has some ways of dealing with disambiguation.

The disambiguation process in Wikipedia deals with the problem in page titles. For instance, there are two cities by the name Hyderabad, one in Pakistan and one in India, and there are scores of other things that start with the word Hyderabad, which interest people enough to have written separate pages on Wikipedia. Resolving such plurality of meanings is a tricky issue.

Disambiguation Links

If a user searches for a term on Wikipedia that has ambiguity, in the sense that more than one page are associated with that term describing different concepts, then Wikipedia helps the user in reaching the correct page.

Topics that are ambiguous have a *hatnote* on the top of their webpage indicating to the user that the word is ambiguous and guides them to other uses of the term. If there is only one other webpage then a link to it is provided. If there are more than one, then it provides a link to disambiguation page, listing all the senses of that term for which webpages exist in Wikipedia.

If the majority of the people agree on one particular meaning of the word, then the Wikipedia [15] takes us to a webpage related to that particular meaning. On that page it also has a link that directs us to a page with links to all different meanings of the word.

When there is a general disagreement on the meanings of an ambiguous word, then Wikipedia does not lead us to the page of any particular meaning but directs us to a page with links to all different meanings of the word.

[15] http://en.wikipedia.org/wiki/Wikipedia:Links_to_disambiguating_pages

Types of ambiguities

Name place ambiguity, as explained above, is one type of ambiguity, where many places share the same name. For instance, there are cities by the name *London*, in both the UK and Canada. The same could be true for the names of people. Two or more different people with same names could have entries on Wikipedia. Similarly, certain classes of concepts might have ambiguity as well. For instance, the word *plant*, might mean a living thing, or an industrial unit.

2.2.1.7 Using Wikipedia for WSD

Mihalcea (2007) used Wikipedia as a source of sense annotations. Hyperlinks within Wikipedia are created using unique identifiers, which consist of one or more words separated by spaces or underscores, and occasionally parenthetical explanations. These identifiers are also reflected in the URLs. For instance, http://en.wikipedia.org/wiki/Space_Music_%28album%29 is the URL for Space Music (album), which incorporates all the three words in it. *Anchor text* represents the surface form of the hyperlinks. Another example is "Henry Barnard, [[United States—American]] [[educationalist]], was born in [[Hartford, Connecticut]]". It contains links to the Wikipedia pages on *United States*, *educationalist*, and *Hartford, Connecticut*. The double brackets surrounding words convert surface forms into hyperlinks. *[[United States—American]]* is a special kind of link known as the *piped link*, which links the surface form *American* to the Wikipedia article *United States*. These links can be used as *sense annotations* for WSD.

They used hyperlinks as sense annotations for the corresponding concepts. Since the hyperlinks are created by the users, they are mostly accurate and lead to the correct pages. They used the links for all the hyperlinked occurrences for the given word, thus for the word *bar*, five annotations were extracted: *bar(counter)*,

bar(establishment), *bar(landform)*, *bar(law)*, and *bar(music)*.

For a word to be disambiguated they extracted all the paragraphs in Wikipedia, where that word was part of a link, or a piped link. Then they extract the left most part of the link. Thus, from the link *[[musical_notation—bar]]*, *musical_notation* is extracted what they call a label. Then the labels are mapped to their PWN senses by two human annotators. That mapping is the WSD step and thus a sense tagged corpus is created. This sense tagged corpus was then used to train a classifier, Naive Bayes in this case.

The ambiguous words used were a part of the words used in the SENSEVAL-2 and SENSEval-3. 30 words were chosen that had at least two senses in the WordNet. Two baseline cases were used: Most Frequent Sense (MFS), using an informed sense tagged corpus; and the corpus based version of the Lesk algorithm (Kilgarriff & Rosenzweig 1999). Wikipedia based WSD was found reliable with average accuracy of 84.65% using Wikipedia as compared to 72.58% for the baseline case of MFS, and 78.02% for the baseline case of Lesk-corpus. The study also showed that with increased size of data the accuracy increased.

2.2.2 Morphology

Morphology is the branch of linguistics that deals with *morphemes*, where *morphemes* are the smallest meaning bearing units of words. For instance, the word *house* is a morpheme. Another example is the word *increasingly*, which can be analyzed into three morphemes: increase, ing, and ly. Here, *increase* is the base form or the stem, *ing* indicates that after concatenating it with *increase* it becomes *increasing* which is an adjective, and *ly* indicates that after concatenating it with *increasing* it becomes an adverb.

Lately, computers have been increasingly used for carrying out morphological analysis of wordforms, giving rise to the area of *computational morphology*.

Chapter 5 in this thesis deals with the related experimental work done as one of the aims of this research work.

2.2.2.1 Analogy

de Saussure (1959) described a phenomenon in natural languages where in the long term word forms tend to follow a certain pattern as given below:

$$\begin{aligned} &Pref_1 + Suf_1 : Pref_1 + Suf_2 = \\ &Pref_2 + Suf_1 : Pref_2 + Suf_2 \end{aligned} \tag{2.14}$$

Thus, four words *walks*, *walking*, *talks*, and *talking* could be segmented as:

$$\left\{ \begin{matrix} walk \\ talk \end{matrix} \right\} \left\{ \begin{matrix} s \\ ing \end{matrix} \right\} \tag{2.15}$$

That is the correct segmentation of the words into prefixes and suffixes as any person with even rudimentary knowledge of English would figure out. But erroneous segmentations could also be created, as below:

$$\left\{ \begin{matrix} wal \\ tal \end{matrix} \right\} \left\{ \begin{matrix} ks \\ king \end{matrix} \right\} \tag{2.16}$$

Still the original words can be produced by combining the prefixes and the suffixes but the segmentation point chosen is incorrect since stems *wal* and *tal* do not exist in English and also *ks* and *king* are not valid endings either.

In order to only create the valid segmentations a heuristic can be used so that a segmentation is only valid if there is support for it in the corpus. So a segmentation would only be considered valid if within the corpus at least 3 words

are found which form the same proportion in Equation 2.3 (Pirelli 1993), (Yvon 1996).

2.2.2.2 Harris's Approach

Harris (1955) describes an unsupervised approach where utterances are segmented into phonemes. It counts the number of phonemes following a *Prefix* of phonemes, denoted by *br(n)*, where n is the prefix length. The utterance is segmented whenever *br(Prefix)* reaches a local maximum.

Harris's approach can be adapted to segment words rather than utterances and letters could be used instead of phonemes. To graphically represent it a *trie* can be generated with labeled edges, corresponding to individual letters. There is a unique root and leaves correspond to end of word markers.

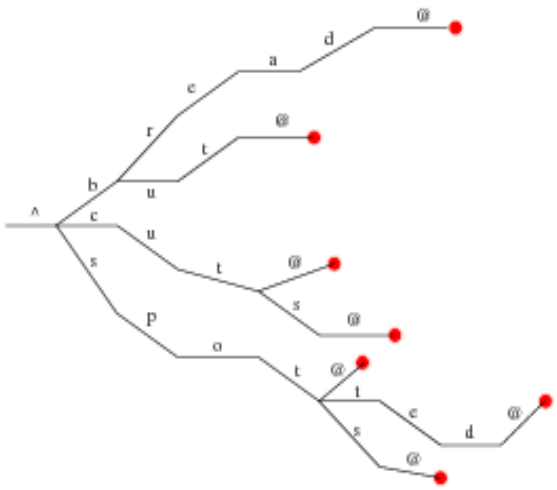


Figure 2.2: Trie structure for a list of words. (reprinted with the author's permission.)

Figure 2.2 (Kazakov & Manandhar 2001) gives a pictorial view of a *trie* for words but, cut, cuts, bread, spot, spots, and spotted. Figure 2.3 (Kazakov 2000)

```
but      : ^-3, b-2, u-1, t-1, @-0
cut      : ^-3, c-1, u-1, t-2, @-0
cut-s    : ^-3, c-1, u-1, t-2, s-1, @-0
bread    : ^-3, b-2, r-1, e-1, a-1, d-1, @-0
spot     : ^-3, s-1, p-1, o-1, t-3, @-0
spot-s   : ^-3, s-1, p-1, o-1, t-3, s-1, @-0
spot-ted : ^-3, s-1, p-1, o-1, t-3, t-1, e-1, d-1, @-0
```

Figure 2.3: Segmentation points and *br(Prefix)* (shown after the last letter of each prefix). (reprinted with the author's permission.)

gives the *br(Prefix)* values for the words, as can be computed from the *trie*. It gives segmentation points as cut-, cut-s, spot-, spot-s, and spot-ted, while but and bread are not segmented. The segmentation points are the local maxima occurring for *br(Prefix)*, as can be verified from the figure.

Figure 2.4 (Kazakov & Manandhar 2001) gives different scenarios where different shapes have different segmentation points, based on the occurrences of local maxima as depicted by the character •. If a plateau is reached as in 2.4(c), then all points on the plateau are the segmentation points provided it is followed by a downward slope.

2.2.2.3 Unsupervised Approach

Unsupervised morphological segmentation is a well researched area (Schone & Jurafsky 2000), (Goldsmith 2001), (Adler & Elhadad 2006), (Creutz & Lagus 2007), and (Dasgupta & Ng 2007).

Can & Manandhar (2009) induced morphology using unsupervised learning methodology by using POS tags as syntactic classes to separate the suffixes for pairs of words from any two clusters. They used Clark's distributional clustering approach (Clark 2000) to learn syntactic categories in an unsupervised manner.

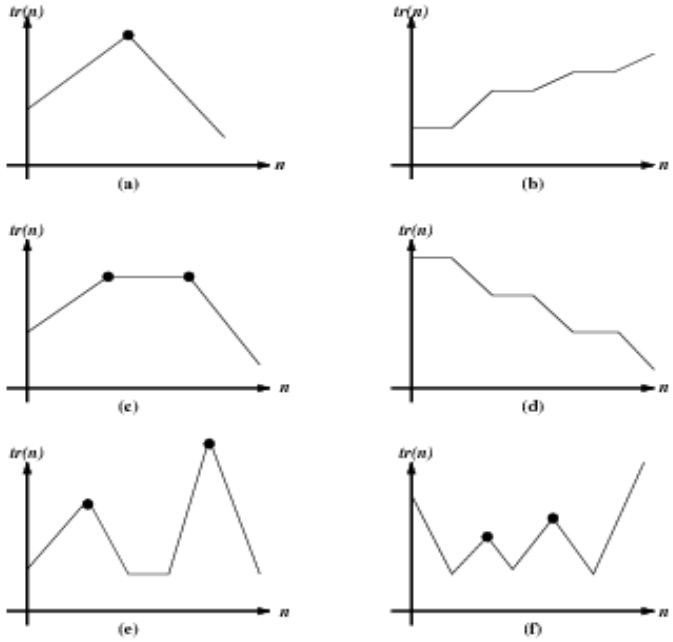


Figure 2.4: Segmentation points for various shapes of *br(n).* (reprinted with the author's permission.)

These pairs of words form the paradigms, with their stems and endings. They repeated the process for English, German and Turkish. For German the compound words were taken into account and such consideration improved precision by a significant margin though at the cost of recall. For Turkish, given its tendency for long words, a validity check was defined which would split a word into a morpheme and the rest of the word and checked if the rest of the word was a valid Turkish word. Validity checks improved precision by a big margin.

Snyder & Barzilay (2008) used non-parametric Bayesian model to identify segmentation boundaries in words in the three Semitic languages: Arabic, Hebrew and Aramaic as well as English. They showed that multilingual learning of

morphology reduced errors by 24% as compared to monolingual learning. The statistical model they used prefered more frequent morphemes.

Goldsmith (2001) used HMMs to learn morphology for modern Hebrew which is morphologically rich as compared to English. There exist certain morphological ambiguity in the analysis which needs to be removed in order to increase the proficiency of morphological analysis. They used rules for word formation for disambiguation.

Creutz & Lagus (2007) described a set of models, jointly named as *Morfessor* that created a lexicon of morpheme like structures called *morphs*, which are extracted from the corpora. Morphs do not need to carry any meaning and could just represent syllables. The use of shorter *morphs* than more stricter *morphemes* made it possible to analyze morphologically rich languages, where words are composed of multiple prefixes, stems and suffixes.

Schone & Jurafsky (2000) described a method of using the well defined technique of Latent Semantic Analysis (LSA) (Deerwester et al. 1990), (Landauer et al. 1998) which uses Singular Value Decomposition (SVD) to take a term-term matrix and decompose it into three matrices U, D, and VT, where D contains the singular values (squared eigenvalues). These values can be ranked so only a few of them are chosen, the rest are truncated, by ensuring that any loss of information would be minimal. They showed that semantics helped in morphology at least as much as frequency based approach.

2.2.3 Information Retrieval

Information Retrieval (IR) is the task of retrieving documents on the basis of a query given by the user. The documents are indexed according to their content, and that helps in quick retrieval since rather than looking at the whole set of documents, the retrieval system only looks at the indexes. The indexes can be

used to cluster documents based on some similarity metric. When a person gives a query, then the query is matched with index terms and the cluster with index terms closest to the query is retrieved.

2.2.3.1 Vector Space Model

Salton et al. (1975) defined the vector space model for indexing. According to them each document could be taken as a point in a multi-dimensional space where each dimension corresponds to a term, (*index element*) in the index. A vector can be drawn from the origin to each point, what could be termed as an *index vector*. If all the index vectors are normalized to one, the documents are nothing more than points on the envelope of a sphere with a unit radius. How close or how far the points are defines how similar or different they are.

In case there are t different terms (dimensions) and D_i different documents in the document space, each document could be represented by a t-dimensional vector $D_i = (d_{i1}, d_{i2}, ..., d_{it})$, where d_{ij} represents the weight of the jth term. Assigning different weights to different terms affects clustering, which might ultimately affect retrieval.

Since points in the near neighborhood of each other correspond to broadly similar documents, any retrieval effort might not just retrieve the one best document, but might also fetch many documents in its neighborhood. Such an approach would broaden the horizon and relevance of search. However if the documents are far apart in the document space, chances are that only one particular document would be retrieved given a particular query. That would mean high precision output, since the only document retrieved would also be the most relevant. But in case there are also other documents in its vicinity that are also relevant to the query, and they are also retrieved along with the best document, recall would also be high along with precision.

The optimal approach would be one that tries to incorporate both the above mentioned characteristics: it does not only have documents far apart that are low on relevance, but also has documents in the neighborhood that are also high on relevance. It results into a clustered document space, where similar documents are found in clusters while the irrelevant documents are found in other clusters. Clusters may also overlap with a document belonging to two or more classes. Each cluster is defined by its centroid, which is basically obtained by taking averages of each index element in all the documents in that cluster.

Thus for a cluster K with m documents, each element of centroid C may be defined as the average weight of the same elements in the corresponding document vectors, that is,

$$c_j = \frac{1}{m} \sum_{i=1}^{m} d_{ij} \tag{2.17}$$

Similar to the cluster centroids, a main centroid may be defined for the entire document space. It could be obtained from the individual cluster centroids, in the same way the cluster centroids were calculated based on index vectors.

A good model is where the intra-cluster distances are small but inter-cluster distances are big, which increases the chances of increasing both recall and precision. It would thus make sense to increase similarity between documents in the same cluster, while decreasing similarity between different clusters or cluster centroids. That could be achieved by giving more weight to terms whose occurrence is highly skewed, they occur with high frequency in some clusters while they occur with very low frequencies in others, and by assigning lesser weights to terms that occur in a large number of clusters. For the purpose the standard $tf - idf$, term frequency - inverse document frequency, could be used.

2.2.3.2 TF IDF

TF stands for Term Frequency and IDF stands for Inverse Document Frequency. Combined it is the best way of weighting terms for indexing. Even though it is considered to be a heuristic, much has been written on its theoretical foundations.

Taking just TF does not take the length of documents into account (Sparck Jones 1972). Larger documents are more likely to contain the same term more frequently than the smaller documents. Thus the document lengths must be normalized. A straightforward way is to divide the TF with length of the document. It essentially normalizes each document vector to length 1 and is called as relative term frequency.

Sparck Jones (1972) in her pioneering work defined the term Inverse Document Frequency (IDF), which later became the cornerstone of research in the field of IR. It gives how common is the term in the entire document space. It is defined as below:

$$IDF(term) = \frac{|N|}{|d : t \in d|} \tag{2.18}$$

where N denotes the total number of documents in the corpus, and $|d|$ denotes the number of documents in which the term t occurs.

The aim is to increase the weights of those terms, which are more frequent in individual documents, or small sets of documents, but rare in the entire document space. They are more useful in discriminating like documents from dislike documents. The reverse is true if the term is found very frequently in the entire corpus but rarely in individual documents. Such terms may not be useful in discriminating documents and are thus assigned lower weights.

2.2.3.2.1 Zipf's Law This is an empirical law outlining an interesting relationship between the frequency of a word and its rank, as outlined below:

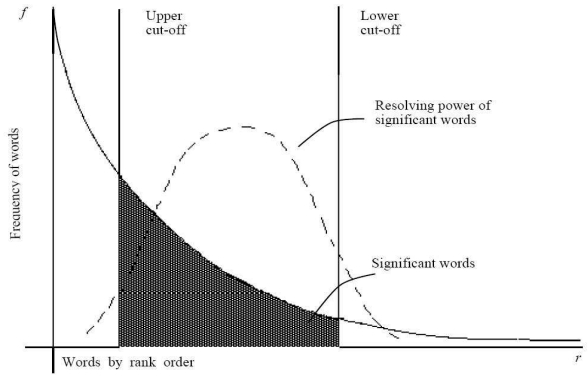


Figure 2.5: Hyperbolic curve relating the frequency of occurrence, f and the rank order, r

"The product of the frequency of use of words and the rank order (of frequency) is approximately constant (van Rijsbergen 1979)."

Luhn (1959) described two cut-off points: the upper and the lower cut-off points, see Figure 2.5. The words that exceeded the upper curve were too common and those below the lower cut-off were too rare. Both of them were not considered to be good discriminators and were thus relegated as the non-significant words. It could also be applied to phrases rather than just words.

2.2.3.3 Performance Measures

In order to evaluate the performance of IR systems, some performance measures have been defined.

Precision and Recall "Precision is the proportion of retrieved documents that are relevant, and recall is the proportion of relevant documents that are retrieved" (Buckley & Voorhees 2000).

Average Precision "The mean of the precision scores obtained after each relevant document is retrieved, using zero as the precision for relevant documents that are not retrieved" (Buckley & Voorhees 2000).

The mathematical formula for Average Precision is (Robertson 2006):

$$AveragePrecision = \frac{1}{|R|} \sum_{r \epsilon R} P@r \tag{2.19}$$

where *R* is the set of relevant documents, *r* is a single relevant document, and *P@r* is the precision at the rank position of *r*.

Its computation involves successively taking larger sets of top retrieved documents, with evenly spaced values of recall and by computing their precision. Normally five (0.1,0.3,0.5,0.7,0.9), nine or eleven recall points are chosen (Oard & Dorr 1996).

2.2.3.4 Probabilistic Information Retrieval

Miller et al. (1999) used Hidden Markov Models (HMMs) to do Information Retrieval. Their results achieved an Average Precision of 28.0%, which was quite closer to the best in TREC-7, MDS/CSIRO, which gave the Average Precision of 28.5% (Voorhees & Harman 1999).

They used a two state HMM where one state was *General English*, representing queries being posed using words that may not have anything to do with the documents being queried but might be common in any natural language queries. The other state being *Document*, which represented the queries that were based on the words taken from the documents themselves. Two transition probabilities were defined to enter these states, a_0 to enter *General English* and a_1 to enter *Document*. Based on this model and the transition probabilities, the posterior probability $P(Document\ is\ relevant/Q)$, which is the probability that a document is relevant given that a query Q is generated, was determined using the Bayes' formula, signifying the relevance of the documents based on the query.

They experimented with the TREC-6 and TREC-7 test collections, with 556,077 documents in the former and 528,155 in the latter, using 50 queries. Documents

were ranked and the rankings were compared with the standard $tf.idf$. Their HMM based system outperformed the standard $tf.idf$, exceeding it by as much as 8%.

Four refinements were made to their system using blind feedback, bigram modeling, query weighting, and document-feature dependent priors.

In the blind feedback model, first the retrieval is carried out using the user query and then based on the relevance of the top documents retrieved thus, another search is carried out before presenting the results to the user. It further improved the Average Precision by 3.5% on the TREC-6 queries.

Certain words are more meaningful when they occur in pairs, for instance *White House*, known as the bigrams. They tried to use bigrams to provide more meaning to Information Retrieval. For that purpose the *Document* state was further divided into two: *Document unigram* and *Document bigram*, with an extra transition probability a_2. The results were even better using bigrams with improvement in both TREC-6 and TREC-7 tasks.

They figured from the previous TREC evaluations that the words in the title had a dis-proportional impact on the retrieval compared to the words in the rest of the document. In order to account for this discrepancy in importance of words, they devised a simple way of repeating words in query to signify their importance. The results were further enhanced in this way with an improvement in Average Precision of 2.9% for TREC-6 and 1.2 for TREC-7. They called their methodology *Query Section Weighting*.

They also set aside the norm of using the same prior probabilities for all the documents in the Bayes' formula. Instead they used certain assumptions, such as that the articles in journals are more informative than those from a supermarket tabloid. Such assumptions assign different prior probabilities to different documents. They found that most descriptive features were source, length, and

average word-length. Using this heuristic they found marginal improvement for TREC-7 but more improvement for TREC-6.

2.2.3.5 Multi-Lingual Information Retrieval (MLIR)

Exponential growth in information on the internet, and with people from so many different countries having different mother tongues trying to express themselves in their native languages, web is becoming more and more diverse and multi-lingual in nature. In order to tap such a huge resource of instant information, techniques must be developed to cut across linguistic barriers and retrieve information in any language, given the subject of the query given by a user. Realizing such a goal is quite challenging yet people are working and trying to make it a reality.

Salton (1970) carried out one of the earliest experiments in multi-lingual IR on the SMART system. The experiments were carried out using the German and English corpora with queries in both English and German. Multilingual thesaurus was created for the said purpose by manually translating the available English version into German. The multilingual thesaurus assigned concept numbers to concept categories in English, and also provided the corresponding German translation. For instance, for the category *activity* in English, the concept number 234 is assigned and the thesaurus also contains its German translation *aktivitet*. Some of the concept categories have no corresponding entries for German. The process would take queries and documents in both English and German, compare them with the multilingual thesaurus and create the corresponding concept vectors. Since the same concept categories (numbers) are used for the same concepts in both English and German, the system can accept mixed language input and process it.

The system would create the concept vectors for queries and documents in

English and German by comparing them against the thesaurus and replacing words with concept numbers. The MLIR system then compares the query vectors in English and German, with document vectors in both English and German, essentially performing four groups of experiments: English-English, English-German, German-English, and German-German. The document vectors, in either English or German, sufficiently similar to query vectors, in either English or German, are then retrieved.

The English collection consisted of 1095 document abstracts, while the German collection consisted of 468 document abstracts, with 50 common documents. 48 queries were used both in English and German. They were originally available in English but were translated into German manually.

Salton discovered that the performance of his system on cross-lingual was almost equally efficient as on mono-lingual, with minor variations in recall values ranging from 2-3%. Yet he found the runs with German corpus to be less effective than the English one. Thus it was concluded that some aspects of the German collection needed improvement.

The problem identified related to the completeness of the thesaurus. It was found that approximately 6.5 words per English abstract were not found in the thesaurus, while the figure for the German abstracts was 15. Those missing words might be very important from the IR point of view and thus it needed to be sorted out. This was latter confirmed. One particular document with 14 missing entries, had 12 of them playing a major role in the analysis of the document. Thus the need was felt to use more complete thesaurus for future experiments.

Hull & Grefenstette (1996) worked on English documents using the translated French queries. The size of the documents they used, which consisted of news components from the TIPSTER text collection, comprised of nearly half a million documents, with a total size of 1.6 GB of text. 50 queries, selected

from the previous TREC experiments, were chosen and translated into French. The process of translation was carried out by an external translator and was not done automatically. It has been recognized that normally the queries are small in length, thus it was decided to use shorter queries, with an average of seven words per query.

They also built a word-based transfer dictionary from the on-line French-English dictionary (Oxford Hachette, 1994), by removing a large amount of excess information. Yet it encountered a lot of ambiguity in translation, since many French entries had many translations in English. For instance, one word *pendre* had 23 different translations and there were 521 entries, most of them common words, with ten or more translations. That undoubtedly introduced a lot of ambiguity in translation, exactly what makes Natural Language Processing (NLP) so hard and tedious. Yet resolving ambiguities was not done in this experiment to keep things simple. The queries were retranslated into English using the transfer dictionary and the translated queries were then input to the monolingual IR system, a modified version of SMART (Buckley 1985).

The experiment compared the Average Precision values for the original English queries, to their three different translations using different versions of the transfer dictionary: The first used the dictionary generated automatically described previously; and cleaner versions of the same. The first cleaner version simply removed entries which missed on the correct definition or were irrelevant. The second cleaner version was sought by incorporating multi-word noun phrases. The original English queries had an average precision of 0.393; Automatic word-based transfer dictionary had an average precision of 0.235; Manual word-based transfer dictionary has an average precision of 0.269; and Manual multi-word transfer dictionary had a value of 0.357. The difference in average precision scores for the first (mono-lingual) and the last case (multi-lingual) is

not significant. The conclusion that could be drawn is that Multi-lingual IR can be as efficient as mono-lingual IR, given that a comprehensive transfer dictionary is available.

2.2.3.6 Probabilistic Multi-lingual IR

Xu et al. (2001) used the HMMs for CLIR, which is an extension of the Miller et al (Miller et al. 1999), which used HMMs for monolingual IR. For this study the queries were in English and the documents were in Chinese. They used two manual lexicons and one parallel corpus. The test corpora used were TREC5 Chinese track (TREC5C) and TREC9 cross-lingual track (TREC9X).

It also defines a *General English* state and a *Document* state. The former used for generating queries that might not be relevant to the documents, and generated using some probability distribution from the available English vocabulary. The latter generating queries by selecting words from the documents at random using some probability distribution. The probabilities for entering the two states are α and 1-α respectively. α was fixed at 0.3 based on prior experience. The HMM models the query generated by a user.

Since the quality of retrieval of documents can be judged from the ranking of documents retrieved, in this study they used Baye's rule to estimate the page ranks. The aim was to ascertain the posterior probability $P(Document\ is\ relevant/Query)$, or in other words its the probability of a document being relevant given that the query Q is generated. It can be evaluated from the probability $P(Query/Document\ is\ relevant)$, which is the probability that query Q is generated given that the document is relevant, which in turn depends on which state was entered, General English or Document, to create the query. In the document state to generate the query, it chooses a Chinese word at random from the document and then translates it into English, using a manually created bilingual

lexicon on its own, using a parallel corpus on its own, and then combining both.

For the lexicons they assumed uniform translation probabilities. So if a word in Chinese could have n translations into English, each of them was equally probable. For the parallel corpus, they used statistical machine translation models (Brown et al. 1993) in order to automatically create a bilingual lexicon.

Based on the queries generated they carried out IR by retrieving documents based on queries in English. The system performed at 90% performance level of monolingual IR. They proved that using a mixture model, combining bilingual word lists and the parallel corpora, can work better than using either of them alone.

2.2.3.7 Dictionary-Based MLIR

Pirkola (1998), studied the effects of using a general dictionary and a domain-specific dictionary, using structured and unstructured queries, on CLIR and compared its performance with the monolingual IR. It was found that structured queries created using both the domain specific and the general dictionaries performed almost equally well, but not better, as the baseline queries used for monolingual IR.

The findings were based on medicine and health related queries and thus a medical dictionary was used for the purpose. The languages of interest were English and Finish, the author being a native Finnish speaker, and thus could produce quality translations of English queries into Finnish.

The study used TREC's health related topics, documents and relevance assessments. The collection comprised of 514,825 documents, with 34 health related requests. Two Machine Readable Dictionaries (MRDs) were used: a general dictionary and a medical dictionary. The general dictionary had 65,000 Finnish and 100,000 English entries, while the medical dictionary had 67,000

Finnish and English entries. The Information Retrieval system used was INQUERY.

There were four types of query types: the structured Natural language sentence based queries (structured NL/S); the unstructured Natural language sentence based queries (unstructured NL/S); the structured Natural language Word and Phrase based queries (structured NL/WP); and the unstructured Natural language Word and Phrase based queries (unstructured NL/WP). The baseline queries written in English were translated into Finnish by the author, emphasizing more on the quality of Finnish language than on precision of translation. They tilted the results in favor of the baseline queries, which is clearly evident from the comparison results.

The NL/S queries were processed in a manner that important words were extracted from them and new queries were formed, the NL/WP queries. As an example the original query (NL/S): "What research is ongoing to reduce the effects of osteoporosis and prevent the disease", was reduced to "osteoporosis prevent reduce research", which is the NL/WP. Yet it can be seen that the order of words in NL/WP has changed owing to their relative importance in the original query. The NL/WP query was translated into Finnish and then the query was expanded and structured based on the MRDs: the general and the specific (medical) dictionaries. The structured query was once again in English owing to a retranslation process using the two dictionaries. But since a word in Finnish may have many English translations, all of them were incorporated into the re-translated queries in English. That caters to some extent the disambiguities inherent in any word or phrase translations between any two languages.

Three translation methods were used in the study: *gd* translation, in which translations were done using the general dictionary; $sd \rightarrow gd$, with translations first done in the domain-specific dictionary and then in the general dictionary

only if the first one failed; *sd and gd*, translations done in both with duplicates removed.

It was observed that the baseline queries performed the best, with Precision at 10% recall at 37.9% in the case of NL/S, and 31.8% in the case of NL/WP. With unstructured queries and for NL/S, the best performance was observed in the case of *sd and gd* with P@10% being 20.4%; followed by *sd* $\rightarrow$ *gd* with P@10% of 19.2%; and least of all for the simple case of *gd* with P@10% at 15.4%. The performance was further improved with structuring of the queries, with the three results as: 30.9%; 30.4%, and 35.9%.

In the case of NL/WP and the same performance measure, the results were for unstructured: 16.5% for *gd*; 14.6% for *sd* $\rightarrow$ *gd*; and 19.3% for *sd and gd*. The results improved as before after structuring the query, with the results as: 24.9% for *gd*; 26.1% for *sd* $\rightarrow$ *gd*; and 31.1% for *sd and gd*.

2.2.3.8 Corpora based Approaches for IR

Braschler & Schäuble (1998) used document alignments to create a multilingual resource using the *relevance feedback* approach. A query in the source language would be used to retrieve documents in any of the languages in the comparable corpora. First the query would return documents in the *source language*. Document alignment mappings were then used to locate the most relevant documents in the *target language*. Terms were then extracted from the highest ranked retrieved documents, forming a new query that was used for a new search. They showed that their approach combined with pseudo-translation of the query, where the query is translated into the *target language* when the *relevance feedback* approach did not retrieve any documents. This combined approach gave better results than using any of the two approaches separately.

Latent Semantic Analysis (LSA) has also been used along with the paral-

lel corpora for Cross-Lingual Information Retrieval (CLIR) (Young 1994), and (Chew et al. 2007). Talvensaari et al. (2007) used comparable corpora for multilingual IR.

2.3 Machine Learning (ML)

Machine Learning is the branch of CS that deals with automatic learning of concepts by machines through experience, either supervised by a teacher or without him.

Supervised learning is more expensive in the NLP context, since you need previously annotated data to learn a concept. Unsupervised learning does not require previously annotated data for learning, and the software learns from the text itself. Unsupervised learning can be seen as the clustering task (Manning & Schütze 1999).

2.3.1 Clustering

Clustering is the task of partitioning objects into groups or clusters (Manning & Schütze 1999). A number of clustering techniques are defined, such as K-means (Hartigan 1975). But here we will concentrate on Hierarchical Clustering (HC) (Manning & Schütze 1999).

In HC, as the name suggests, is a clustering approach that builds a hierarchy of clusters. It can be bottom up, Hierarchical Agglomerative Clustering (HAC) or top down, known as divisive (Jain & Dubes 1988). In the bottom up approach initially every data point belongs to a separate cluster and progressively they are merged, based on some similarity metric, to form one big cluster. The merging is done based on how similar two clusters are, or how smaller is the distance between them. The divisive technique goes the other way. So initially all the

data points belong to one cluster, and progressively they are divided into smaller and smaller clusters, till each data point belongs to one cluster. The splitting is done based on coherence. So a cluster would be split if it is least coherent, or in other words the data points in it are least similar.

A number of similarity metrics can be used, such as the Euclidean distance, Manhattan Distance (Black 2006), and Cosine Distance (Lee 1999).

Based on the similarity measures, similarity functions are defined which tell between which two data points distance will be measured in order to merge or divide the clusters. Common similarity functions that are used in Information Retrieval are single link and complete link (van Rijsbergen 1979). In single link clustering, the distance is measured between two closest data points in the clusters. In complete-link clustering, the distance is measured between the two most dissimilar data points in the clusters.

Sedding & Kazakov (2004) describe WordNet based text document clustering. WordNet provides semantic relations between words in terms of synonymy and hypernymy, among others.

They built on this basic infrastructure to improve on their document clustering. They defined a few preprocessing steps: POS tagging, stopword removal, stemming, assigning WordNet Categories, pruning, and clustering, in that order. While tagging gives syntactic information, WordNet adds meaning in terms of synonymy and hypernymy.

Tagger assigns a POS tag to each word in the corpus and is done before any other modifications are done, since order of words is very important in any tagging exercise. Stopword removal removes all the words that do not add much meaning to the corpus. For this particular study all tokens that were not nouns, verbs or adjectives were removed. Stemming refers to getting the basic form of a word while removing any morphological inflections that might provide syntactic

and semantic wrapping to the word. WordNet categories, as described above, add meaning to the words. Pruning prunes all the words that occur below a certain threshold in the corpus, because they might be good discriminators but we might end up with clusters with just a few documents, and might affect the efficiency of the clustering technique in terms of recall and precision. Then the terms or words were assigned weights using the *tf.idf* weighting mechanism. Finally clustering was done using bisecting *k*-means algorithm, which was found to be the current best clustering technique (Steinbach et al. 2000).

They used Reuters-21578 as the test collection for being not specific to any domain, free availability and for comparable studies. The corpus comprised of 21578 newswire articles from 1987.

Five configurations of data were used: *Baseline*, which includes all the basic pre-processing techniques, i.e. stopword removal, stemming, pruning and weighting but POS tags are removed; *PoS_Only* that is identical to Baseline in every sense except that the POS tags are kept; *Syns* that includes all WordNet senses of each PoS tagged token over and above all other aspects in the previous configuration; *Hyper_5* that includes 5 levels of hypernyms over and above everything in Syns; and *Hyper_All* that includes all hypernym levels.

Results indicated that the quality of clustering increases with the number of clusters. Better clusters were obtained for Baseline than for any other configuration when the background knowledge was added using WordNet. That might be due to the reason that WordNet provides many senses for each word, thus for every correct sense many incorrect senses were added, which is the added noise. The results also indicated that including only five levels of hypernyms was better than using all. It could be because with added levels of hypernymy the terms become too general and loose their discriminating power, which is bad for clustering.

2.3.2 Measures of Clustering Quality

Some of the measures that could be employed to ascertain the veracity of clustering are Purity, Precision, Recall and F-score. They can be used for comparing the results of clustering with the gold standard, an external measure. Purity defined in (Wong & Fu 2000), indicates how many of the documents in a cluster are correctly assigned a class. If K is the set of clusters, C is the set of classes, N is the number of documents, W_k is a particular cluster, C_j is a particular class, and $|w_k \cap c_j|$ denotes the number of documents in cluster k that belong to a certain class, then:

$$purity(K, C) = \frac{1}{N} \sum_{k} \max_{j} |w_k \cap c_j| \tag{2.20}$$

Precision, as defined in (Church et al. 1991) creates a relationship between the fraction of documents in cluster C that also belong to class L, as below:

$$precision(C, L) = \frac{|C \cap L|}{|C|}, C \in C_{ALL}, L \in L_{ALL} \tag{2.21}$$

Recall, is defined as the fraction of documents in class L that is also in cluster C. Thus,

$$Recall(C, L) = \frac{|C \cap L|}{|L|}, C \in C_{ALL}, L \in L_{ALL} \tag{2.22}$$

F-Score (Wong & Fu 2000), (Steinbach et al. 2000), combined the evaluation metrics of both precision and recall by assigning them equal weights, as:

$$F - Score(C, L) = \frac{2 * Precision(C, L) * Recall(C, L)}{Precision(C, L) + Recall(C, L)} \tag{2.23}$$

Gini Index can also be used to ascertain the purity of clustering.

2.3.3 Decision Trees

Decision trees are a supervised learning approach with a set of examples belonging to different classes. The learning algorithm learns takes in a table of attributes and at each node of the tree decides which attribute to put, which would split the data set that helps in reducing the expected Entropy by the maximum (Mitchell 1997), or that has the maximum Informaton Gain. More details on it can be found in Section 4.7.4.

2.4 Building Resources from Corpora

Corpora, either monolingual or multilingual, parallel or comparable, are a very useful linguistic resource, which in machine readable form can be used for computational linguistics tasks. Building these resources is an expensive task in terms of human and capital costs required for the purpose. Advancements in computer technology has made it possible to harness their computational power to automate the task.

2.4.1 Extracting Linguistic Resources from Wikipedia

Adafre & de Rijke (2006) used the multilingual aspect of Wikipedia to produce parallel corpora. They also created a bilingual English-Dutch lexicon using hyperlink information on a typical Wikipedia page, which was done manually. (Ahn et al. 2004; Ferrández et al. 2007) used it to develop a cross-lingual question answering system. (Kawaba et al. 2008) used Wikipedia titles in the multilingual context to retrieve blog feeds in English and Japanese. (Potthast et al. 2008) used

Wikipedia to construct a multilingual retrieval model, using the comparable corpora in different languages in Wikipedia. (Richman & Schone 2008) used it for Multilingual Named Entity Recognition. Other uses include text classification (Gabrilovich & Markovitch 2006), information extraction (Ruiz-Casado et al. 2005), computing semantic relatedness (Zesch et al. 2007), and named entity disambiguation (Bunescu & Pasca 2006).

Automatic extraction of lexicons makes the task less labor-intensive and makes the resultant lexicons more amenable to changes and adaptable to new wordforms that keep appearing. Since, they are machine readable they are easier to use for other NLP/IR tasks or to build more resources.

Tyers & Pienaar (2008) used a list of English words to build a multilingual lexicon using multilingual nature of Wikipedia. The lexicon was built for Macedonian (mk), Afrikaans (af), Iranian Persian (fa) and Swedish (sv), the languages for which native speakers were available for manual evaluation of results. They chose a set of nouns in English and for each noun they would go to the Wikipedia webpage in English and then collect the corresponding words/phrases in other languages using the links for them on the original English webpage. Their evaluation gave Precision ranging from 69% for Swedish to 92% for Iranian Persian.

Zesch & Gurevych (2008) defined Wikipedia and Wiktionary [16] as *Collaborative Knowledge Base* CKB, as opposed to a *Linguistic Knowledge Base* LKB, such as WordNet (Fellbaum 1998).

Wiktionary[17], like Wikipedia, is freely available online and is editable by anyone with due access to the internet and with some basic knowledge of the web technologies. But unlike its cousin it attracts lesser contributions from the online community and has fewer number of languages covered with lesser cross-lingual translations available for a commonly used word, such as car, than Wikipedia.

[16] http://www.wiktionary.org/
[17] http://www.wiktionary.org/

Thus, Wikipedia is more comprehensive in that sense.

They developed Java APIs to exploit the information contained within the Wikipedia and the Wiktionary using their database dumps and have made them freely available for research purposes[18]. The APIs are useful for data mining tasks.

They imported database dumps into a database rather then using the crawler, since a crawler goes through the webserver to retrieve particular web pages and puts an extra overhead. They used indexing which is available as part of the database and makes accessing particular webpages really fast and efficient. Thus it is more suitable for large-scale NLP applications. Another disadvantage of using a crawler is that probably the results are not reproducible since the online edition of Wikipedia keeps changing, while no matter how many times you run a program on the same database dumps, they are going to yield the same results. Yet, an advantage of using the crawler is that it automatically incorporates more updated information since it directly connects with the server and latest information can be accessed as soon as it is available on the server.

Wikipaedia has also been used to extract lexicons as an auxiliary task. (Sagot & Fišer 2008) created a WordNet for French, what they called WOLF. It was based on the *extend* approach (Vossen 1996). They used freely available resources, such as: JRC-Acquis[19] parallel corpus, Wikipedia and the EUROVOC[20] thesaurus. To extract synsets for monosemous words a bilingual lexicon is enough since no disambiguation is required. For that purpose they created a bilingual English-French lexicon with 314,713 entries.

Jones et al. (2008) showed that using domain specific dictionaries improved the performance of the Cross Lingual Information Access (CLIA) systems. They

[18]http://www.ukp.tu-darmstadt.de/ukp-home/research-areas/nlp-and-wikis/
[19]http://langtech.jrc.it/JRC-Acquis.html
[20]http://eurovoc.europa.eu/

used Machine Translation (MT), augmented with domain specific phrase lexicons mined from Wikipedia, for query translation. They created domin specific bilingual lexicons for English-Spanish, Spanish-Italian and English-Italian. The domain they chose was Cultural Heritage (CH). Domains are represented as categories in Wikipedia with each category covering articles related to the domain in multiple languages.

They automatically created the lexicon in three steps. In the first step they used a crawler to collect pages in their domain of interest. In total they downloaded 458,929 English webpages. In the second step they extracted hyperlinks to in Spanish and Italian. In the third step they extracted the basenames, embedded titles within the URLs and put them together to build the multilingual lexicon. Each Wikipedia webpage has the name of the article embedded in the corresponding URL. E.g., the URL http://en.wikipedia.org/wiki/Cupid_and_Psyche is a URL to the webpage whose title is *Cupid and Psyche*. The corresponding webpage in Italian is http://en.wikipedia.org/wiki/Amore_e_Psiche. Putting *Amore e Psiche* and *Cupid and Psyche* together make an English-Italian translation pair. Their lexicon contained about 90,000, 70,000 and 80,000 multiple word phrases in English, Italian and Spanish.

Given a query, it will first be translated using the WorldLingo[21] MT system. Then they will search for the longest subsequence that would match an entry in the domain specific dictionary, and would be translated in the target language using the lexicon. The process would be repeated till no match was found. They discovered that at least one phrase was found in 90% of the queries, which shows the usefulness of the approach. For the phrases that have already been recognized and translated using the domain specific lexicons, they are translation once again to the MT system. If the two translations mis-match, the one done by the domain-

[21] http://www.worldlingo.com/

specific dictionaries takes precedence.

Their system was assessed by bilingual speakers of languages and was found to have improved upon the performance of the WorldLingo MT system. According to them 79%, 58%, 40%, and 45% of the incorrectly translated phrases were corrected using the domain specific dictionaries for EN-IT, EN-ES, IT-EN, and ES-EN respectively.

Sato (2009) used the crawler to extract an English-Japanese person-name lexicon. The algorithm picks a person-name from a pool of monolingual names and then searches for their Japanese transliterations using Yahoo Japan!'s[22] search engine and then rank them according to a transliteration score to choose the best candidate from amongst a list of possible candidates.

2.4.2 Building Multilingual Lexicons and WordNets using Parallel Corpora

Recently, there have been efforts in building WordNets for other languages automatically. Parallel corpora are a good source of lexical semantic information and hence can be used to extract that information to automatically build a WordNet.

Two approaches for creating a WordNet are the *merge approach* and the *extend approach* (Vossen 1998), whereby in the first approach a WordNet is created independently and then merged with already existing resources, while in the second a new WordNet is created using the structures and choice of words in already existing WordNet. The latter approach makes the coverage of the new WordNet limited to the words in an existing WordNet and presupposes that the same semantic relationships hold true for any language. But then you do not have to re-align the new WordNet with the old ones. BalkaNet (Tufis 2000) and MultiWordNet (Pianta et al. 2002) are examples of that.

[22] http://www.yahoo.co.jp/

Fišer (2007) used the translated versions of the George Orwell's Nineteen Eighty-Four (Dimitrova et al. 1998) in five different languages viz. English, Czech, Romanian, Bulgarian and Slovene. Pair-wise word alignment was done to create bilingual lexicons which were later combined to create a multilingual lexicon. The multilingual lexicon was then compared with the corresponding WordNets: PWN was used for English and BalkaNet for the rest. Since BalkaNet is aligned with the PWN, comparing the lexicon entries with the corresponding language WordNet in BalkaNet would give the same synset ID as for the corresponding English entry in PWN.

If all the language entries in the multilingual lexicon entry shared the same synset ID, it was assigned to the Slovene lexical entry as well. The Slovene entries that shared the same synset ID were combined into the form of a synset as they were treated as synonymous.

Sagot & Fišer (2008) used the JRC-Acquis parallel corpus[23], Wikipedia, and the EUROVOC thesaurus[24] to create the French WordNet WOLF[25], based on the *extend approach* (Vossen 1998).

Since 82% of the literals in the PWN are monosemous and hence do not require any disambiguation, bilingual translations were enough and Wikipedia and EUROVOC were used for the said purpose. For the rest of the literals in PWN which are polysemous in nature, the parallel corpus was used to create a multilingual lexicon for English, French, Romanian, Czech and Bulgarian. Since word-alignment would only yield mappings between words, multi-word expressions were not used. Entries in each language in the multilingual lexicon were then compared with the corresponding WordNet in BalkaNet. The synsets IDs were then taken from the WordNets and if all the languages, except French, shared the

[23]http://langtech.jrc.it/JRC-Acquis.html
[24]http://eurovoc.europa.eu/drupal/
[25]http://raweb.inria.fr/rapportsactivite/RA2008/alpage/uid96.html

same synset ID for the same multilingual lexical entry, the same synset ID was assigned to the French word.

WOLF contains 32,351 synsets containing 38,001 unique literals. For evaluation it was compared with the French WordNet in the EuroWordNet (Vossen 1998), FREWN. Precision was calculated as:

$$\frac{|WOLF| \cap |FREWN|}{|WOLF|} \tag{2.24}$$

and Recall was calculated using:

$$\frac{|WOLF| \cap |FREWN|}{|FREWN|} \tag{2.25}$$

They achieved precision and recall of 80.4% and 74.5% respectively over nouns, and 63.2% and 52.5% over verbs, with combined figures of 77.1% and 70.3%.

Lefever & Hoste (2010a,b, 2009) defined the Cross-Lingual Word Sense Disambiguation task which was Task 3 in SemEval-2010. The participants were asked to automatically determine the correct sense of pre-defined set of nouns from the contextual information available in the Europarl corpus in any or all of the five languages, viz. German, French, Spanish, Italian and Dutch (5 of the 11 languages in which parallel corpora are available for European Parliamentary proceedings).

A sense inventory was created by first word-aligning the parallel corpora using GIZA++ (Och & Ney 2003). The resultant alignments were then verified by certified translators, who were also asked to build the sense inventory, which lists different meanings of each of the target words. The sense inventory lists all possible combinations of words in the six languages corresponding to that meaning. and then clustering by meaning of the target word.

The sense inventory translations were used by the annotators to assign sense tags to the 20 trial and 50 test sentences each for 5 trial words and 20 test words.

They produced two frequency based baselines: one for the *Best result* evaluation and the other for the *Out-of-five* evaluation from the word alignments obtained from GIZA++ ordered by the frequency of that particular alignment in the whole corpus.

Performance Issues with Word Alignment

GIZA++ (Och & Ney 2003) is based on statistical models and hence its results may not be much reliable in case of terms that do not occur frequently. Such limitations on the part of GIZA++ renders the whole process of alignment and subsequent creation of multilingual synsets error prone and subject to noise.

Specia et al. (2005) identified that for the English-Portuguese parallel corpora GIZA++'s word alignment accuracy fell to 29%. Such low levels of accuracy might be attributed to the fact that how close or far apart are two languages linguistically. English and French or English and German might give fewer number of errors due to their linguistic proximity.

Och & Ney (2003) observed that if German was used as a source language alignment error rate (AER) was higher than if English was used as a source language in English-German word alignment, the reason being in GIZA++ German word compounds, which occur frequently, are not aligned with more than one English word. AER, as calculated by them, was only 21.1 in case of German-English translation when German was used as the source language as opposed to 10.0 when English was used as the source language for a corpus of size 0.5K. Increasing the corpus sizes to 34K reduced AER substantially to 8.8 and 4.6 respectively.

For the English-French parallel corpora, AER was 27.8 when French was used as a source language as opposed to 23.1 when English was used as a source

language for a corpus of size 0.5K. Though it declined to no more than 8.6 for either case for a corpus of size 1,470K. They provide no alignment results for English and Greek.

Charitakis (2007) reported results for English-Greek word alignment using Uplug (Tiedemann 1999) as a tool. 50.63% of the results were found to be accurate. They attributed it to the small size of the corpora used (400,091 words) and different morphology of the two langues.

CHAPTER 3

Extraction of Multilingual Lexicons from Wikipedia

The spread of internet over the years, has brought under its umbrella a diverse group of people with different linguistic, cultural, religious and political backgrounds. That is also reflected in certain online resources. This is an effort to harness the prowess of one such resource, Wikipedia, to create linguistic resources.

3.1 Main Idea

Wikipedia is an online resource of multilingual information, which has been introduced in section 2.1.1. Here, it is used in the context of creating new resources.

Figure 3.1 gives a snapshot of the English Wikipedia page on art. The title is highlighted and the links to further articles on the same topic in other languages are zoomed in for better viewing.

In this project the potential subjectivity of Wikipedia articles is of no conse-

Figure 3.1: A Wikipedia snapshot showing the links to pages in other languages.

quence as long as the titles are meaningful and correct. That is a safe presumption since administrators working for Wikipedia and their software bots, which carry out more mundane tasks such as correcting ISBN numbering, and[1] adding missing references section, together make sure that any such errors are removed if they ever occur.

The idea is to start at a Wikipedia article given by the user, extract its title and the translations of the title by following the language links on the Wikipedia page on languages of interest. The title and its translations are put together to form an entry in the multilingual lexicon. More URLs are collected from each page visited, if they are valid Wikipedia pages, following a search technique. It follows the links till the list of URLs to be searched is exhausted (Shahid & Kazakov 2009).

The rest of the chapter is organized as follows: Section 3.2 discusses the

[1]http://en.wikipedia.org/wiki/Wikipedia:Bots

methodology, and describes the crawler that we used and the search strategy adopted by it; Section 3.3 discusses the general and domain specific lexicons that we generated and also gives histograms of number of words per keyword for different languages in a particular lexicon; Section 3.4 discussed some programming related issues that we encountered during our work; Section 3.5 gives an analysis of languages in Wikipedia; Section 3.6 gives results of evaluation; and finally, Section 3.7 is for conclusion.

3.2 Methodology

In order to extract the titles and their translations, a web crawler is used. A Web crawler is software that runs either online or offline. It follows a URL to download a page and then to extract useful information if required. It may follow other pages based on links on the first URL.

There are many crawlers available on the net but the one that was chosen, for its brevity and usefulness, was the one available on the Java Sun Development Network, now part of Oracle Technology Network[2]. We have made some necessary modifications. The crawler by design used Depth First Search (DFS), whereby the crawler first goes down deep one particular path before it backtracks and looks for other options, to explore the set of URLs in its memory. This was not considered optimal for the purpose of building multilingual lexicons, because rather than looking at all the links on one page, it would go down deep one particular path visiting pages whose topic would rapidly drift away, rarely, if ever, backtracking to look at other options that might be closer to the needs of the user.

Najork & Wiener (2001) showed that the crawlers that use BFS as the search strategy find good quality pages at the early stages of the search, however, as the

[2]http://www.oracle.com/technetwork/index.html

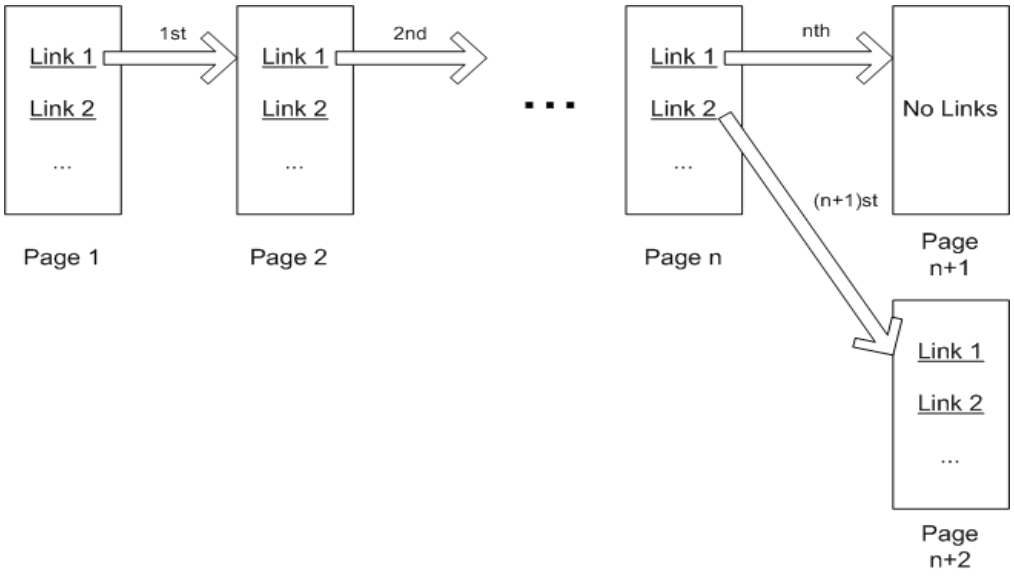


Figure 3.2: Pictorial illustration of Depth First Search.

search progresses the quality of pages also goes down. They defined the good quality pages as those having higher Page Rank (Brin & Page 1998), defined by the indegree of a page which measures how many pages with high Page Rank connect to that page. Having many links from other pages increases the chances that a high quality page is found early. BFS explores pages in the order they are discovered, reducing chances that it would vist pages whose topics drift away from the starting point. That provides credence to our choice of BFS as the search strategy.

Figures 3.2 and 3.3 give pictorial views of DFS and BFS respectively.

A starting point has to be provided to the program. This could be any Wikipedia page. The program goes to all the links on that page, one by one, and then follows them to download more webpages in the order they were originally discovered, making it BFS. It keeps track of the pages already visited so that the program avoids having redundant information. Each page that is visited by the program

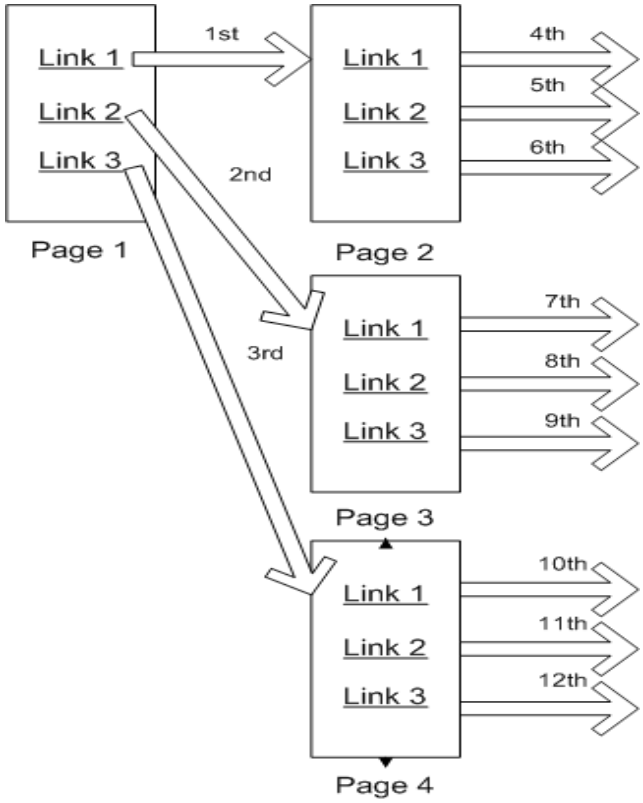


Figure 3.3: Pictorial illustration of Breadth First Search.

has links to corresponding pages in other languages.

3.3 Lexicon Generation

In order to create the lexicon, the crawler parses the HTML information of a page. It takes out the title of the original page by looking for the '⟨tiltle⟩' tags on the page and extracting only the initial information till it encounters Wikipedia related information, since each title apart from the topic that it refers to also contains some information regarding Wikipedia. For instance, the page on *Science* has the title embedded in its HTML as:

⟨title⟩Science - Wikipedia, the free encyclopedia⟨/title⟩

Where between the *title* tags we have the actual information regarding the title, which in the case above is *Science - Wikipedia, the free encyclopedia*. In it " - Wikipedia, the free encyclopedia" is redundant information and has to be removed to get the title. It is true for all the languages in Wikipeida.

It then goes to the corresponding page in the other language, takes out its title, and thus enters a row in the file, giving the keyword in the first language followed by the keyword in the other language, in the form of tuples: ⟨keyword_first_language, keyword_second_language⟩. For instance, assume that we have chosen English as the first language, and the search takes us to http://en.wikipedia.org/wiki/Computer. The title of the page is also Computer. Let us suppose we have chosen French as the second language, then in the language frame we can seek the link from the word Français to URL http://fr.wikipedia.org/wiki/Ordinateur. Our program would go to the corresponding page based on this link. That is the corresponding page in French on the same keyword as for the original language, which is Computer in this case. It takes out the title once again, which in this case is Ordinateur. The assumption is that this title is the French translation of the original English term. So the first entry that could be put into the file would be ⟨Computer,Ordinateur⟩.

The process is then repeated for other languages, each time extracting the translations of the English title and puts them in the form of tuples in the lexicon. Each such tuple forms the lexicon entry.

The web crawler works till either its "To Search" list is exhausted or the program runs out of heap space. After that we have to restart the program to get more results starting from a new page each time.

3.3.1 Algorithm

Algorithm 1 defines the whole process of crawling through Wikipedia pages and collecting titles and creating multilingual tuples to be put as entries in the lexicon.

Algorithm 1 Build Multilingual Lexicon

Data Structures
queue URLsToBeSearched (ENQUEUE-AT-END* function)
list URLsSearched (in order not to repeat the discovery of entries in the lexicon)
list MultilingualLexicon (contains keywords and their translations)
string CurrentURLToBeSearched (the URL from which new pages and titles are to be extracted)
string LinksOnCurrentURLToBeSearched (URLs on the current page which is being searched)
string Link (a link on the current page)
Initialize
$URLsToBeSearched \leftarrow \{StartingWikipediaPage\}$
$URLsSearched \leftarrow \{\}$
$MultilingualLexicon \leftarrow \{\}$
repeat
  $CurrentURLToBeSearched \leftarrow head(URLsToBeSearched)$
  $EnglishTitle \leftarrow EnglishWikipediaWebpageTitle(CurrentURLToBeSearched)$
  $ForeignTitles \leftarrow ForeignWikipediaWebpageTitles(CurrentURLToBeSearched)$
  $MultilingualLexiconEntry \leftarrow$
    $< EnglishTitle, ForeignTitle1, ..., ForeignTitleN >$
  $MultilingualLexicon \leftarrow MultilingualLexiconEntry$
  for all $Link \in LinksOnCurrentURLToBeSearched$ **do**
    if $Link$ is a valid Wikipedia page **then**
      $URLsToBeSearched \leftarrow append(URLsToBeSearched, Link)$
    end if
  end for
until $URLsToBeSearched\,! = 0$
* ENQUEUE-AT-END enqueues at the end of the queue as defined on page 74 by Russell & Norvig (1995)

The search is exhaustive in the sense that no valid link on a page would be missed and it will search for all the pages in the *to search* list until and unless

either the list is exhausted or the memory has run out.

3.3.2 General Lexicons

The first set of lexicons considered for generation are the general lexicons. General lexicons can have entries covering any conceivable domain, constrained here by the topics covered on Wikipedia, the starting point of the search, and the number of entries that is aimed for in the lexicon.

3.3.2.1 EBG and EGFP

Two sets of languages were considered for creating the lexicon. One was ⟨English, Bulgarian, Greek⟩, hence after known as EBG, and the other was ⟨English, German, French, Polish⟩, hence after known as EGFP. The program was run to extract around 20,000 entries for each dataset.

EBG is a general lexicon with more than 20,000 entries. Yet, most of the entries have nothing for Bulgarian and Greek since these 2 languages are under represented in Wikipedia. After the removal of entries that had *null* for either Bulgarian or Greek, we ended up with around 4,000 entries (see Figure 3.4 for a snapshot of the lexicon).

EGFP is also a general lexicon with around 20,000 entries collected. Of these, only 10,000 were useful in the sense that all the languages had something for the corresponding English word/phrase (see Figure 3.5 for a snapshot of the lexicon).

As can be seen from the two lexicon samples, the lexicons are fairly general and cover a wide array of topics and concepts, from religion to politics to science, geography and history. That is due to the underlying diversity of Wikipedia which is fairly general in itself.

Not every language on wikipedia is equally represented, with some languages

English	Bulgarian	Greek
Computer	Компютър	Ηλεκτρονικός υπολογιστής
Machine	Машина	Μηχανή
Digital camera	Цифров фотоапарат	Ψηφιακή φωτογραφική μη
Middle Ages	Средновековие	Μεσαίωνας
Hero of Alexandria	Херон	Ήρων
Electricity	Електричество	Ηλεκτρισμός
United Kingdom	Обединено кралство Великобритания и Северна Ирландия	Ηνωμένο Βασίλειο
Washing machine	Перална машина	Πλυντήριο ρούχων
Internet	Интернет	Διαδίκτυο
English language	Английски език	Αγγλική γλώσσα
Solar System	Слънчева система	Ηλιακό σύστημα
Square kilometre	Квадратен километър	Τετραγωνικό χιλιόμετρο
American Civil War	Американска гражданска война	Αμερικανικός Εμφύλιος Πό
Referendum	Референдум	Δημοψήφισμα
Gross domestic product	Брутен вътрешен продукт	Ακαθάριστο Εγχώριο Προϊό
Population density	Гъстота на населението	Πυκνότητα πληθυσμού
World War II	Втора световна война	Β΄ Παγκόσμιος Πόλεμος
F. Scott Fitzgerald	Франсис Скот Фицджералд	Φράνσις Σκοτ Φιτζέραλντ
Winter Olympic Games	null	Χειμερινοί Ολυμπιακοί Αγώ
European Union	Европейски съюз	Ευρωπαϊκή Ένωση
Pharmacology	Фармакология	Φαρμακολογία
Economy	Икономика (наука)	Οικονομικά
Science fiction	Фантастика	Επιστημονική φαντασία
Islam	Ислям	Ισλάμ
Arc welding	null	Ηλεκτροσυγκόλληση τόξου
Nobel Prize	Нобелова награда	Βραβείο Νόμπελ
Investment banking	Инвестиционно банкиране	null
Daylight saving time	Лятно часово време	Θερινή ώρα

Figure 3.4: Selected Entries from EBG

English	German	French	Polish
Computer	Computer	Ordinateur	Komputer
Text editor	Texteditor	Éditeur de texte	Edytor tekstu
Earth	Erde	Terre	Ziemia
Cold War	Kalter Krieg	Guerre froide	Zimna wojna
Newspaper	Zeitung	Presse écrite	Gazeta
Connecticut	Connecticut	Connecticut	Connecticut
Poet	Poet	Poète	Liryka
Country music	Country-Musik	Musique country	Muzyka country
Census	Volkszählung	Recensement	Spis statystyczny
Legislature	Legislative	Pouvoir législatif	Władza ustawodawcza
Coup d'état	Putsch	Coup d'État	Zamach stanu
The Old Man and the Sea	Der alte Mann und das Meer	Le Vieil Homme et la mer	Stary człowiek i morze

Figure 3.5: Selected Entries from EGFP

far more equal than others, such as English, which pre-dominates Wikipedia. Others have far fewer number of articles on Wikipedia such as the Greek. That causes the code to find a lot of *null* entries in languages other than English. After English, German, French, Japanese, and Spanish have the largest number of articles.

3.3.2.2 Histograms for EBG

Based on the initial results, histograms were plotted to study the length of our entries for each language. The first ones were plotted for EBG and then for EGFP.

The average number of words in a Bulgarian keyphrase, of which there were only around 7,000, as found to be only 0.608, which eludes to a large number of nulls in the lexicon.

Greek is one of the least represented among the European languages on Wikipedia. Similar to Bulgarian, most of the entries were null as well, 15,564 out of 20,569.

Since “null” entries are not of much use it was but essential to remove them and then analyze the data using the histograms. That would give a much better and clear picture of the results.

A large number of keywords/keyphrases were removed from the EBG corpus after the purging of the corpus off all the nulls. Only 4,267 keywords/keyphrases were left which constituted only a little more than 20% of the original. Figure 3.6 shows the histogram for English without nulls in the EBG corpus.

As can be seen from the histogram for English, single word titles are the most frequent, numbering 2,700, followed by phrases of length 2 that number 1,114, followed by the rest.

Removing the *nulls* reduces the number of keyphrases of length 1 from 6,681

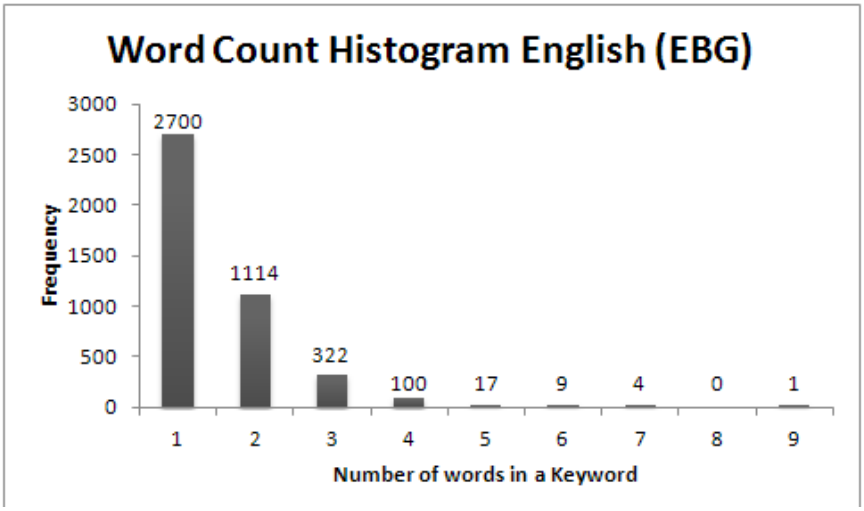


Figure 3.6: English Histogram for the EBG corpus without nulls

in the original lexicon to just 2,700, a drop of 60%. The keyphrases of length 2 have dropped from 7,892 to 1,114, a drop of around 86%. It has a reasonable explanation. The keyphrases with large number of words in them are uncommon and may refer to a very specific topic. The chances are low that a person writing in another language would also be interested to write on the same topic. In the original lexicon the largest keyphrase had a length of 26, but after the nulls have been removed the largest English phrase has a length of only 9 words, which is almost a three times drop. There are very few keyphrases of length 5 or more and hence such topics are uncommon.

Figure 3.7 gives the Bulgarian histogram without nulls. In this case, similar to that of English, the highest frequency is that of keyphrases with just one word in them, which is 2,232, which dropped from 4,031 in the original lexicon, a drop of around 45%. The ones with length 2 dropped from 2,198 to 1,462, a drop of 34%. The reason for this drop has more to do with Greek than either English

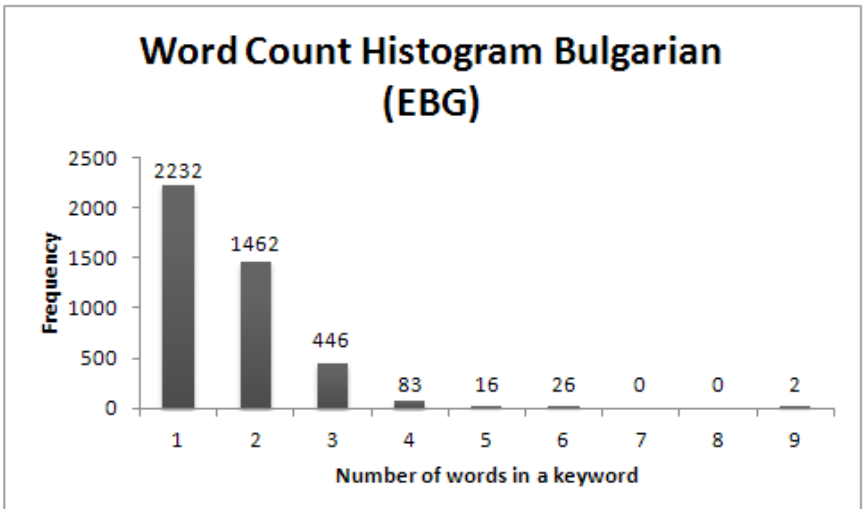


Figure 3.7: Bulgarian Histogram for the EBG corpus without nulls

or Bulgarian. Most of the keyphrases were dropped because their corresponding values for Greek were null. If it were only an English-Bulgarian lexicon, then out of a total of 20,569, 7,381 entries would have something for both English and Bulgarian, which comes to around 36% of the total. Even though 64% of the keyphrases would still have an entry only for English, it would have been a much better figure than around 80% that we have now. The largest keyphrases remain with 9 words in them.

Looking at the Greek histogram (Figure 3.8), one can see that the highest number of keyphrases are those with just one word in them, numbering 2,200, which is a very small drop as compared to 2,460 which we had originally. It translates of a drop of just over 10%. It can be explained in terms of the available resources in wikipedia in English, Bulgarian and Greek. Since Greek is the smallest of them all, there are very few one word keyphrases that have webpages in English and Greek but not Bulgarian. And that drop basically refers to those

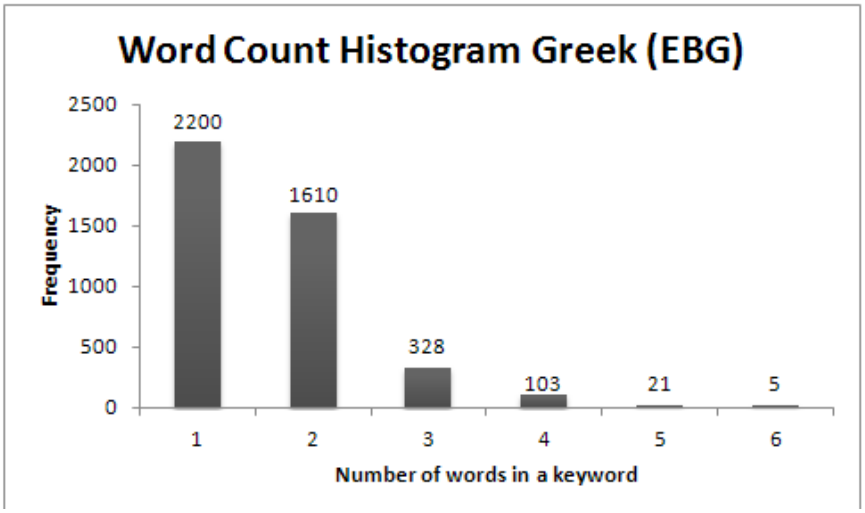


Figure 3.8: Greek Histogram for the EBG corpus without nulls

single word keyphrases that do not have a corresponding Bulgarian entry in the corpus. The largest keyphrase in Bulgarian now has 6 words in it as opposed to 7 earlier.

Across the languages, it can be seen that they tend to follow the same pattern with keyphrases of length 1 being the most frequent, followed by keyphrases of length 2. The lengths of largest phrases have much less variance now. Both English and Bulgarian have largest phrases of length 9, while Greek has the largest phrase of length 6.

3.3.2.3 Histograms for EGFP

Similar to the EBG lexicon, an EGFP lexicon was created, which corresponds to four languages, English, German, French and Polish. But opposed to the EBG lexicon, which had one of the least used European languages on the internet, the Greek, this lexicon comprises the languages which have the largest number of

articles in Wikipedia. English tops the list with over 2 million articles, followed by German with over 750,000, French with over 650,000 and Polish with over 500,000 as of 31st May 2008.

Once again the *null* entries were removed so that we only had entries in the lexicon where translations were there for all the languages considered. Figure 3.9 depicts the English histogram after the removal of nulls, that reduced the total number of entries in the lexicon from 20,383 to 10,157. Once again the phrases with length 1 are the most frequent followed by phrases of length 2. The same can be observed for other languages (Figures 3.10, 3.11, and 3.12).

Interestingly, despite being the second most used language on Wikipedia, German has *null* entries for 6,253 English phrases, that is more than 30% of the entries. French has 6,956 *null* entries and Polish has 8,805 *null* entries.

Across the languages, French and Polish have the highest length phrases with a length of 13. The figure is 10 for German and 11 for English. German also has a large number of entries of length 1, totalling 5,741. The figures for English, French and Polish are 5,026, 4,353, and 4,424 respectively. It depicts the property of German which has more compound words, not only reducing the probability of lengthy phrases but also increasing the probability of smaller length phrases.

3.3.2.4 Removal of Redundancy and Numeric Values

The multilingual lexicons thus created still contained redundancy and many of the entries were purely numeric in nature. Due to some implementation issues, the process had to be restarted and that caused some redundancy as some of the articles already vistited were revisited on subsequent runs. Also, sometimes a link on a Wikipedia page links to part of an article, and thus the same parts of the same article may be visited creating the same entry in the lexicon. For in-

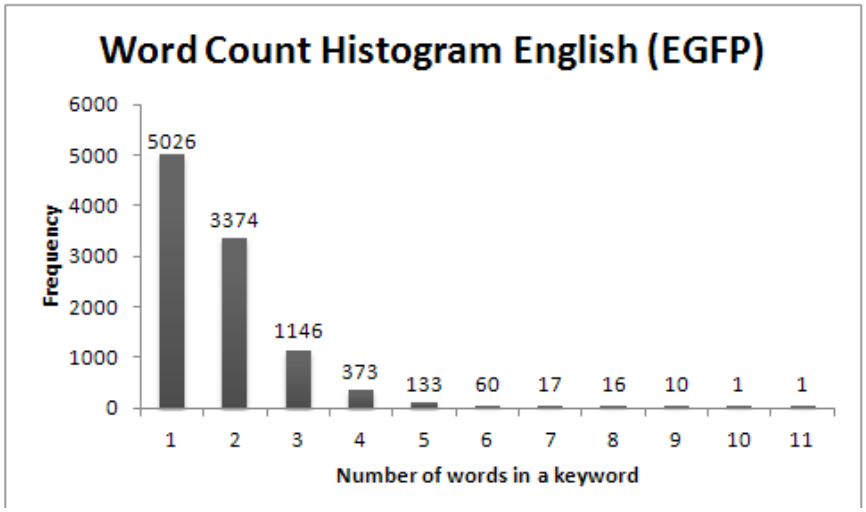


Figure 3.9: English Histogram for the EGFP corpus without nulls

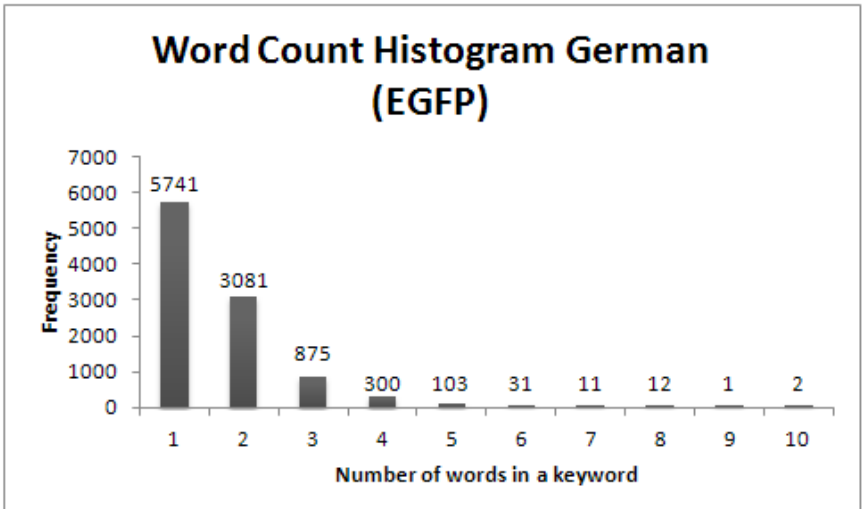


Figure 3.10: German Histogram for the EGFP corpus without nulls

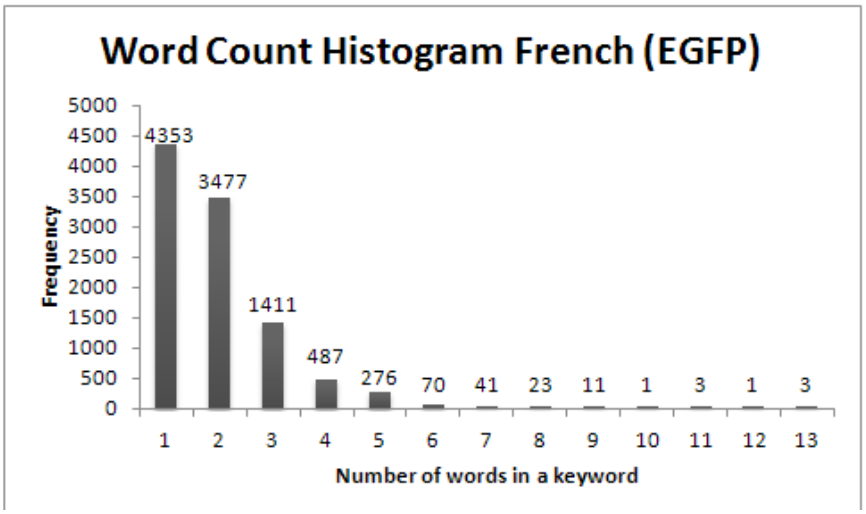


Figure 3.11: French Histogram for the EGFP corpus without nulls

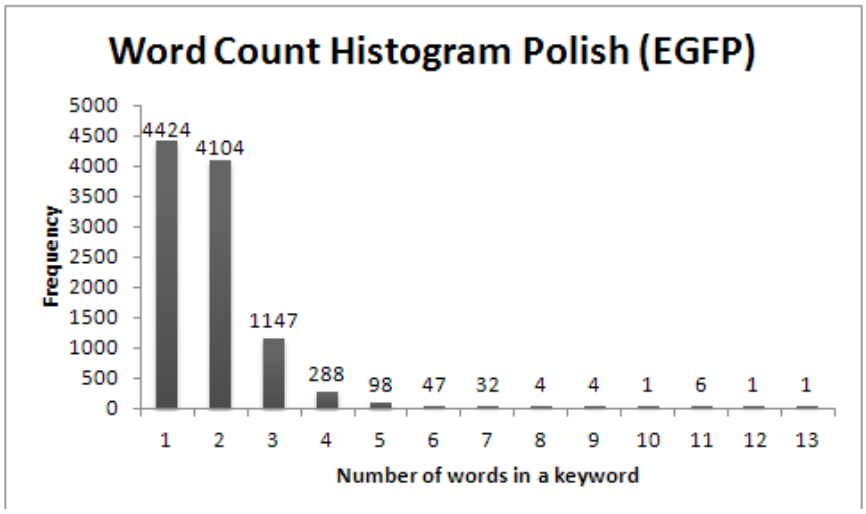


Figure 3.12: Polish Histogram for the EGFP corpus without nulls

stance, http://en.wikipedia.org/wiki/Center_of_mass and http://en.wikipedia.org/wiki/Center_of_mass#Barycenter_in_astronomy have the same title, Center of mass. Later these redundancies were removed to get only the unique entries in the lexicon.

There were also a lot of entries that merely were different years. Like there are Wikipedia articles for different years, such as 2008. But a 2008 in English would still be 2008 in any other language, barring differences in writing styles. Such entries are not of much use in any such multilingual lexicon. So apart from the entries which occurred more than once, such purely numeric entries also had to be removed.

After the removal of both, only 1,467, down from 4,267, entries were left in the EBG lexicon, and 5,109, down from 10,157, in the EGFP lexicon. That translates into a drop of 65% for EBG and a drop of 50% for EGFP. But that is just 7% of the original for EBG, and 25% of the original for EGFP.

EBG Histogram

Now we look at the histograms after the removal of redundancies and the purely numeric values. Figure 3.13 depicts the histogram for English in the EBG corpus. As can be seen a large number of the keyphrases have length of either 1 or 2, with the single word keyphrases being most common, numbering 857 (58% of the total). The largest keyphrase is of size 9.

Figure 3.14 depicts the histogram for the Bulgarian. Once again a vast majority of the keyphrases are of length 1 and 2, which make up 89% of the total. The most common are of length 2, numbering 650 (44% of the total), closely followed by keyphrases of length 1, numbering 653. The largest keyphrase is 9 words long, same as the English.

Figure 3.15 depicts the histogram for the Greek. As for English and Bulgarian, Greek also has vast majority of keyphrases of either length 1 or 2, totalling

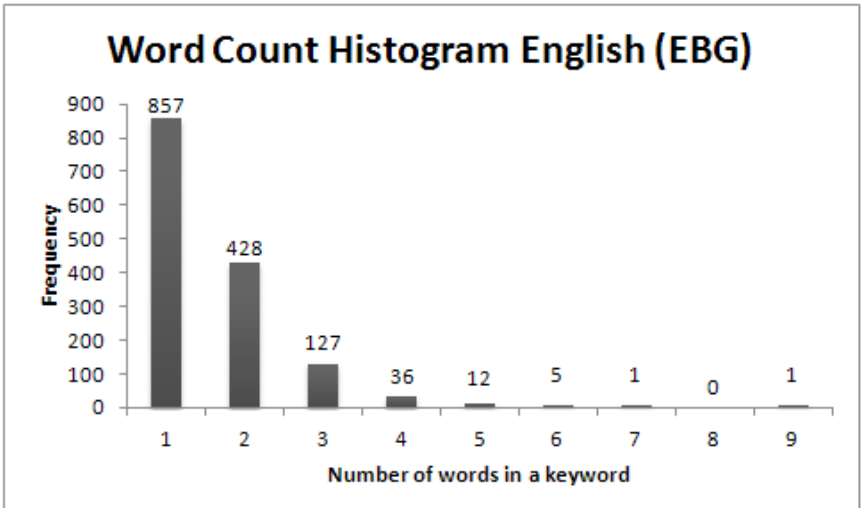


Figure 3.13: The final English Histogram for the EBG corpus

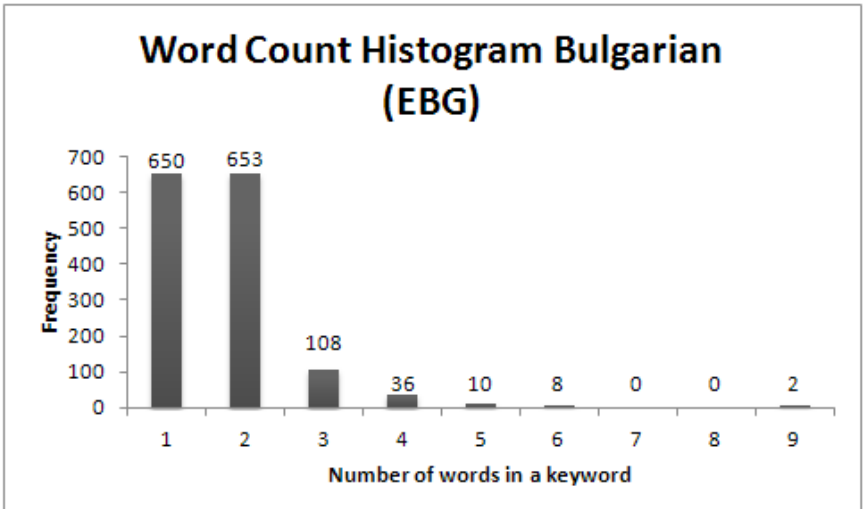


Figure 3.14: The final Bulgarian Histogram for the EBG corpus

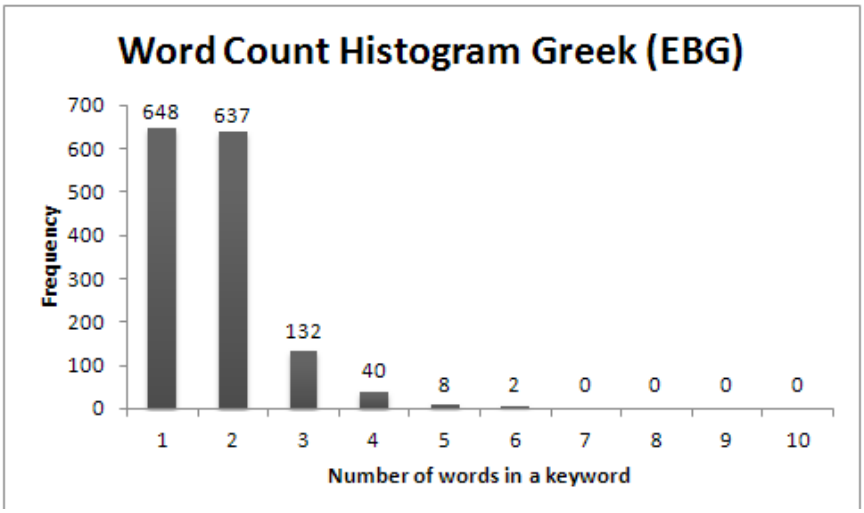


Figure 3.15: The final Greek Histogram for the EBG corpus

1,285 (88% of the total). The largest keyphrase is of size 6.

EGFP Histogram

Now we look at the histograms for the EGFP corpus after the removal of redundancies and purely numeric values. Figure 3.16 depicts the histogram for the English. A large number of entries have length 1 or 2, totalling 4,065 (2,027 of length 1 and 2,038 of length 2) out of 5,109, or 80% of them. The largest keyphrase has size 11.

Figure 3.17 is for German. 4,300 out of 5,109 entries have length of either 1 or 2, 84% of the total. The most common being of length 1 (2,452 entries), followed by length 2 (1,848 entries). The largest keyphrase is 10.

Figure 3.18 is the French histogram. But contrary to German but akin to English, in French the most frequent words are of length 2, numbering 2,099 and the second most frequent are of length 1, numbering 1,718. Combined they make up 3,817 of the total (74.5%), lesser than either English or German. The largest

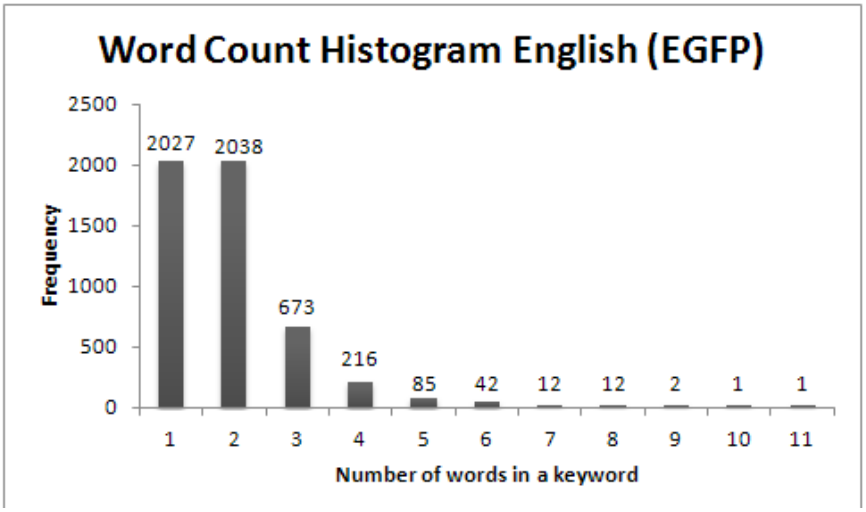


Figure 3.16: The final English Histogram for the EGFP corpus

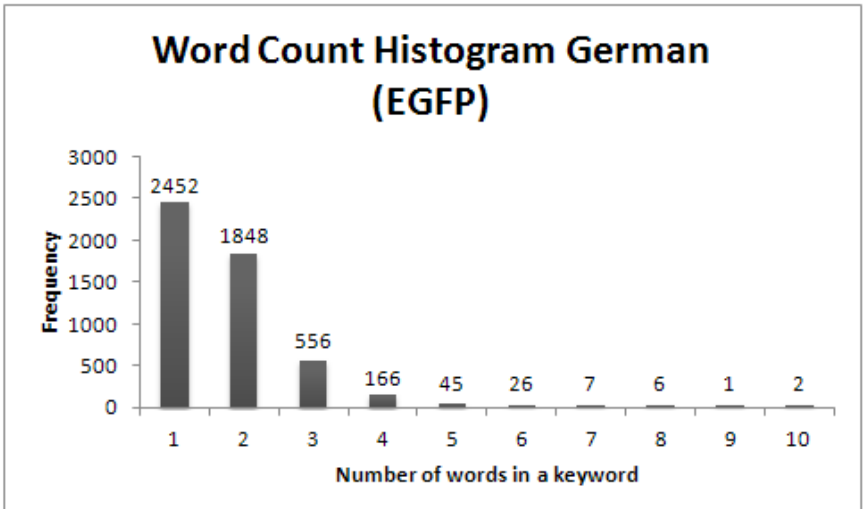


Figure 3.17: The final German Histogram for the EGFP corpus

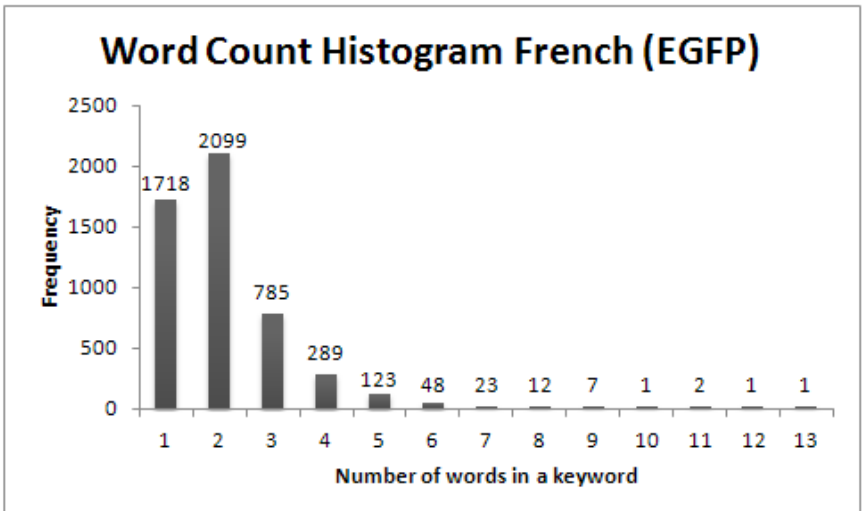


Figure 3.18: The final French Histogram for the EGFP corpus

keyphrase is of size 13.

Figure 3.19 depicts the histogram for Polish. Similar to English and French, the most common words have length 2, numbering 2,422 followed by single word keyphrases, numbering 1,761, with a combined total of 4,183 (82% of the total). The largest keyphrase is of size 13.

German, once again shows that it uses fewer words to describe the same concept. The maximum length of a phrase is 10, as opposed to 11 for English and 13 for French and Polish. In German the most frequent phrases have length 1, as opposed to 2 for all other languages. That probably indicates the use of compound words in German. French and Polish seem to use more words as in both cases there are many more phrases of length 2 than length 1.

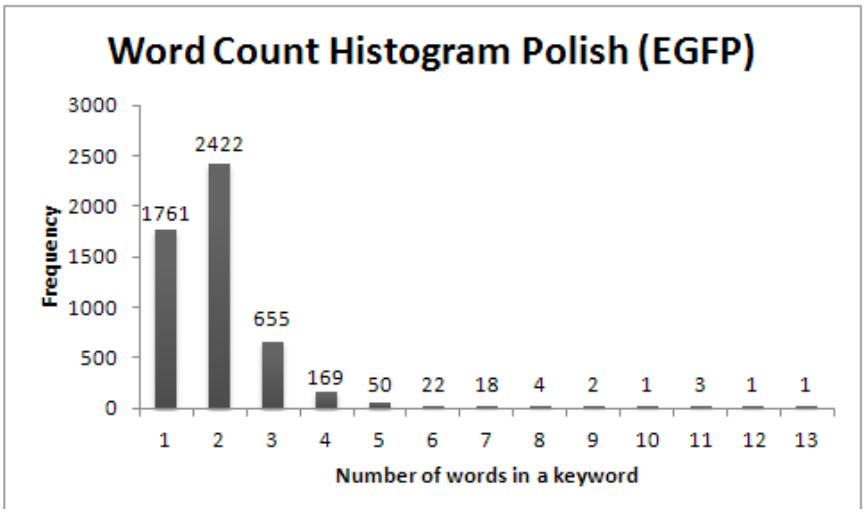


Figure 3.19: The final Polish Histogram for the EGFP corpus

3.3.2.5 HeptaLex

The EBG and EGFP were just a prelude to HeptaLex (Figure 3.20), which as the name implies is a lexicon of 7 languages: English, German, French, Polish, Bulgarian, Greek and Chinese. It has 4,603 unique entries. It is quite dense in the sense that there is only one missing value in it, there is nothing in German for the English entry “0 (number)”.

The histogram for English phrases (Figure 3.21) indicate that the most frequent are the keyphrases of length 1 as is the case with both EBG and EGFP after removal of entries containing *null* or duplicate entries.

While crawling through Wikipedia and collecting URLs of interest, several of them were found to be irrelevant to lexicon generation and hence had to be ignored. We identified parts of the page URL that indicated their relevance to the task at hand. The list is shown in (Figure 3.22).

English	German	French	Polish	Bulgarian	Greek	Chinese
Wikipedia	Wikipedia	Wikipédia	Wikipedia	Уикипедия	Βικιπαίδεια	維基百科
Encyclopedia	Enzyklopädie	Encyclopédie	Encyklopedia	Енциклопедия	Εγκυκλοπαίδεια	百科全书
English language	Englische Sprache	Anglais	Język angielski	Английски език	Αγγλική γλώσσα	英语
Venice	Venedig	Venise	Wenecja	Венеция	Βενετία	威尼斯
Film director	Regisseur	Réalisateur	Reżyser	Режисьор	Σκηνοθέτης	電影導演
Uniform Resource Locator	Uniform Resource Locator	Uniform Resource Locator	Uniform Resource Locator	Унифициран локатор на ресурси	Uniform Resource Locator	统一资源定位符
Web search engine	Suchmaschine	Moteur de recherche	Wyszukiwarka internetowa	Търсачка	Μηχανή αναζήτησης	搜索引擎
University	Hochschule	Université	Uniwersytet	Университет	Πανεπιστήμιο	大學
Monopoly	Monopol	Monopole	Monopol	Монопол	Μονοπώλιο	垄断
Computer	Computer	Ordinateur	Komputer	Компютър	Ηλεκτρονικός υπολογιστής	計算機
University of Oxford	University of Oxford	Université d'Oxford	Uniwersytet Oksfordzki	Оксфордски университет	Πανεπιστήμιο της Οξφόρδης	牛津大学
Population density	Bevölkerungsdichte	Densité de population	Gęstość zaludnienia	Гъстота на населението	Πυκνότητα πληθυσμού	人口密度
Presidential system	Präsidentielles Regierungssystem	Régime présidentiel	System prezydencki	Президентска република	Προεδρική Δημοκρατία	總統制
Dictatorship	Diktatur	Dictature	Dyktatura	Диктатура	Δικτατορία	专政
European Community	Europäische Gemeinschaft	Communauté européenne	Wspólnota Europejska	Европейска общност	Ευρωπαϊκή Κοινότητα	歐洲共同體
Benazir Bhutto	Benazir Bhutto	Benazir Bhutto	Benazir Bhutto	Беназир Бхуто	Μπεναζίρ Μπούτο	贝娜齐尔·布托
Thomas Edison	Thomas Alva Edison	Thomas Edison	Thomas Alva Edison	Томас Едисън	Τόμας Έντισον	托马斯·爱迪生
Art	Kunst	Art	Sztuka	Изкуство	Τέχνη	艺术
California	Kalifornien	Californie	Kalifornia	Калифорния	Καλιφόρνια	加利福尼亚州
Buddhism	Buddhismus	Bouddhisme	Buddyzm	Будизъм	Βουδισμός	佛教

Figure 3.20: A sample from HeptaLex

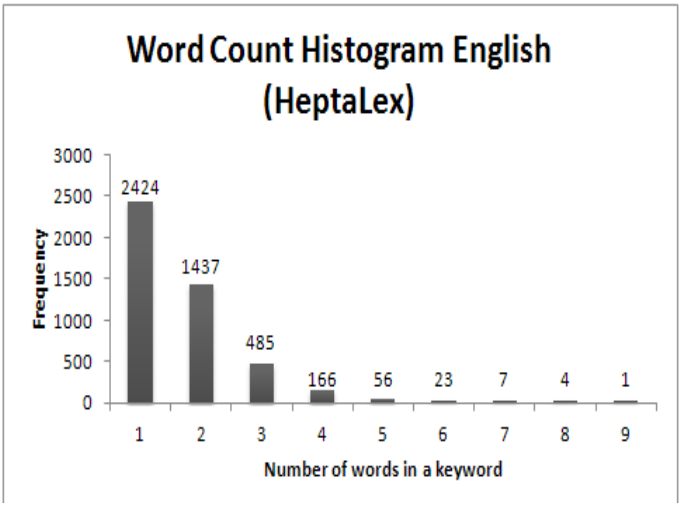


Figure 3.21: English Histogram for the HeptaLex

Substring in the Wikipedia URL	What it refers to
Image	Contains an image
Category	Wikipedia defines groups for articles. Each article can appear in more than one category.
Book sources	Gives details about the book mentioned as an ISBN number in an article.
Portal	Portal is an introductory page to a topic.
Help	Help pages for Wikipedia.
Pages that link to	Lists the pages that link to a particular page.
Talk	Wikipedia pages for discussion on articles.
#	Points to another part of the same webpage.
Special	Links that give some special information about the article, such as the recent changes.
Wikipedia	Wikipedia specific links on any page such as Wikipedia:Contact_us.
"png" and "gif"	.png and .gif files that will not give any titles.
Template	Template information for creating pages on Wikipedia.
wikimedia	Foundation that supports Wikipedia. Wikimedia links are found on Wikipedia articles.

Figure 3.22: Parts of URLs that indicated irrelevant Wikipedia pages

3.3.3 Domain Specific Dictionaries

Apart from the one general dictionary, two domain specific dictionaries were also created: one in the domain of Computer Science and the other in the domain of Artificial Intelligence with level of a category defined, which could be used to build taxonomic structures and could also be used to define relationships such as hypernymy and hyponymy.

In order to extract domain specific dictionaries use was made of categories in Wikipedia (Kazakov & Shahid 2008). A category in Wikipedia can be thought of as a particular domain, which may contain subcategories (see Figure 3.23). Each category has in it some articles on topics related to the category. For instance, the *Arts* category contains articles on *Fine art*, and *Human figure (aesthetics)* among others.

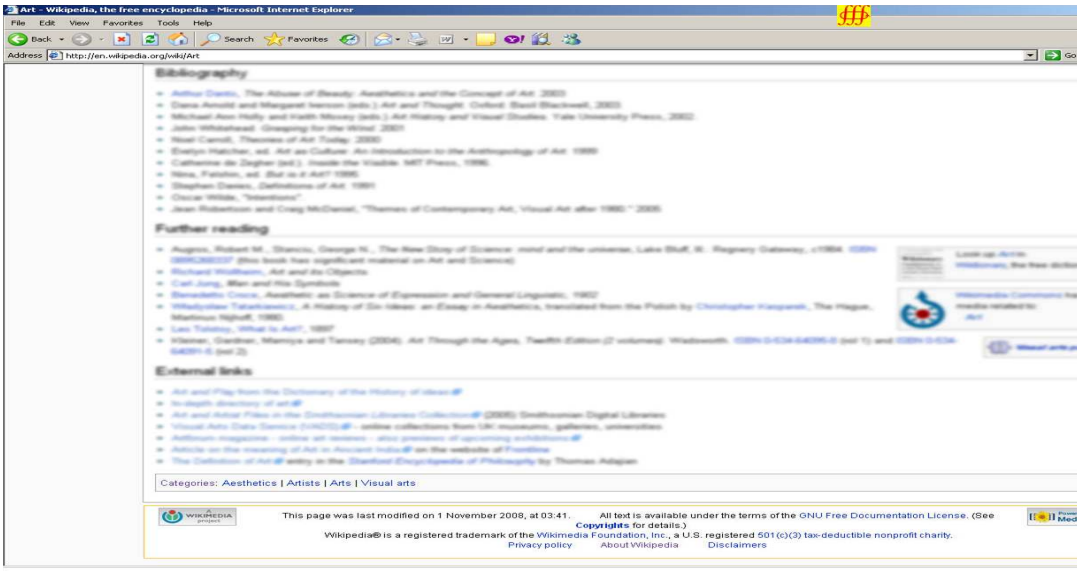


Figure 3.23: Categories on a typical Wikipedia webpage

We developed these domain specific dictionaries for *Computer Science* and *Artificial Intelligence*.

3.3.3.1 Computer Science Specific Lexicon

The Computer Science domain specific lexicon is based on the Computer Category. It has got almost 2,500 entries in it in 37 different languages: English, German, French, Polish, Japanese, Italian, Dutch, Spanish, Russian, Swedish, Chinese, Norwegian (Bokmal), Finnish, Catalan, Ukrainian, Turkish, Romanian, Czech, Hungarian, Slovak, Danish, Arabic, Korean, Lithuanian, Slovenian, Bulgarian, Estonian, Norwegian (Nynorsk), Thai, Greek, Hindi, Welsh, Latvian, Cantonese, Urdu, Irish, and Classical Chinese.

This time a wide variety of language families were considered and much bigger questions were asked, such as how any two languages may be related and could such relationships between languages be determined using information

overlap between any two languages on Wikipedia.

For the purposes of this particular lexicon, category information of Wikipedia was used, which bunches together articles belonging to one particular category. For instance, a person interested in politics might look into *Politics* category and find articles of interest on topics as diverse as *Legislative act* and *Regional autonomy*. Such diversity of information when bunched together in the form of categories and subcategories, makes searching of relevant information simpler and might also help in building taxonomies (see Algorithm 2 for details).

Algorithm 2 Algorithm for the creation of Computer Science specific lexicon

```
Get first address from CommandLine
Add address to Queue
Entries ← 1000
while Entries < 1000 do
   Get first WebPageURL from Queue
   Call URLAlreadyVisited()
   if Not visited then
      for Each link in WebPageURL do
         process link (parse and filter)
         add link to Queue
         Entries ← Entries + 1
      end for
   end if
end while
while Links in Queue > 50 do
   Get first URL from Queue
   Call findTitle()
   Remove URL from URLTable
   Insert URL into AlreadySearchedURLsQueue
end while
```

Apart from the checks put on URLs it was also ensured that only those URLs were considered which had the substring *en.wikipedia.org* in it (see Figure 3.25). It was also ensured that no such URL was considered which was about a page on a purely numeric value, such as the number *11*, since translations of *11* would be

English	Japanese	Russian	Arabic	Thai
Computer science	計算機科学	Информатика	معلوماتية	วิทยาการคอมพิวเตอร์
Computation	null	null	معلوماتية نظرية	การคำนวณ
Computer graphics	コンピュータグラフィックス	Компьютерная графика	رسوميات حاسوبية	คอมพิวเตอร์กราฟิกส์
Computer programming	プログラミング (コンピュータ)	Программирование	برمجة	การเขียนโปรแกรม
Symbolic computation	null	null	null	null
Alan Turing	アラン・チューリング	Тьюринг, Алан Матисон	آلان تورنج	แอลัน ทัวริง
Formal methods	形式手法	null	null	null
Compiler	コンパイラ	Компилятор	مصرف (برمجة)	โปรแกรมแปลโปรแกรม
Parsing	構文解析	Синтаксический анализ	null	null
Routing	ルーティング	Маршрутизация	التسيير (علم الشبكات)	null

Figure 3.24: A snapshot of the Computer Science specific lexicon with few of the languages

the same in all the languages.

Figure 3.24 shows a snapshot of the lexicon, showing some of the languages in the lexicon, depicting the variety of writing styles that have been covered. Once again, English has been used as the pivotal language, being the most prolific of all the languages on Wikipedia.

One can observe that quite a few entries for some languages are null. The lexicon itself is very sparse with some languages, such as Classical Chinese, Urdu and Welsh having fewer than 100 entries. This basically sheds light on the interest of people belonging to particular language in Computer Science and related subjects. It does not in any way mean that the language itself is not widely spoken. For instance, Urdu is quite widely spoken in South Asia but is not well represented on Wikipedia.

3.3.3.2 Category Translations

This work was further extended (Kazakov & Shahid 2008) to extract translations of categories for Computer Science (CS) and Artificial Intelligence (AI). As already shown Wikipedia defines categories that encompass different areas

Substring in the Wikipedia URL	What it refers to
Image	Contains an image
Category	Wikipedia defines groups for articles. Each article can appear in more than one category.
Book sources	Gives details about the book mentioned as an ISBN number in an article.
Portal	Portal is an introductory page to a topic.
Help	Help pages for Wikipedia.
Pages that link to	Lists the pages that link to a particular page.
Talk	Wikipedia pages for discussion on articles.
#	Points to another part of the same webpage.
Special	Links that give some special information about the article, such as the recent changes.
Wikipedia	Wikipedia specific links on any page such as Wikipedia:Contact_us.
"png" and "gif"	.png and .gif files that will not give any titles.
Template	Template information for creating pages on Wikipedia.
wikimedia	Foundation that supports Wikipedia. Wikimedia links are found on Wikipedia articles.

Figure 3.25: Substrings of URLs that render them irrelevant

of interest. Each category may contain subcategories and articles on particular topics in it (see Figures 3.26 and 3.27). In this case we only looked for categories and their subcategories for CS and AI.

The CS domain specific dictionary (Figure 3.28) contains a little over 2,000 entries in 36 different languages. Classical Chinese was left out for this exercise. The AI domain specific dictionary (Figure 3.29) was much smaller with around 450 entries.

3.4 Some Programming Related Issues

While creating the lexicon, some memory issues were encountered with Java. In Java everything is an object and objects are stored in dynamic memory, or heap space. For each URL stored on a vector, for the purposes of being explored

A
- [+] Algorithms (47)
- [+] Artificial intelligence (29)
- [+] Computer science awards (4)

C
- [+] Cellular automata (5)
- [+] Computer science competitions (2)
- [+] Computational science (15)
- [+] Computer architecture (16)
- [+] Computer programming (23)
- [+] Concurrency (6)

D
- [+] Data structures (12)
- [+] Databases (17)

E
- [+] Computer science education (1)
- [+] Events (computing) (0)

G
- [+] Computer graphics (28)

H
- [+] Human-computer interaction (15)

L
- [+] Computer science lists (0)
- [+] Computer science literature (3)

M
- [+] Mathematical optimization (5)

O
- [+] Operating systems (37)

O cont.
- [+] Computer science organizations (8)

P
- [+] Programming languages (42)

S
- [+] Computer scientists (25)
- [+] Computer security (17)
- [+] Software engineering (28)

T
- [+] Theoretical computer science (16)

W
- [+] Computer science websites (0)

μ
- [+] Computer science stubs (6)

Pages in category "Computer science"

The following 19 pages are in this category, out of 19 total. This list may sometimes be slightly out of date (learn more)

- Computer science
- List of computer science fields

*
- Topic outline of computer science
- Portal:Computer science

A
- ACM Computing Classification System
- Adaptive educational hypermedia

A cont.
- Adaptive hypermedia
- Authoring of adaptive hypermedia

C
- History of computer science
- Computer scientist

E
- Empirical modelling

H
- Charles Leonard Hamblin

I
- Informatics

I cont.
- Information and Computer Science

M
- MALINTENT

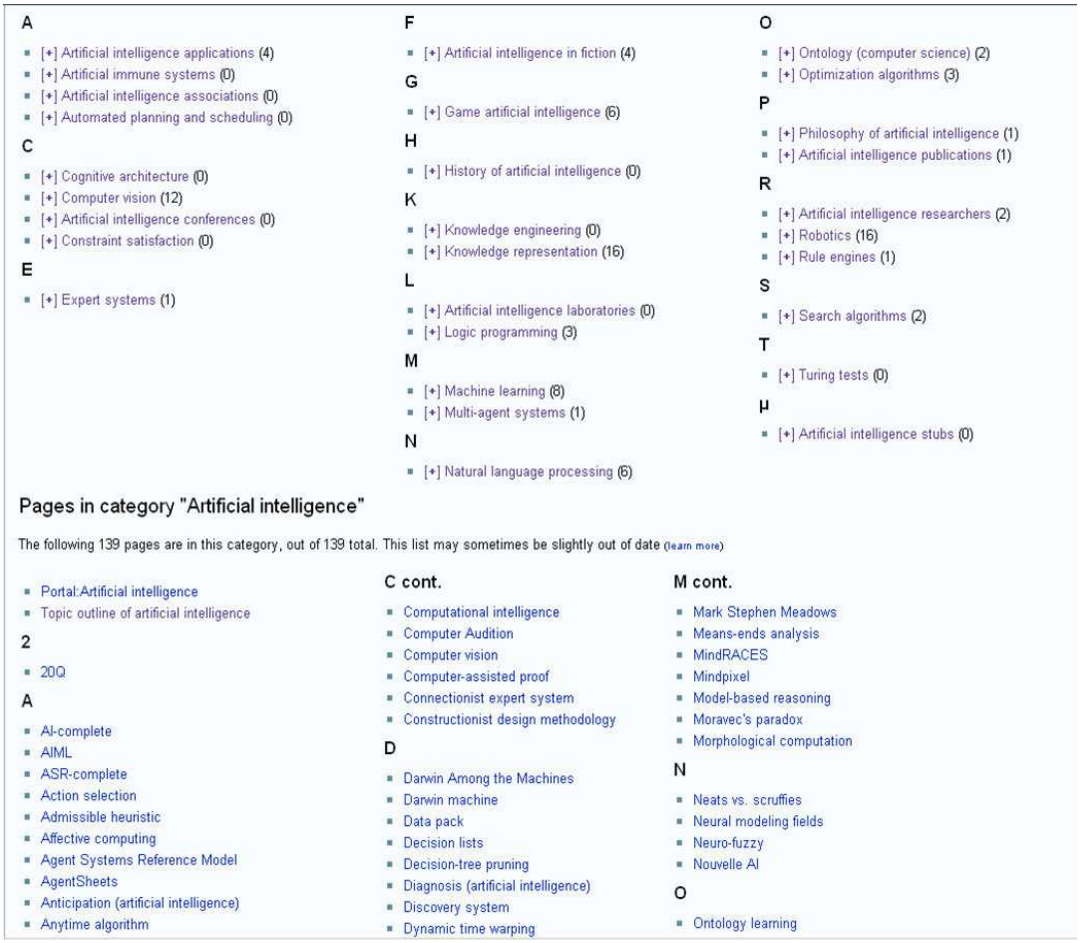
O
- Overlapping subproblem

P
- Program (mathematical object)

S
- Klaus Samelson

U
- User talk:Spychiehalla

Figure 3.26: Subcategories for Computer Science



A
- [+] Artificial intelligence applications (4)
- [+] Artificial immune systems (0)
- [+] Artificial intelligence associations (0)
- [+] Automated planning and scheduling (0)

C
- [+] Cognitive architecture (0)
- [+] Computer vision (12)
- [+] Artificial intelligence conferences (0)
- [+] Constraint satisfaction (0)

E
- [+] Expert systems (1)

F
- [+] Artificial intelligence in fiction (4)

G
- [+] Game artificial intelligence (6)

H
- [+] History of artificial intelligence (0)

K
- [+] Knowledge engineering (0)
- [+] Knowledge representation (16)

L
- [+] Artificial intelligence laboratories (0)
- [+] Logic programming (3)

M
- [+] Machine learning (8)
- [+] Multi-agent systems (1)

N
- [+] Natural language processing (6)

O
- [+] Ontology (computer science) (2)
- [+] Optimization algorithms (3)

P
- [+] Philosophy of artificial intelligence (1)
- [+] Artificial intelligence publications (1)

R
- [+] Artificial intelligence researchers (2)
- [+] Robotics (16)
- [+] Rule engines (1)

S
- [+] Search algorithms (2)

T
- [+] Turing tests (0)

μ
- [+] Artificial intelligence stubs (0)

Pages in category "Artificial intelligence"

The following 139 pages are in this category, out of 139 total. This list may sometimes be slightly out of date (learn more)

- Portal:Artificial intelligence
- Topic outline of artificial intelligence

2
- 20Q

A
- AI-complete
- AIML
- ASR-complete
- Action selection
- Admissible heuristic
- Affective computing
- Agent Systems Reference Model
- AgentSheets
- Anticipation (artificial intelligence)
- Anytime algorithm

C cont.
- Computational intelligence
- Computer Audition
- Computer vision
- Computer-assisted proof
- Connectionist expert system
- Constructionist design methodology

D
- Darwin Among the Machines
- Darwin machine
- Data pack
- Decision lists
- Decision-tree pruning
- Diagnosis (artificial intelligence)
- Discovery system
- Dynamic time warping

M cont.
- Mark Stephen Meadows
- Means-ends analysis
- MindRACES
- Mindpixel
- Model-based reasoning
- Moravec's paradox
- Morphological computation

N
- Neats vs. scruffies
- Neural modeling fields
- Neuro-fuzzy
- Nouvelle AI

O
- Ontology learning

Figure 3.27: Subcategories for Artificial Intelligence

English	German	Japanese	Chinese	Arabic	Korean	Bulgarian	Thai	Greek
Computer science	Informatik	計算機科学	计算机科学	حوسبة	전산학	Информатика	วิทยาการคอมพิวเตอร์	Επιστήμη υπολογι
Computer architecture	Rechnerarchitektur	コンピュータアーキテクチャ	電脑架構	null	컴퓨터 구조	null	null	null
Semantics	Semantik	意味論	null	null	의미론	Семантика	null	null
Algorithms	Algorithmus	アルゴリズム	算法	خوارزميات	알고리즘	Алгоритми	ขั้นตอนวิธี	Αλγόριθμοι
Artificial intelligence	Künstliche Intelligenz	人工知能	人工智能	ذكاء اصطناعي	인공지능	null	ปัญญาประดิษฐ์	Τεχνητή νοημοσύνη
Computer programming	Programmierung	プログラミング	程序设计	برمجة	컴퓨터 프로그래밍	null	การเขียนโปรแกรม	null
Operating systems	Betriebssystem	オペレーティングシステム	操作系统	نظم تشغيل	운영 체제	null	ระบบปฏิบัติการ	null
Programming languages	Programmiersprache	プログラミング言語	程序设计语言	لغات برمجة	프로그래밍 언어	Езици за програмиране	ภาษาโปรแกรม	Γλώσσες προγραμ

Figure 3.28: Lexicon for Categories of Computer Science and its Subcategories

English	German	French	Japanese	Chinese	Arabic	Korean	Bulgarian	Thai	Greek
Artificial intelligence	Künstliche Intelligenz	Intelligence artificielle	人工知能	人工智能	ذكاء اصطناعي	인공지능	Изкуствен интелект	ปัญญาประดิษฐ์	Τεχνητή ν
Chess	Schach	Échecs	チェス	国际象棋	شطرنج	체스	Шахмат	null	Σκάκι
Game theory	Spieltheorie	Théorie des jeux	ゲーム理論	博弈论	نظرية الألعاب	게임 이론	Теория на игрите	ทฤษฎีเกม	Θεωρία π
Search algorithms	null	Algorithme de recherche	検索アルゴリズム	搜尋演算法	null	검색 알고리즘	null	null	null
Machine learning	Maschinelles Lernen	null	機械学習	机器学习	تعلم آلي	기계 학습	null	การเรียนรู้ของเครื่อง	null
Robotics	Robotik	Robotique	ロボット工学	机器人学	null	로봇공학	Роботика	null	null
Computer vision	Maschinelles Sehen	Vision par ordinateur	コンピュータビジョン	计算机视觉	رؤية حاسوبية	null	null	คอมพิวเตอร์วิทัศน์	null

Figure 3.29: Lexicon for Categories of Artificial Intelligence and its Subcategories

further to collect new URLs, a new String object is created. Since each page contains a number of links, and if for each page all these URLs are put on to the vector, soon the size of the vector becomes so large so as to consume all the heap space.

To avoid such a problem from occurring we set the lower and upper limits of URLs in the list of URLs to be searched, while extracting the Computer Science specific lexicon. To avoid it becoming too big, an upper limit of 1,000 URLs was set. To reduce the chances of it running out of URLs to search for a lower limit of 50 was defined so if the number of URLs to be searched ever fell below that level the program would start looking for more.

To make searching for URLs already visited more efficient, later for the construction of Computer Science specific lexicons, hash tables were used. For the said purpose 28 different database tables were created. One which stored all the URLs to be visited. 26 tables, one for each letter in English alphabet, were used to store already visited URLs. One last table, named URLExtra, stored all other URLs. Such storing of already visited URLs improved efficiency by taking less time to figure if a particular URL has already been visited or not.

We also calculated how long it took to create a particular lexicon for both EBG and EGF. In order to get a total of 20,590 entries it took the crawler more than 5 hours for EBG, and almost 9 hours for the EGFP.

3.5 Analysis of Languages in Wikipedia

Lexicons have the basic purpose to find translations or meanings of a word. Yet, they could be used for other tasks, such as discovering which two languages are inter-related. We used the Computer Science specific lexicon for the said purpose.

We base our findings on the principle that any two languages on Wikipedia would share a fair number of concepts discussed in them if they share the cultural background. For instance, Latin and Italian are likely to have much material on the Roman Catholic Church. Similarly, languages spoken in the Middle East might have overlapping entries on Islam in their lexicons. Similarly languages that are linguistically related may have greater chances of such overlap since material in one language can easily be translated into another, with some of the shared diction, morphology and semantics.

Thus two languages with similar number of entries in the lexicon may demonstrate a different degree of overlap. Such patterns are not too difficult to identify. We used a simple relationship to calculate the degree of overlap.

$$\frac{Language_1 \cap Language_2}{Language_1 \cup Language_2} \tag{3.1}$$

where the numerator identifies the number of entries present in both the languages, and the denominator identifies the number of entries in each of the two languages.

This similarity (min = 0, max = 1) can be measured on a sample of Wikipedia pages, as we did, and the resultant clustering dendrogram is as shown in Figure 3.30.

It is difficult to make any judgements based on this data alone. Some of them make more sense, such as German and French have been bunched together but

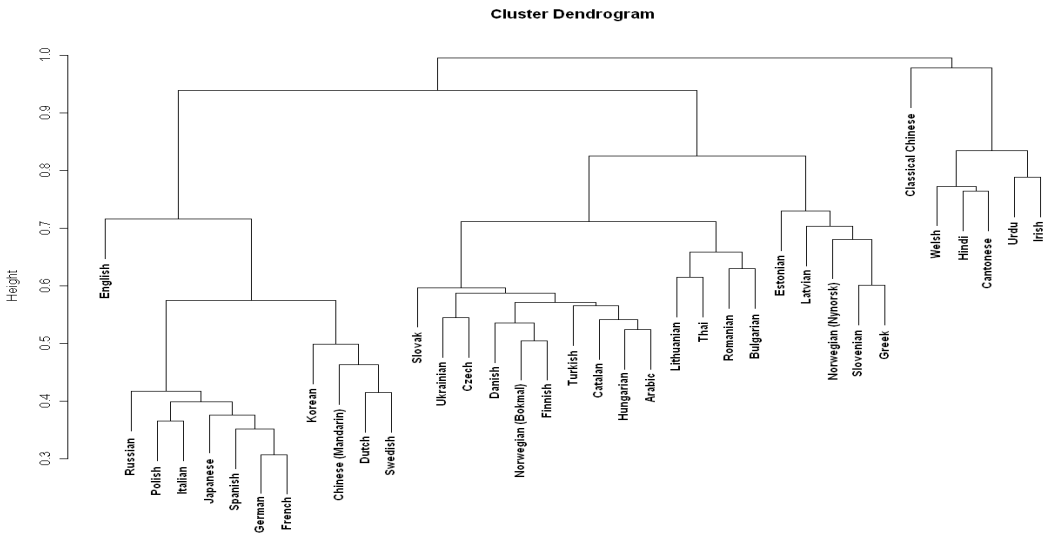


Figure 3.30: The Language Clusters for the CS Domain

others can be attributed to noise in the data, such as Urdu and Irish, which are far apart in every aspect, have been bunched together. Such grouping of languages to build taxonomic structures need closer scrutiny with less sparse data.

Two graphs (Figures 3.31 and 3.32) were plotted, one showing the number of entries in the lexicon for each language, as percentage of English entries. And the other depicting the total number of articles in each language on Wikipedia. Two of the outliers, English and Classical Chinese were removed. The first figure shows that the expected overlap between any language and English is the decreasing linear function of its rank. Even if the number of articles as a function of rank is non-linear in nature. The numbers represent the ranks of the languages, e.g. 2 for German, and 36 for Irish.

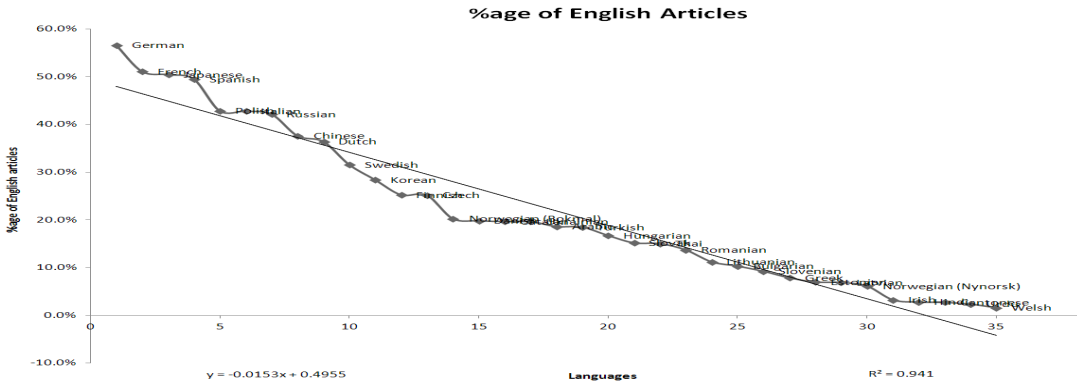


Figure 3.31: Percentage of English articles for each language with the trend line

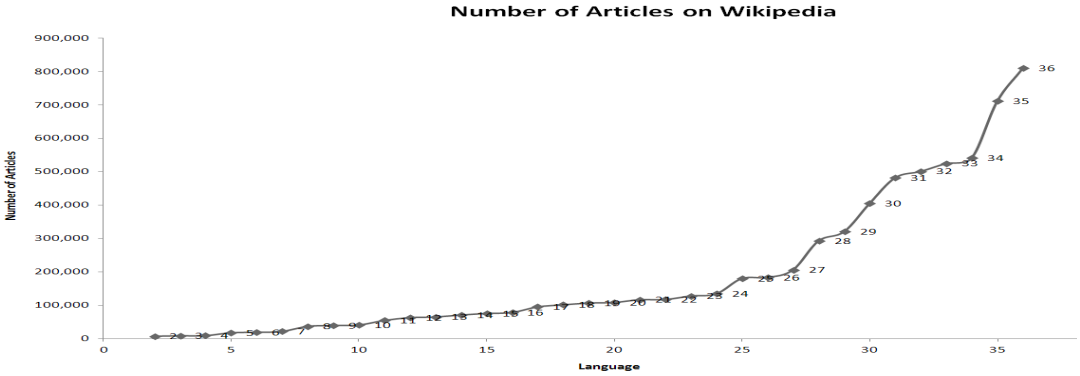


Figure 3.32: Total number of articles on Wikipedia for each language

3.6 Evaluation

We have evaluated the HeptaLex by asking native speakers to look at randomly chosen 100 entries and give their feed back if the entries were correct or not, and if they were not perfectly correct whether they were some morphological variation of what it should be? Or the translations were semantically related to the original English word using some relationship, such as hypernymy/hyponymy.

The words/phrases in English, that were chosen at random from HeptaLex are as in Tables 3.1, 3.2, and 3.3.

Table 3.4 gives results of our findings from evaluation by native speakers.

3.7 Conclusion

We used the Wikipedia articles in multiple languages to create multilingual lexicons by putting together titles of articles, which are faithful translations of each other, in the languages of interest. We created general lexicons using different set of languages: English-Bulgarian-Greek (EBG), English-German-French-Polish (EGFP), and the HeptaLex in seven different languages.

The idea was further extended to incorporate the notion of categories in Wikipedia, where each article may belong to one or more categories. Categories may be further divided into sub-categories. We have used them to create domain specific lexicons for Computer Science and Artificial Intelligence.

Most of the entries in the lexicons are of length 1 and 2. However, variation can be seen across languages in the maximum length of strings. German shows the trend of smaller phrase lengths and even the maximum phrase length is smaller than other languages. French and Polish show the opposite trend. It may indicate the use of compound words in German. French and Polish, on the other

hand, may use more words to express the same concept, and in fact after taking only the unique entries and removing the purely numerical values, both have more phrases with length 2 than with length 1. However, throughout the lexicons and across the languages, the phrases of length 3 and greater are relatively rare and their frequency drops with the increase in length.

We also calculated how probable it was to find an entry in any language given that it already exists for English. We came up with the following figures: Bulgarian 0.355, Greek 0.239, German 0.691, French 0.656, and Polish 0.565.

We also created language clusters based on the fact how many concepts are shared between any two languages. We did it for 37 languages from the Computer Science specific lexicon. We also built a relationship between English and other languages in the Computer Science specific lexicon and figured that the expected overlap between any language and English is the decreasing linear function of its rank even if the number of articles in Wikipedia as a function of rank is non-linear in nature.

Finally we created the lexicons based on the translations of categories and subcategories for Computer Science and Artificial Intelligence.

These lexicons can be used by translators and interpreters. The domain specific dictionaries can also be useful in the class where students from diverse backgrounds, specially with non-English background, end up learning things and need to discuss ideas. Such lexicons can be helpful in bridging that gap. They can also be used to create taxonomic structures by trying to ascertain the tree structures that implicitly exist on Wikipedia where each node represents a category or subcategory; and to improve performance of domain specific information retrieval systems (Jones et al. 2008).

English
Multilingualism
Language
Linguistics
Brain
Art
Official language
Recent changes
Volunteer
Communication
Grammar
Science
Philosophy
Semantics
Syntax
Translation
Population
Capital
Government
Area
Water
Population density
Time zone
Cold War
Nationalism
Kindergarten
Capital punishment
Feminism
Potato
Maize
Milk
Writing system
Economics
Human geography

Table 3.1: First part of the 100 entries chosen from HeptaLex, at random, in English

English
Law
Politics
Ethics
Jew
Recursion
Poverty
Child
Reproduction
Bone
Skin
Hair
Pregnancy
Society
Information
Advertising
Book
Loanword
Deer
Domestic sheep
Court
Nation
Natural disaster
Genocide
War
Dictatorship
Racism
Waste
Iron
Bronze
Forest
Earthquake
Ship
Fishing
Rail transport
Coal
Marriage
University

Table 3.2: Second part of the 100 entries chosen from HeptaLex, at random, in English

English
Noun
Paper
Vertebrate
Sponge
Heart
Insect
Eye
Sleep
Scissors
God
Universe
History
Religion
Crusades
Literature
Animal
Fear
Painting
Sculpture
Number
Time
Year
Experiment
Company
Poetry
Cemetery
Poet
Sound
Lion
Ice

Table 3.3: Third part of the 100 entries chosen from HeptaLex, at random, in English

Language	Percentage Correct	Hyponyms Hypernyms	Morphological Variation	Incorrect
French	93%	3%	2%	2%
Bulgarian	98%	0%	2%	0%
Greek	94%	2%	2%	2%
Chinese	97%	2%	1%	0%

Table 3.4: Results of evaluation of HeptaLex by native speakers of the languages

CHAPTER 4

Extraction of Multilingual Synsets from Aligned Corpora

Parallel corpora, which are multiple translations of the same text, carry contextual information that can be used to extract semantic information, such as which word in one language translates into which word in the other, and which two words are synonymous in a language of interest. Extraction of useful semantic information from parallel corpora forms one of the cornerstones of this thesis.

4.1 Main Idea

The idea is to take any parallel corpus, such as the European Parliamentary Proceedings (Europarl), in the languages of interest, word align them so that we know which word in one language is a translation of which word in another language, and then where there is no 1:1 correspondence, group those words to form

phrases. The sum of all translations of these phrases can be used as sense tags to disambiguate the original English corpus or the words of any other monolingual subset of the same corpus, in the same way in which a set of synonyms in the same language narrows down the meaning of a word. The results can then be evaluated using applications requiring word disambiguation, or directly on a disambiguated multilingual corpus, if such exists. The range of relevant tasks includes document clustering and document classification (Shahid & Kazakov 2011).

4.2 Assumptions

Since we used GIZA++ (Och & Ney 2003) for word alignment, we had to bring our parallel corpus in line with its constraints.

- GIZA++ does not accept sentences of length greater than 101, so we removed all the paragraphs that had length greater than that. That reduced our data set by almost one third. That still left us with 33,508 paragraphs. Och & Ney (2003) obtained best results with a corpus of size 34K sentences for English-German and 1470K sentences for English-French.
    - That might impact the performance by losing a part of the information.
- The sentence splitter that we had available, would split wherever it encountered a period. But a sentence in one language may not necessarily translate as a sentence in another. A sentence in one language may be translated into two or more sentences in another language. Thus, the splits may not be ideal.
    - Thus, we treated paragraphs as sentences for our work.

The rest of the chapter is organzied as follows: Section 4.3 discusses the parallel corpus that we used and how we pre-processed it; Section 4.4 discusses how we word aligned the corpus using GIZA++; Section 4.5 gives the procedure of collating words into phrases to create the multilingual proto-synsets; Section 4.6 discusses disambiguation by using the proto-synsets as sense tags to annotate the original English corpus; Section 4.7 discusses the challenges encountered and indirect evaluation of the proto-synsets; Section 4.8 discusses the results of evaluation; Section 4.9 discusses creation of proto-synsets using SemEval data; Section 4.10 gives a theoertical analysis of the SemEval data and provides theoretical support for use of our methodology; and finally, Section 4.11 offers some conclusions regarding the methodology.

4.3 Parallel Corpora and Pre-processing

Parallel corpora are growing in number. The availability of these corpora makes it possible to use them for NLP/IR tasks, specially in the multilingual context.

We used the European Parliamentary Proceedings (Europarl)[1] for the purposes of this research as they are freely available in 11 European languages: Romance (French, Italian, Spanish, Portuguese), Germanic (English, Dutch, German, Danish, Swedish), Greek and Finnish.

They are decently pre-processed, especially their latest versions, which are sentence aligned. When we started work on word-alignment in 2009, the Europarl corpus available at the time were not sentence aligned, and a number of pre-processing steps had to be carried out to bring it into the required input format for GIZA++, which makes use of IBM Models for word alignment and Hidden Markov Models to carry out pair-wise word alignment of parallel corpora. The

[1] http://www.statmt.org/europarl/

pre-processing proved to be quite time consuming as it involved a lot of manual work.

For our experiments, we chose four languages, namely, English, German, French and Greek. This choice provided a certain spread across the families of Indo-European languages, and also ensured that the approach and the software were not limited to texts in the Roman alphabet.

4.3.1 Structure of the Europarl Corpus

Europarl covers the debates taking place in the European Parliament, manually translated into different languages by human translators. Despite their best efforts and high expertise, the translators may use words which, at times, are not the best. The corpus also contains a certain amount of mis-alignments, missing text and other such imperfections which makes their use more difficult. We can not improve upon the translations themselves, but we can remove mis-alignments at the paragraph level. Thus a certain amount of pre-processing is all but essential to improve the alignment at the paragraph level, which ultimately contributes to improvement in word alignment.

Before discussing pre-processing steps involved it would be prudent to see how the corpus is structured in the first place and what refinements need to be made to them so that they could be used for our needs.

Any typical day in the European parliament would comprise of debates by different public figures duly moderated by the speaker of the house, which in this case is the President of the European Parliament. Since the Parliament can discuss any topic under the sun that falls under its jurisdiction, it is pertinent to label the proceedings according to topic of discussion. The tag *<CHAPTER ID...>* fills that role. Each set of debates by different speakers is covered in one chapter, and the text associated with the tag is the title of the debate. For instance, *Safety*

advisers for the transport of dangerous goods is one such topic of discussion. Each speaker's name is recorded in the tag <*SPEAKER ID...*>, which gives the number of the speaker, his/her name, and at times his/her language. Each speech is further broken down into paragraphs, which are separated and identified by the tag <*P*>.

An example is given below of a part of the English corpus (Table 4.1) and its translation in German (Table 4.2).

English
<CHAPTER ID=1> Resumption of the session <SPEAKER ID=1 NAME="President"> I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period. <P> Although, as you will have seen, the dreaded 'millennium bug' failed to materialise, still the people in a number of countries suffered a series of natural disasters that truly were dreadful. You have requested a debate on this subject in the course of the next few days, during this part-session. In the meantime, I should like to observe a minute' s silence, as a number of Members have requested, on behalf of all the victims concerned, particularly those of the terrible storms, in the various countries of the European Union. Please rise, then, for this minute' s silence. <P> (The House rose and observed a minute' s silence)

Table 4.1: A sample from English part of the Europarl corpus

Similar texts exist for other languages, including French and Greek in which we were interested.

German
<CHAPTER ID=1> Wiederaufnahme der Sitzungsperiode <SPEAKER ID=1 NAME="Die Präsidentin"> Ich erkläre die am Freitag, dem 17. Dezember unterbrochene Sitzungsperiode des Europäischen Parlaments für wiederaufgenommen, wünsche Ihnen nochmals alles Gute zum Jahreswechsel und hoffe, daß Sie schöne Ferien hatten. <P> Wie Sie feststellen konnten, ist der gefürchtete "Millenium-Bug " nicht eingetreten. Doch sind Bürger einiger unserer Mitgliedstaaten Opfer von schrecklichen Naturkatastrophen geworden. Im Parlament besteht der Wunsch nach einer Aussprache im Verlauf dieser Sitzungsperiode in den nächsten Tagen. Heute möchte ich Sie bitten - das ist auch der Wunsch einiger Kolleginnen und Kollegen -, allen Opfern der Stürme, insbesondere in den verschiedenen Ländern der Europäischen Union, in einer Schweigeminute zu gedenken. Ich bitte Sie, sich zu einer Schweigeminute zu erheben. <P> (Das Parlament erhebt sich zu einer Schweigeminute.)

Table 4.2: Sample of German translation of the English example

4.3.2 Pre-processing

The Statistical Machine Translation website, from which the corpora have been downloaded, suggests performing the following pre-processing tasks in order to use corpora with tools like GIZA++:

- tokenize the text (required)
- lowercase the text (recommended)
- strip empty lines and their correspondences (required)
- remove lines with XML-Tags (starting with "⟨") (required)

The above guidelines were religiously followed. GIZA++ takes aligned sentences to word align the parallel corpus, but we skipped the sentence alignment step since the sentence splitter available with the corpora was not able to see one-to-many relationships between sentences in different languages. At times more than one sentence in one language was aligned to a sentence in the other language. The sentence splitter would split whenever it encountered the *period*, without regards for the fact if splitting at that point would create one-to-one sentence alignments or not.

Also, due to one's lack of knowledge of any of the other languages, it was quite impossible to manually verify the sentence splits. Even if one had the requisite language knowledge, such a task would have been time prohibitive. Thus, for the sake of practicality, paragraphs were used instead of sentences as the main units providing contextual information. Such an assumption appears appropriate, although GIZA++ only accepts 'sentences' of up to 101 words, beyond which they are truncated. We shall see how this decision may have affected the performance.

The above tasks were time consuming and despite our best efforts it was not possible to manually align more than 33,508 paragraphs, which were then word aligned with GIZA++.

4.4 Word Alignment

A couple of pre-processing steps are required before the actual word alignment. These are outlined below:

1. Creation of vocabulary and sentence files

2. Creation of word classes.

4.4.1 Creation of Vocabulary and Sentence Files

A pre-processing step is required before the actual word alignment. An auxiliary tool named as *plain2snt*, which is part of the GIZA++ package, is used to create the vocabulary and sentence files for each of the two languages.

Since GIZA++ relies on statistical models for word alignment, computing the word frequencies is key to carrying out the task efficiently.

Vocabulary Files

The vocabulary file contains information about each word in the corpus. It assigns a unique ID to each and also gives its frequency of occurrence.

Table 4.3 contains a sample of the English vocabulary file used for alignment.

2 resumption 1
3 of 60474
4 the 128443
5 session 175
6 i 19160
7 declare 84
8 resumed 109
9 european 9509
10 parliament 5047

Table 4.3: Sample of the English vocabulary file

The first column is the unique identifier assigned to each word, the second is the word itself and the third is the frequency with which it occurs in the corpus. One such file is produced for each language in the language pair fed to GIZA++[2].

Sentence Files

Based on the unique IDs in the vocabulary files, a *sentence* file is generated that covers both input languages, where words are encoded with the unique IDs

[2]http://giza-pp.googlecode.com/svn/trunk/GIZA++-v2/README

from the vocabulary files and the frequency with which the sentence pair, parallel sentences, occurs in the parallel corpora. Table 4.4 contains a snapshot from the English-German sentence file.

1 2 3 4 5 (the 1st encoded sentence in English) 2 3 4 (the 1st encoded sentence in German)
1 6 7 8 4 5 3 4 9 10 11 12 13 14 15 16 17 18 6 19 20 21 22 23 24 25 26 27 28 29 30 4 31 32 25 33 26 34 35 36 37 (the 2nd encoded sentence in English) 5 6 7 8 9 10 11 12 13 14 4 15 16 17 18 19 10 20 21 22 23 24 25 26 27 28 10 29 30 31 32 33 34 (the 2nd encoded sentence in German)

Table 4.4: Sentence file containing 2 pairs of sentences for English-German

One can see information for two different sentence pairs in the snapshot above. The first line is the frequency with which the sentence pairs occur in the parallel corpora. The second line is the encoded sentence for English and the third for German. The same is repeated for all sentence pairs in the parallel corpora. The lines in Table 4.4 correspond to the sentences in Table 4.5.

English
resumption of the session i declare resumed the session of the european parliament adjourned on friday 17 december 1999 , and i would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period .
German
wiederaufnahme der sitzungsperiode ich erkläre die am freitag , dem 17. dezember unterbrochene sitzungsperiode des europäischen parlaments für wiederaufgenommen , wünsche ihnen nochmals alles gute zum jahreswechsel und hoffe , daß sie schöne ferien hatten .

Table 4.5: Sample of English-German input

4.4.2 Creation of Word Classes

To solve the problem of sparse data in language modelling, word classes are often used. People have used clustering techniques to solve that problem (Jardino & Adda (1993); Brown, Pietra, deSouza, Lai & Mercer (Brown et al.); Martin et al. (1998)). Machine Translation (MT) also faces the problem of sparse data. Creating word classes mono-lingually does not seem to be useful for MT problems (Fung & Wu 1995), since two languages are involved and we need to create some kind of correspondence between word classes for both. (Och 1999) defined the method of *bilingual word clustering* to create corresponding word classes for the two languages, that could be used for MT. *mkcls* is the tool that does this task for GIZA++. An example of a word class in English is *today tomorrow*, with the corresponding word class in Spanish as *hoy mañana mismo*. It does not mean that *today* and *tomorrow* would be translated with the same Spanish word, but that their translations would lie in the same word class.

In the next step GIZA++ was run, where one language was defined as the source language and the other as the target language. This description of languages as source or target is rather arbitrary but for sake of consistency we always defined English, the pivotal language, as the target language. But for certain languages the direction of translation may induce errors and may increase the error rates (Och & Ney 2003).

The output files generated by GIZA++ give the probablity of word alignment being correct, and also alignment scores for sentence pairs. The higher the two metrics, the better the alignment. A few examples of word alignment are given in Table 4.6.

The final output (Table 4.7) is in the form of sentence pairs in the two languages. The first line lists the sentence pair number, the length of the sentences in the source and target languages, and the alignment score for sentence pairs.

English	German	Probability of correct word alignment
lumped	zusammengefügt	0.40572
clichés	klischees	1
unambiguously	unmißverständlich	0.578567

Table 4.6: Examples of word alignment probability for English and German

Sentence pair (1) source length 3 target length 4 alignment score : 0.00126905 resumption of the session NULL ({ }) wiederaufnahme ({ 1 2 }) der ({ 3 }) sitzungsperiode ({ 4 })

Table 4.7: A snapshot of the result of word alignment for English-German for 1 sentence

The second line contains the target language sentence. The third line contains the word alignment information.

The third line in Table 4.7 is the German translation of the English sentence in the second line, with added information of word alignment. It is possible that no word in the source language is aligned to any word in the target language. That case is covered by the word *NULL* in the source language sentence. In the case above it is followed by a set of indices, which is empty. It indicates that there is no word in the target language that is not aligned with any source language word. The German word *wiederaufnahme* has two indices in its set of indices, 1 and 2. It means that this single German word is aligned with the two English words *resumption of*, which are indexed as number 1 and 2. *der* is aligned with *the*, indexed as 3, and *sitzungsperiode* is aligned with *session* indexed as 4.

The corresponding alignments for French and Greek are given in Table 4.8.

This information is later used to generate synsets such as the ones shown in Table 4.9:

The detailed procedure of how to create synsets and the algorithm will be

discussed later.

4.5 Collation of Words into Phrases

The output of GIZA++ gives word-aligned pairs of sentences for a given pair of languages as has been discussed in the previous section. We used it to word align a parallel corpus with four languages, viz. English, German, French and Greek.

It is natural that a word in one language is not necessarily translated into exactly one word in the other, but may be translated into none or more than one word in the target language. As a result a lot of contextual information which, if used properly, could be used to create *multilingual proto-synsets* (Figure 4.1), the antecedents of the multilingual synsets that are the ultimate aim of this part of the research project.

PWN synsets are sets of synonyms, for instance "dog, domestic dog, Canis familiaris" is one synset and represents one particular concept. In the case of multilingual proto-synsets, the corresponding entry would be ⟨dog, hund, chien, $\sigma\kappa\acute{\upsilon}\lambda o\varsigma$⟩.

We call these proto-synsets, rather than synsets. This can be compared with the work of PWN where lexical entries alone were used. There is a lot of non-semantic morphological variation. For instance in our case *dog* and *dogs* can appear in an otherwise identical n-tuple of words, thus creating 2 different 'proto-synsets', which would be considered to be different concepts. Hence there are

NULL ({ }) reprise ({ 1 }) de ({ 2 }) la ({ 3 }) session ({ 4 })
NULL ({ 3 }) $\epsilon\pi\alpha\nu\acute{\alpha}\lambda\eta\psi\eta$ ({ 1 }) $\tau\eta\varsigma$ ({ 2 }) $\sigma\upsilon\nu\sigma\delta o\upsilon$ ({ 4 })

Table 4.8: The corresponding French and Greek sentences for the English and German sentences in Table 4.7

resumption of	wiederaufnahme	reprise de	$\epsilon\pi\alpha\nu\acute{\alpha}\lambda\eta\psi\eta\ \tau\eta\varsigma$

Table 4.9: Generated multilingual synsets

a lot more proto-synsets then there are meanings represented by them. We shall discuss and follow up ideas about the possible ways of merging proto-synsets if the words in pivotal language are morphological - inflectional and derivational - variations of each other. As opposed to the PWN synsets, which are sets of synonyms and are mono-lingual in nature, the multilingual proto-synsets are translations of a word in different languages, four in our case. We call them synsets since translations of a word in pivotal language, all represent the same concept. Also, the PWN synsets define relationships between different synsets, such as hypernymy, and meronymy. We have not defined any such relationships, though the current project could be extended to do that.

Languages are not equal in terms of their richness of vocabulary and morphology. Thus, a word in one language can correspond to more than one word in another. German, for instance, is well known to make extensive use of compound words in its vocabulary (Berton et al. 1996). Compound German words

resumption of	wiederaufnahme	reprise de	επανάληψη της
session	sitzungsperiode	session	συνσδου
adjourned on friday	erkläre am freitag	interrompue vendredi	διακοπεί παρασκευή
like once again	nochmals	renouvelle	ξανά
pleasant festive period	ferien	vacances renouvelle vacances	περάσατε διακοπές
thank you	vielen dank	merci	ευχαριστώ
shall do so gladly	will tun gerne	ferai volontiers	πράξω ευχαρίστως

Figure 4.1: An example of proto-synsets created using our word-aligned parallel corpus.

are likely to be aligned with more than one word in English. For instance in our word aligned corpus, the German word *wiederaufnahme* aligns with English words *resumption of*, French word *merci* aligns with English words *thank you*, and the Greek word $\pi\rho o\tau\epsilon\acute{\iota}\nu\omega$ is aligned with three English words *i propose that*. The existence of such cases makes the case for collation of words into phrases. If every word in one language was at most aligned to one word in the other language, it would be enough to put together the single-word translations to form the proto-synsets. But in our case forming single-word proto-synsets would not cover the meaning of compound words in languages other than the one in which that compound word exists.

We devised our own algorithm (Algorithm 3) for the said purpose. In the

Algorithm 3 Phrase Generation Algorithm

Data Structures
int N (number of words in the PL)
int M (number of non-PLs)
int array a[1..N]
int array t[1..M,1..L]
Initialize:
for $i = 1 \rightarrow N$ **do**
  $a[i] \leftarrow i$
end for
$L \leftarrow$ number of words in $lang.l$
for $i = 1 \rightarrow L$ **do**
  if word i in $lang.l$ is aligned with word j in the PL **then**
    $t[l, i] \leftarrow j$
  else
    if word i in $lang.l$ is aligned with words $j, j+1, ..., j+k$ in the PL **then**
      $t[l, i] \leftarrow j$
      for $z = 1 \rightarrow k$ **do**
        $a[j + z] \leftarrow a[j]$
      end for
    end if
  end if
end for

algorithm $a[1..N]$ stores the phrase number in $a[i]$ to which word *i* belongs in the pivotal language. Pivotal language words $a[j], ..., a[j + k]$ are put in the same group if in another language they are aligned with the same word. $t[l, i] := k$ stores information regarding the ith word in language l is aligned with the kth word in the pivotal language.

Later the PL words belonging to a group are merged to form a phrase and so are the corresponding mapped words in non-PL languages. These give rise to the much talked about multilingual proto-synsets, the building blocks of a WordNet like structure.

The process of generation of phrases is deterministic and hence not prone to errors. Essentially the quality of output is totally dependent on the quality of input, or the output of GIZA++ which as has already been discussed is highly error prone and given to mis-alignments.

Due to variations in language structures and semantics there are cases where a word in one language aligns with two or more words in another language or none at all. If there were perfect one-to-one alignments the whole process of aggregation of words into phrases would have been redundant. Table 4.10 summarizes our findings related to how many English words are aligned with a word in non-pivotal language. Even though a large number of alignments are 1:1, there are cases a word in non-pivotal language aligns with more than one English word. German has the fewest number of 1:1 word alignments, 85.7% among the languages chosen, probably indicating the existence of compound words in it.

Numbre of Aligned Words / Language	1	2	3	4	5	6	7	8	9
German	85.7	7.9	2.5	1.1	0.7	0.6	0.5	0.3	0.6
French	91.6	4.9	1.5	0.6	0.4	0.3	0.2	0.1	0.3
Greek	87.4	7.8	2.2	0.9	0.5	0.4	0.2	0.2	0.3

Table 4.10: Proportion (in [%]) of phrases of a given length aligned with each word in the non-pivotal language

A 1:N alignment suggests that the group of N words forms a phrase for which there exists a unitary lexical item in another language, and therefore that phrase is likely to represent a well-defined and potentially useful semantic concept. When L words in English are aligned with M words in one non-pivotal language L_1, and N words in another non-pivotal language L_2, then the phrase aggregation algorithm step will pair 2 phrases $M_{L_1} : N_{L_2}$ with the phrase created by collating L words in English.

Initially all words are assumed to be phrases on their own. Gradually we start grouping them into phrases based on the fact that how other languages see the formation of these groups in English, the pivotal language. For instance, if a word in German is aligned with more than one word in English that probably is the case where the English words should be merged to form a phrase. The algorithm decides which English words need to be made part of the same group. The words that should belong to the same group are assigned the same group number, but that decision is dependent on the other, non-pivotal, languages as well. Let us make a run through the algorithm for a few words just to demonstrate how it works. For the aligned words that would be considered for this example, please see Figure 4.2.

Figure 4.2 shows the aligned words in pivotal and non-pivotal languages in the form of a graph, where an edge indicates the fact that its vertices, in the form of words, have been word aligned by GIZA++. Thus, *Friday* in English is aligned with *freitag* in German, *vendredi* in French and $\pi\alpha\rho\alpha\sigma\kappa\epsilon\nu\acute{\eta}$ in Greek. Some words in one language are aligned with more than one word in another. For instance, *vendredi* in French is aligned with both *on* and *Friday* in English.

This example is taken from the actual proto-synsets that were created. Before forming phrases we make a list of words in English and assign them a unique ID and a unique number, what we call *Number*, for the sake of simplicity. The IDs

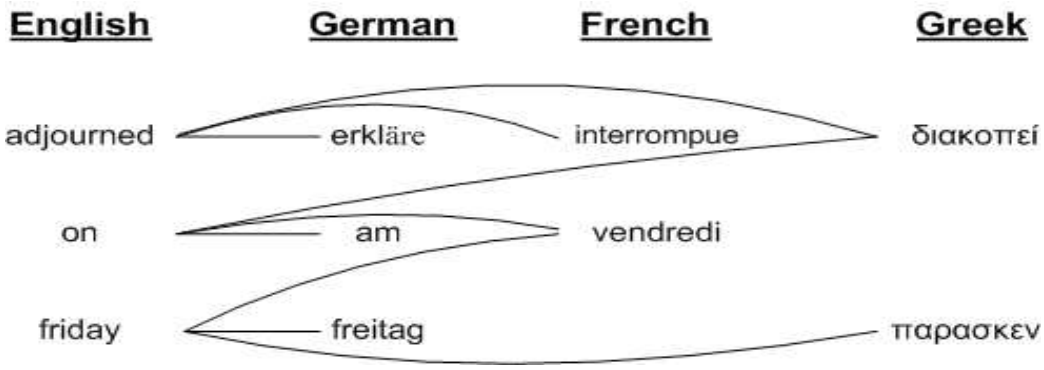


Figure 4.2: Graphical representation of aligned words in the pivotal and non-pivotal languages (cf. Table 4.23).

do not remain unique with time and refer to the group number to which that word belongs. Let us suppose we get the following for English (Table 4.11):

14 adjourned 14
15 on 15
16 friday 16

Table 4.11: Snapshot of the English words, with their *IDs* (group numbers) and *Num* values

Subsequently we make a list of words in each of the non-pivotal languages and assign them the unique IDs. The first non-pivotal language in this case is German, and we have the following information for German based on the results of word alignment (Table 4.12).

erkläre 14
am 15
freitag 16

Table 4.12: Corresponding German words and their *Num* values

Here we have a word in German followed by numbers signifying the number of English words in the corpus with which it is aligned. Thus *erkläre* is aligned

with word numbered 14 in the English corpus. Where the first word in the English corpus is assigned number 1, the second number 2 and so on. *erkläre* is word aligned with the English word *adjourned* which has the index number 14. We would put information in the table for German as (Table 4.13):

14 *erkläre*

Table 4.13: Example of a German word aligned with more than one English word that are not consecutive

followed by two more entries (Table 4.14):

15 am
16 freitag

Table 4.14: The German entries following the ones in Table 4.13

Since none of the German words carry any information regarding creation of phrases, the IDs of the English words remain the same. If we were dealing with only two languages, we would have called it a day and each English word would have been put against the German word as such, to give the following (Table 4.15):

adjourned *erkläre*
on am
friday freitag

Table 4.15: Multilingual proto-synsets in the absence of French and Greek word-aligned corpora

But we are dealing with not one but four languages here, each of which carries some extra information which could be exploited to refine our search for existence of useful phrases.

The next language to look at is French and it has got the following information (Table 4.16):

interrompue 14
le
vendredi 15 16

Table 4.16: Corresponding entries for French for the generation of phrases

As can be seen we have three French words here among which the word *le* is aligned with no English word and is hence not followed by the index of the aligned English word. *interrompue* is aligned with the English word with the index number 14 which as can be seen above is *adjourned*. *vendredi* on the other hand is aligned with two English words *on* and *Friday* and they are also consecutive to each other. Now that is some useful information and we exploit it. The information that it carries is that the two English words *on* and *Friday* could be merged to form a phrase *on Friday*, but we are not yet done. So rather than straight away creating the phrase we store this information for future reference by assigning the same group number *15* to each of the two English words, and they now look like (Table 4.17):

14 adjourned 14
15 on 15
15 friday 16

Table 4.17: English group numbers after reading through information about English-French word alignment

As for French we keep the following information in its table (Table 4.18):

14 interrompue
15 vendredi

Table 4.18: Corresponding information for French

Next, we have Greek. Looking at it we figure that we have got the following information Table 4.19):

$\delta\iota\alpha\kappa o\pi\epsilon\acute{\iota}$ 14 15
$\tau\eta\nu$
$\pi\alpha\rho\alpha\sigma\kappa\epsilon\upsilon\acute{\eta}$ 16

Table 4.19: Entries for Greek for English-Greek word alignment

In this case the first Greek word $\delta\iota\alpha\kappa o\pi\epsilon\acute{\iota}$ is aligned with two consecutive English words indexed 14 and 15 respectively. It implies that all words in English with index number (ID) 15 should get a new index (ID) of 14, and we end up with the following in the English table (Table 4.20):

14 adjourned 14
14 on 15
14 friday 16

Table 4.20: Revised group number information for English words after encountering information in Greek

The second Greek word $\pi\alpha\rho\alpha\sigma\kappa\epsilon\upsilon\acute{\eta}$ does not convey much information about grouping of words and thus has no bearing on the IDs of the English words. The Greek table now look like (Table 4.21):

14 $\delta\iota\alpha\kappa o\pi\epsilon\acute{\iota}$
16 $\pi\alpha\rho\alpha\sigma\kappa\epsilon\upsilon\acute{\eta}$

Table 4.21: Greek table after changing group number information in English

As can be seen since $\delta\iota\alpha\kappa o\pi\epsilon\acute{\iota}$ has been aligned with two consecutive English words, one index number (15) is skipped.

Table 4.22 gives the phrase group information indicating how the words in all the different languages could be collated to form phrases. Table 4.23 gives the final phrases generated.

That is what we call the multilingual proto-synset or proto-synset for short, since it is not in a very refined form. Refinements can be made to such proto-

English ID	English Word	English Num	German ID	German Word	French ID	French Word	Greek ID	Greek Word
14	adjourned	14	14	erkläre	14	interrompue	14	διακοπεί
14	on	15	15	am	15	vendredi	15	
14	friday	16	16	freitag	16		16	παρασκευή

Table 4.22: Phrase group information for final generation of phrases.

English Phrase	German Phrase	French Phrase	Greek Phrase
adjourned on friday	erkläre am freitag	interrompue vendredi	διακοπεί παρασκευή

Table 4.23: The final phrases generated.

synsets to create synsets and different relationships can be defined between them to create a WordNet like lexical resource.

As a result of running the algorithm we were able to create more than 1.5 million proto-synsets.

4.5.1 Author's Note

The process of creating proto-synsets in such a manner is deterministic and is not given to errors if implemented properly. The quality of output is only dependent on the quality of input, in this case the pair-wise alignments as given by GIZA++.

4.6 Disambiguation

The proto-synsets thus created can be used for other NLP tasks, such as Word Sense Disambiguation (WSD), and Multilingual Information Retrieval (IR). In order to evaluate the results we used them to sense tag the original English corpus and then to evaluate it by measuring whether using this additional information helps in the classification of documents by improving the quality of clustering.

Tagging the original English corpus with multilingual sense tags is a straightforward process. As can be seen from Figure 4.3, if we start reading down a language column, e.g. English, we are in fact reading through the corpus. Thus the English corpus reads like "resumption of the session ...". Thus if we assign any tags to any of the proto-synsets we are essentially assigning sense tags to phrases or words in the original corpus and thus the corpus would be sense tagged, or in other words WSD is being performed. Importantly, the process has the potential of providing all words in a given language corpus with a multi-lingual proto-synset.

Thus we devised a very simple, even if time consuming, process of WSD by assigning the same sense tag to the same proto-synset, wherever it occurs. Since a number of proto-synsets will appear repeatedly, they will get the same sense tag throughout which we will take to suggest that the same word/phrase is used in the same sense in all these cases. Figure 4.4 gives a snapshot of the proto-synsets with the corresponding indices assigned.

4.7 Evaluation

This is an example of cross-lingual WSD where translations of a word or phrase in a non-pivotal language can help us in narrowing down the sense of a word

Num	English	German	French	Greek
1	resumption of	wiederaufnahme	reprise de	επανάληψη της
3	the	der	la	
4	session	sitzungsperiode	session	συνσδου
5	i declare	ich erkläre	je déclare	κηρύσσω
7	resumed	wiederaufgenommen	reprise	επανάληψη
8	the		la	
9	session	sitzungsperiode	session	συνόδου
10	of	des	du	του
11	the	die		
12	european	europäischen	européen	ευρωπαϊκού
13	parliament	parlaments	parlement	κοινοβουλίου
14	adjourned on friday	erkläre am freitag	interrompue vendredi	διακοπτεί παρασκευή
17	17	17.	17	17
18	december 1999	dezember unterbrochene	décembre dernier	δεκεμβρίου
20	,			
21	and		et	και
22	i would	wünsche	je vous	απευθύνω θερμές
23	like once again	nochmals	renouvelle	ξανά
27	to			να
28	wish you	wünsche ihnen	vux	περάσατε
30	a happy	jahreswechsel	vux	περάσατε
31	new year		renouvelle	
34	in		en	
35	the			
36	hope	hoffe	espérant	ελπίζοντας
37	that	daß	que	
38	you enjoyed	sie schöne	vous renouvelle	περάσατε
40	a			
41	pleasant festive period	ferien	vacances renouvelle vacance	περάσατε διακοπές
44	.	.	.	.
45	(	(	(	(
46	the	das	le	
47	house	parlament	parlement	σώμα
48	rose and observed	erhebt	debout observe	όρθιο τηρεί
51	a	einer	une	ενός
52	minute '	schweigeminute	minute silence	λεπτού σιγή
53	s silence			
56	)	)	)	)
57	yes	ja	oui	ναι
58	,	,	,	,
59	mr	herr	monsieur	κύριε

Figure 4.3: A sample snapshot of the database of proto-synsets.

or phrase in the pivotal language. It has been done in an unsupervised manner, meaning thereby that we do not need a sense annotated corpus to begin with. That is a big advantage over supervised techniques in the sense that any annotated corpus would require a lot of investment in terms of both time and money, and to make sure that manual annotations are what they should be, or in other words reaching the consensus amongst the annotators.

4.7.1 Baseline Comparison for Extraction of Multilingual Synsets

The main idea in our thesis is to generate phrase-based multilingual synsets and to evaluate their potential benefits. Previously people have only generated the single word multilingual lexicons from the word aligned data (Fišer 2007) and (Sagot & Fišer 2008).

Num	English	German	French	Greek	Index
1	resumption of	wiederaufnahme	reprise de	επανάληψη της	1
3	the	der	la		3
4	session	sitzungsperiode	session	συνσδου	4
5	i declare	ich erkläre	je déclare	κηρύσσω	5
7	resumed	wiederaufgenommen	reprise	επανάληψη	7
8	the		la		8
9	session	sitzungsperiode	session	συνόδου	9
10	of	des	du	του	10
11	the	die			11
12	european	europäischen	européen	ευρωπαϊκού	12
13	parliament	parlaments	parlement	κοινοβουλίου	13
14	adjourned on friday	erkläre am freitag	interrompue vendredi	διακοπεί παρασκευή	14
17	17	17.	17	17	17
18	december 1999	dezember unterbrochene	décembre dernier	δεκεμβρίου	18
20	,				20
21	and		et	και	21
22	i would	wünsche	je vous	απευθύνω θερμές	22
23	like once again	nochmals	renouvelle	ξανά	23
27	to			να	27
28	wish you	wünsche ihnen	vux	περάσατε	28
30	a happy	jahreswechsel	vux	περάσατε	30
31	new year		renouvelle		31
34	in		en		34
35	the				35
36	hope	hoffe	espérant	ελπίζοντας	36
37	that	daß	que		37
38	you enjoyed	sie schöne	vous renouvelle	περάσατε	38
40	a				40
41	pleasant festive period	ferien	vacances renouvelle vacances	περάσατε διακοπές	41
44	.	.	.	.	44
45	(	(	(	(	45
46	the	das	le		46
47	house	parlament	parlement	σώμα	47
48	rose and observed	erhebt	debout observe	όρθιο τηρεί	48
51	a	einer	une	ενός	51
52	minute '	schweigeminute	minute silence	λεπτού σιγή	52
53	s silence				53
56	)	)	)	)	56
57	yes	ja	oui	ναι	57
58					58

Figure 4.4: A sample of indexed proto-synsets: snapshot from the database.

For the purposes of baseline evaluation of extracted synsets we did what other people have done and generated only the word-based multilingual lexicon and compared it with our synsets.

4.7.1.1 Experimental Design

In the word alignments that GIZA++ (Och & Ney 2003) comes up with we have many words in a non-pivotal language (language other than English) which align with N words in the pivotal language (English in our case), or in other words they have $1 : N$ word mapping. We used this information to generate the phrases. Lets suppose we skip this step and generate only one-word based multilingual lexicon.

For the purposes of illustration we have assumed the case where we have

three languages. English is the pivotal language, and German and French are non-pivotal languages. Figure 4.5 shows the 1:N mapping between them as a result of word alignment.

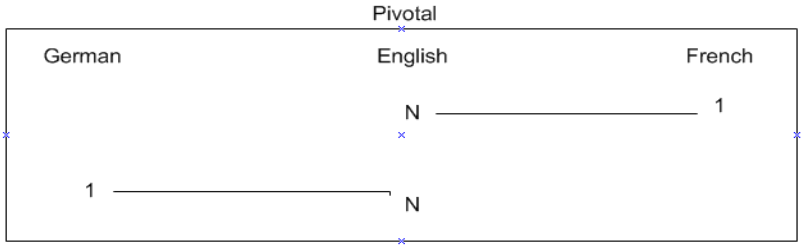


Figure 4.5: 1:N mappings between the pivotal language, English, and non-pivotal languages, German and French.

To make it simpler, let us assume we have three words in English, the first two are aligned with a word in French and the last two are aligned with a word in German, as shown in Figure 4.6.

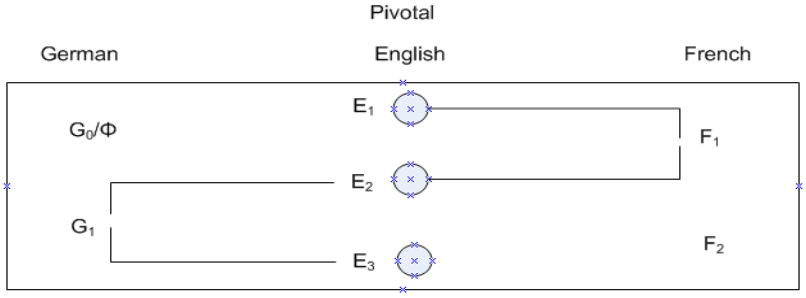


Figure 4.6: Alignment of words in English, German, and French.

As a result of this proposed alignments we get the phrase-to-phrase alignments as in Figure 4.7 and word-to-word alignments as in Figure 4.8.

Now even in our case there are cases where there is 1:1 word alignment between a word in a non-pivotal language and a word in the pivotal language. In such cases synsets produced have exactly one word in each language. These cases must also occur if we do not generate phrases. Thus, there is a certain

German phrase	English phrase	French phrase
$G_0/\Phi + G_1$	$E_1 + E_2 + E_3$	$F_1 + F_2/\Phi$

Figure 4.7: Phrases formed as in our synsets.

German word	English word	French word
G_0/Φ	E_1	F_1
G_1	E_2	F_1
G_1	E_3	F_2/Φ

Figure 4.8: Words put together without forming phrases.

amount of overlap between our synsets and word-based multilingual lexicon generated as a result of not collating words into phrases. It is depicted in Figure 4.9.

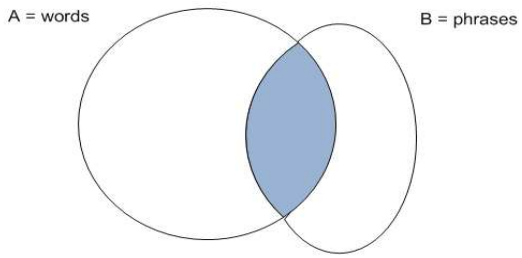


Figure 4.9: Words put together without forming phrases.

We are interested in seeing whether taking out the shared synsets and take sets A and B without them, are there many more one-word synsets than phrase-based synsets? In other words we are interested in knowing if $|B \setminus \{A \cap B\}| < |A \setminus \{A \cap B\}|$? The results are given in Table 4.24.

Number of phrase-based synsets	1576888
Number of word-based synsets	1817018
Number of common synsets	1312027
Number of phrase-based synsets that are not common	264861
Number of word-based synsets that are not common	504991
Ratio of Common Synsets to the size of phrase-based synsets	0.83
Ratio of Common Synsets to the size of word-based synsets	0.72

Table 4.24: Results of Baseline Comparison

As can be observed there are more word-based synsets (504991) that are not in the intersection set as compared to the phrase-based synsets (264861), which is ratio of 1.91. It gives us the proportion of phrase-based synsets (16.8%) which are not created by the other approach where they only consider 1:1 word alignments.

It shows the benefits of our approach to the baseline approach, as adopted by other people, in terms of number of entries. Our approach produces the genuine phrases from the aligned data, which would not exist if the words were not aligned in a particular fashion by GIZA++. Since, GIZA++ produced a number of 1:N word alignments they should be combined into phrases as done by our approach but other approaches have confined themselves to only considering 1:1 word alignments.

4.7.2 Issues with Evaluation

The intuitive way of evaluation would be to compare the results with a semantically disambiguated parallel corpora. We have parallel corpora and semantically disambiguated corpus in one language[3], but not both. That lack of such semantically tagged parallel corpora, leaves us with the option of carrying out an eval-

[3]http://www.cse.unt.edu/ rada/downloads.html#semcor

uation in an indirect fashion, where the results themselves are not compared to any gold standard but their impact on some NLP/IR application is gauged, and it is ascertained whether it improves the performance or not.

One such application is document clustering, where we cluster documents based on some similarity metric. The advantage of using document clustering is that it is also unsupervised and the original corpus need not be annotated with any class information.

4.7.3 Using Clustering for Evaluation

We have used Hierarchical Agglomerative Clustering (HAC) for clustering documents. In HAC initially each node, a document in our case, is a cluster which are progressively merged to form one final cluster that sits at the top, and hence it is the bottom-up approach. As a result a binary tree is generated which can be traversed from top to bottom to get the desired number of clusters (Shahid & Kazakov 2011).

In order to evaluate how good the clustering is, a number of metrics can be used, such as the Davies-Bouldin Index (DBI) (Davies & Bouldin 1979), Normalized Mutual Information (Kvålseth 1987), and the Gini Index (Breiman et al. 1984), which can be used to determine *Cluster Purity* (Alfred et al. 2007).

DBI is a measure of clustering dispersion, taking into account both within cluster distances as well as distances between clusters to judge the quality of clustering. That is an *internal criterion* for measuring the quality of clustering, as it does not make use of external gold standard data set for evaluation. Good scores on that account do not necessarily translate into good scores on effectiveness or performance in an application specific task.

In an application specific task, clustering is compared against the gold standard classes. That is the *external criterion* for measuring the quality of cluster-

ing. Both Normalized Mutual Information and Cluster Purity (or Impurity) are external measures of measuring clustering quality.

Among the external criteria, measuring cluster purity (or Impurity) is a simple and easy approach. It suffers from the fact that purity is highest when each document is assigned to its own cluster and hence is not a good option to trade off cluster quality with the number of clusters.

Mutual Information (MI) on its own suffers from the same problem. It reaches a minimum, 0, when the knowledge about a document being in a particular cluster does not convey any information about which class it might belong to. It reaches its maximum when the clusters exactly mirror the classes, but also when we have one-document clusters. So it suffers from the same problem as does purity.

Normalized Mutual Information (NMI) of the set of clusters Ω and the set of classes $\mathbb{C}$, takes care of that by using the concept of *Entropy* from Information Theory which tends to increase with the number of clusters and is in the denominator of the equation, as given below:

$$NMI(\Omega, \mathbb{C}) = \frac{I(\Omega; \mathbb{C})}{[H(\Omega) + H(\mathbb{C})]/2} \tag{4.1}$$

where $I(\Omega; \mathbb{C})$ is mutual information, $\Omega = \{w_1, w_2, ..., w_k\}$ is the set of clusters, $\mathbb{C} = \{c_1, c_2, ..., c_j\}$ is the set of classes, and H denotes entropy (Equation 4.2). Entropies are measured for both the set of clusters $H(\Omega)$ and the set of classes $H(\mathbb{C})$.

$$H(\Omega) = -\sum_k P(w_k) log P(w_k) \tag{4.2}$$

$$H(\Omega) = -\sum_k \frac{|w_k|}{N} log \frac{|w_k|}{N} \tag{4.3}$$

Equation 4.2 gives the equation for Entropy for the set of clusters where $P(w_k)$ is the probability of a document being in the cluster w_k. Equation 4.3 is equivalent to equation 4.2 with estimates of the probability, where N is the total number of documents in all the clusters.

Entropy is a measure of disorder, and is maximum when the classes are maximally intermixed in the clusters, and minimum when the clusters have homogenous content or in other words the documents in a cluster belong to the same class. It also increases with the number of clusters and reaches its maximum when there is one document per cluster. In equation 4.1 the denominator ensures that NMI is low when every document is a cluster on its own.

In our case once we have clustered data we convert the hierarchical clustering binary tree, dendrogram, into a 'flat' set of clusters of a predefined size, equal to the number of clusters in the gold standard. Since we never reach a point where we end up with one document per cluster, hence, purity (or Impurity) is used as the measure of clustering quality, which is easy to measure.

Impurity measures the overall cluster quality. Mathematically it is composed of the Gini Index (GI), which is calculated for each cluster separately. Once the GI values are calculated for all the clusters, overall impurity is calculated.

The GI gives how diverse a cluster is in terms of the classes to which its documents belong to. If all of them belong to the same class then the value of GI is 0. If all of them belong to separate classes, then it approaches 1 asymptotically in the number of classes. Mathematically it is defined as below:

$$GiniC_k = 1.0 - \sum_{c=1}^{n} \left(\frac{P_{kc}}{N_k} \right)^2 \tag{4.4}$$

The GI is calculated for each cluster separately and gives an indication of how good a split it provides in terms of cluster homogeneity. In the equation above k is a particular cluster, n is the number of classes, P_{kc} denotes the number of

documents in cluster k belonging to class c. The GI only measures the level of purity for one cluster. In order to judge the quality of clustering, we need to take it to the next level and define a parameter that measures the level of purity for the complete set of clusters. Impurity (Alfred et al. 2007) is one such measure.

The size of a cluster may have a disproportionate effect on the GI, since it is not normalized. To measure the effect of GI of one particular cluster on overall clustering we need to normalize it so that the size of a cluster will not affect the impurity measure for the entire clustering.

The impurity measure is a weighted sum of Gini Indices. In the impurity measure we multiply the gini of a particular cluster by its size and sum it over the number of clusters. Then we divide it by the total number of documents in the data set to get some measure for the clustering. The idea is to minimize impurity or in other words have clusters with more homogeneity within individual clusters and where clusters have less in common. The equation below formalizes this concept in mathematical terms.

$$Impurity = \frac{\sum_{k=1}^{K} T_{C_k}.GiniC_k}{N} \tag{4.5}$$

In the equation above we have K as the number of clusters, T_{C_k} as the number of documents in a particular cluster and N as the number of documents in the data set.

4.7.3.1 Experiments

Before we cluster any data we needed to decide what really to cluster. We had the original Europarl corpus, which we treated as the gold standard since it is manually created corpora by quality human translators who translate speeches in the European Parliament into the official languages.

In the original English corpus, parliamentary proceedings are divided into

chapters, or topics of discussion, and speeches by individual members of European Parliament. Thus under each chapter, topic of discussion, there are a number of speeches by different individuals. We consider each complete individual speech as a document.

The above documents are fed into the clustering tool which then performs stop-word removal and stemming before figuring out the clusters. Stop-word removal helps in reducing the number of dimensions and hence speeds up the process.

For the purposes of this work we used an in-built clustering tool. When we cluster this original corpus, we put each document into a cluster based on what the clusterer deems fit using the hierarchical agglomerative clustering and parameters: complete link as a link method, which favours compact clusters (Strehl 2002), and Euclidean distance as the distance metric.

These clusters and the documents within would be used to evaluate the performance of the WSD task when sentences are either replaced with their sense tags or are followed by them. Such comparison in terms of purity of clustering gives an indication whether the clustering improved after assigning the sense tags, or it actually deteriorated. Improving purity would suggest that the sense tags, or multilingual synsets, are indeed useful in removing ambiguity.

The original English corpus is in the form of individual speeches followed by the speaker and chapter (class) tags. Below is an example of a document from the said corpus:

> i thank the president-in-office but i do not agree with him that the purpose of nato ' s intervention was to stop a humanitarian disaster , because we have a humanitarian disaster in kosovo today , we have the hell of milosevic , followed by the hell of the kla and the nato forces . if the president-in-office has noticed , almost half the

> questions he has had to answer today relate to kosovo . despite this , he too accepted that the inquisition procedure of the war crimes tribunal against nato after applications by canadian and other pacifist organisations is an open question . ⟨speaker id= 1167 language="el" name="alavanos" ⟩ ⟨chapter id= 59 ⟩

One can see that it is a speech by someone named Alavanos who is originally a Greek speaker as told by his language information which is *el* and he is speaking under topic of discussion *question time (council)* which is identified by its unique ID of 59.

We clustered only the first 1,000 such documents due to complexity of the hierarchical clustering algorithm. We then measured the Gini index and the impurity measure in order to gauge the quality of clustering.

The purity measure for the original corpus does not convey much information on its own. So it has to be compared with the same measures for the sense tagged corpus.

We clustered documents as represented by their sense tags only and also the original sentences followed by sense tags. For the same document as above the two versions are as shown below:

> @32178 @3838 @171280 @4536 @173455 @1562 @173458 @87793 @291 @1010 @173463 @798 @173465 @7348 @173467 @173468 @27 @173471 @4176 @173473 @2588 @58 @30463 @5109 @4176 @173473 @2588 @5404 @34394 @173484 @61 @173486 @538 @173489 @9302 @173491 @308 @173493 @4432 @173489 @798 @1359 @173499 @326 @296 @173502 @44 @173505 @3065 @56958 @19916 @173509 @61 @173511 @173512 @296 @439 @173515 @632 @173519 @49336 @173521 @115 @34394

@44 @173525 @173526 @61 @173528 @173529 @2298 @8099 @173533 @589 @35 @173290 @173538 @173540 @34379 @173542 @173544 @173545 @326 @35914 @173548 @7460 @173551 @173552 @44

i thank the president-in-office but i do not agree with him that the purpose of nato ' s intervention was to stop a humanitarian disaster , because we have a humanitarian disaster in kosovo today , we have the hell of milosevic , followed by the hell of the kla and the nato forces . if the president-in-office has noticed , almost half the questions he has had to answer today relate to kosovo . despite this , he too accepted that the inquisition procedure of the war crimes tribunal against nato after applications by canadian and other pacifist organisations is an open question . @32178 @3838 @171280 @4536 @173455 @1562 @173458 @87793 @291 @1010 @173463 @798 @173465 @7348 @173467 @173468 @27 @173471 @4176 @173473 @2588 @58 @30463 @5109 @4176 @173473 @2588 @5404 @34394 @173484 @61 @173486 @538 @173489 @9302 @173491 @308 @173493 @4432 @173489 @798 @1359 @173499 @326 @296 @173502 @44 @173505 @3065 @56958 @19916 @173509 @61 @173511 @173512 @296 @439 @173515 @632 @173519 @49336 @173521 @115 @34394 @44 @173525 @173526 @61 @173528 @173529 @2298 @8099 @173533 @589 @35 @173290 @173538 @173540 @34379 @173542 @173544 @173545 @326 @35914 @173548 @7460 @173551 @173552 @44

Thus if we start traversing the tree from top-down we can decide at which level to stop if we know the number of classes in the gold standard that we are dealing with. For our specific purposes we have 59 classes which cover 1,000

documents that we have clustered, with and without the sense tags. Thus we only need to go down the tree enough that we have 59 clusters and then we figure out the documents in each cluster to calculate the gini index and the impurity measure to see whether clustering improved after annotating the text with the sense tags.

The clustering tool generates an output that gives information about how the binary tree is structured. That gives useful information to be exploited for the purposes of figuring out which documents lie in which clusters.

Table 4.25 shows an example of the output of the clustering tool.

$[1998; 1.414; (1997, 1995); (a45 : 0.012, a56 : 0.012, a58 : 0.010, a1605 : 0.010, commission : 0.010)]$
$[1997; 1.414; (1996, 1992); (a56 : 0.058, a45 : 0.058, a1605 : 0.053, a1104 : 0.038, 1999 : 0.033)]$
$[1995; 1.414; (1994, 1990); (a20 : 0.011, commission : 0.011, a58 : 0.010, european : 0.010, a326 : 0.010)]$
$[1996; 1.414; (1993, 1986); (a1104 : 0.073, a2654 : 0.059, a2657 : 0.057, a2655 : 0.056, tomorrow : 0.053)]$
$[1992; 1.414; (1989, 1985); (a56 : 0.096, a45 : 0.095, a1605 : 0.092, 1999 : 0.065, a1606 : 0.063)]$
$[1994; 1.414; (1991, 1987); (a20 : 0.012, a58 : 0.012, commission : 0.011, european : 0.011, a326 : 0.011)]$
$[1990; 1.414; (1984, 1977); (thank : 0.014, debate : 0.011, question : 0.011, item : 0.010, a535 : 0.010)]$

Table 4.25: Output of the clustering tool.

In the snapshot above of the output of the clustering tool, the first piece of information is regarding the cluster number. So in the first line *1998* is the cluster number, which has been obtained by merging two clusters *1997* and *1995*. The second parameter is the height of that node, and is the distance between its children. The rest of the information contains the top five words ordered with respect to their tf-idf weights. Using the information above the binary tree structures were created that were later traversed to get the requisite number of clusters. Depth First Search (DFS) was then used to figure out documents in each of the clusters.

While traversing through the tree to look for clusters and the documents within, we need to defined before hand how many clusters are we looking for. In our particular case we defined the number of clusters as 59 which was the number of chapters (or topics of discussion) for the 1,000 documents that we clustered. We look through the tree, keep expanding and adding clusters to a list. We always expand the node with the highest index first. So in the snapshot above, we will expand 1998, 1997, 1996, 1995, 1994 and so forth in order. That makes sense since newer clusters are formed at greater height of the tree and are traversed first going from the root to the leaves.

Once a list contains the requisite number of clusters it is time to explore the clusters for the documents they contain. It is fairly straightforward. We look at the subtrees rooted at the nodes (clusters) in the list, perform Depth-First-Search (DFS) till we reach the leaf nodes. For each node in the list we output the leaf nodes in the subtree rooted at that node. These leaf nodes are the documents which were assigned to individual clusters in the beginning. Thus, we create a list of clusters and the documents within.

Such a list comes in handy while calculating parameters such as the Gini index (Alfred et al. 2007) and the impurity measure to gauge the *goodness* of

clustering obtained. If the impurity measure improves after clustering the sentences with the synsets it implies that probably sense tags are useful in reducing the inherent ambiguity in the language being dealt with.

Measuring impurity for the three cases where we clustered the original English corpus, where we clustered only the sense tags and where we clustered the English sentences followed by their representative sense tags would give us a clue as to whether the sense tags help improve the clustering or not (see Table 4.26).

We expected the clustering to have improved after assigning the sense tags but impurity seems to have deteriorated, though slightly, when we cluster the sentences along with their sense tags, and is totally way off when we clustered only the sense tags. The second result is more intuitive since having only the sense tags does not build any relationship with the sentences that they are assigned to. But the expectation was that assigning the sense tags would improve clustering.

In the implemented version (Algorithm 4) we have also taken care of the case where a word in a non-pivotal language is aligned with more than consecutive series of words in English. Thus if a word is aligned with words with IDs j, $j+1$, ..., $j+z$, and another series of consecutive words with IDs k, $k+1$, ..., $k+zz$, then the second series of consecutive IDs will all be assigned the value k, which is the first one among the series, as the first series will all get j.

Parameter	Original Corpus	Sense Annotated Corpus	Sense Tags Only
Impurity	0.784	0.806	0.874

Table 4.26: Impurity measures for different scenarios used for evaluation

Algorithm 4 Step in Algorithm 3 where a word in non-pivotal language is aligned with more than one consecutive word in the pivotal language.

```
if If the word in the non-pivotal language is aligned with more than one consecutive word in English then
    if it is the first series of consecutive alignments then
        assign the first position, in the sequence of consecutive positions, to all the consecutive positions
    else
        if if it is not the first series of consecutive alignments then
            assign the corresponding first position, in this sequence of consecutive positions, to all the consecutive positions
        end if
    end if
end if
```

4.7.4 Discussion

Still the results of evaluation using document clustering appear to be inconclusive and need further investigation. Clustering is unsupervised as the chapter ID tags have only been used for evaluation and not for the clustering itself. There are several reasons why using these tags as class labels is far from ideal. The individual speeches are quite short, with an average length of around 175 words per speech for the first thousand speeches, which does not allow a common theme and vocabulory to become clearly established. There is also the issue of the differences in wording and style emphasised by the different linguistic background of the speakers. Still, this was the best resource available.

Another consideration was made at the time of a potentially important realization namely that the ‘Chapter ID’ tags could be used as class labels indicating the topic discussed in all speeches included in that chapter.

These class labels could then be used to train and evaluate a classifier. This supervised machine learning setup could then be used to evaluate the relative benefits of multilingual proto-synstes when used as additional features (attributes)

for the texts in question.

4.7.4.1 Why the Tags?

Tags are not ideal for a class label, since:

- they are not systematically chosen.
- individual differences between speakers maybe greater than those between topics (a.k.a. chapters).

Yet, there is no other auxiliary information in the corpus that could be used for supervised learning and supervised learning offers an alternative form of evaluation.

4.7.4.2 Which Machine Learning Approach to Use?

Decision Trees provide a good alternative to carry out evaluation using a supervised learning approach, as they are easy to use, they are fast, and straightforward. They take a table of attributes, same as in the case of IR where a doc is a bag of words.

4.7.5 Using Decision Trees for Evaluation

Decision tree is a supervised learning technique that builds a tree where each node represents an attribute which splits the data set. The learning algorithm decides which attribute to put at a particular node based on principles of information theory.

Entropy is a measure that (Mitchell 1997) measures the homogeneity of a set of examples. Information Gain measures the expected reduction in entropy. The aim is to split the set of examples at a particular branch of a tree so that along one branch all examples belong to one class.

The measures of Entropy and Information Gain are given as below:

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i \tag{4.6}$$

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{4.7}$$

Equation 4.6 is a generalized version of equation 4.2, which gives a clustering specific definition of it.

In the equations above S is a collection of examples with c different possible classifications. p_i is the proportion of examples belonging to class i. In the formula for Information Gain S_v is the subset of examples which belong to class v. The first term is the original entropy and the second term is the expected value of entropy after the data is partitioned using attribute A.

Most algorithms that have been employed to learn decision trees are based on a top-down greedy approach ID3 (Quinlan 1986) and one of its successors C4.5 (Quinlan 1993). CART (Breiman et al. 1984) is another version of decision trees where binary trees are constructed for decision making.

Algorithm 5 gives description of an algorithm based on ID3 (Mitchell 1997):

For the construction of decision trees attributes need to be defined which could be either discrete or continuous valued. In this case the attributes are the words which are real-valued, taking on TF-IDF values.

For the purposes of evaluation we took a subset of the original data such that each class had equal number of instances, yielding 37 classes with 10 instances each. Then TF-IDF values were ascertained for each word in each document. Then we created a *document-term* matrix (see Figure 4.10) where each column is an attribute (term or word) and each row is the document. Each cell in the

matrix is the TF-IDF value of the term (identified by the column) in the document (identified by the row).

This TF-IDF matrix is then fed to Weka[4], a tool for data mining with Java[5] implementation of Decision Trees, named as J48, which is a free version of C4.5. It takes in the term-document matrix above with class information and gives the percentage of correctly classified instances.

4.7.5.1 Experimental Design

We created two term-document matrices, one for the original corpus without the sense tags and the other with the sense tags. They were fed into Weka and 10-fold cross validation was performed.

[4]http://www.cs.waikato.ac.nz/ml/weka/
[5]http://www.java.com/en/

Algorithm 5 ID3$(Examples, Target_attribute, Attributes)$

Create a *Root* node for the tree
If all *Examples* are positive, Return the single-node tree *Root*, with label = +
If all *Examples* are negative, Return the single-node tree *Root*, with label = -
if *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
Otherwise Begin
  A ← the attribute from Attributes that best* classifies *Examples*
  The decision attribute for *Root* ← A
  For each possible value, v_i, of A,
    Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    If $Examples_{v_i}$ is empty
      Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
      Else below this new branch add the subtree
        ID3$(Examples_{v_i}, Target_attributes, Attributes - A)$
End
Return *Root*
{* The best attribute is the one with the highest} *information gain*

	i	declare	resumed	the	session	of	european	parliament	adjourned	on	Friday
doc 1	0.00142417	0.008587772	0.009728037	0.000198595	0.006780494	0.00015857	0.001289072	0.001733379	0.009728037	0.000549075	0.007447507
doc 2	0.002237982	0	0	0.000156039	0	0.000249181	0	0	0	0	0
doc 3	0.001798952	0	0	0	0	0.000200299	0	0	0	0.001387137	0
doc 4	0	0	0	0.000347213	0	0.000554469	0	0	0	0.000959973	0
doc 5	0	0	0	0.000109838	0	0.000233868	0.000950601	0.001278246	0	0	0
doc 6	0.001446081	0	0	0.000100825	0	0	0	0	0	0	0
doc 7	0.003298079	0	0	0.000114976	0	0.000367214	0	0	0	0	0
doc 8	0.002211653	0	0	0.000154203	0	0	0	0	0	0.001705363	0
doc 9	0.002152563	0	0	0.000150083	0	0.00015978	0	0	0	0.000553267	0
doc 10	0.000693692	0	0	0.000169282	0	7.72E-05	0	0	0	0.001069785	0
doc 11	0.000269327	0	0	0.000244118	0	0.000599749	0.000487557	0	0	0.000415346	0
doc 12	0	0	0	0.00022488	0	0.000410416	0.000834105	0	0	0	0
doc 13	0.000939952	0	0	0.000131073	0	0	0	0	0	0.000724779	0
doc 14	0.000660079	0	0	0.000184091	0	8.82E-05	0	0.001285427	0	0.000305385	0
doc 15	0.000773623	0	0	0.000242727	0	0.000430684	0.000700237	0.000941589	0	0.001491315	0

Figure 4.10: A snapshot of a *term-document* matrix with words as attributes and each cell containing the *TF-IDF* value for the corresponding word in the corresponding document

As before, results deteriorate slightly when sense tags are assigned to the original corpus, with correctly classified instances reducing from 40.2703%, for the original untagged corpus, to 39.1892% for the sense tagged corpus.

4.8 Discussion

It was expected, and looked very intuitive as well, that disambiguating the original corpus by assigning sense tags to each word or phrase, would improve clustering and help to assign the correct class to unseen instances, but as shown above it showed otherwise. The proposed reasons for the above could be summarized as below:

- As has been reported before, the word alignment process is error prone and these errors are multiplied over the set of languages used to create the multilingual lexicons. But since we used a well recognized tool, GIZA++ (Och & Ney 2003), we had to live with the inherent rates (Och & Ney

2003).

- We aligned the corpora at the level of paragraphs, and not at the level of sentence, since the tools available[6] at the time did not yield good results. Better tools are already pre-processed data are now available.

- GIZA++ imposes the limit on length of 100 words for a sentence. Sentences longer than that are truncated before being used for alignment. In such a scenario a sentence aligned, rather than a paragraph aligned, and that to at the level of 1:1 sentence correspondence, could have been more useful.

- Using English as the source, rather than target, language could have reduced errors. Reasons are outlined in section 2.4.2 on pp. 72.

- The output of GIZA++ is used to group words into phrases, where possible. The algorithm to do so is one of our contributions, but since it is deterministic and no failure is possible, the quality of its output is completely dependent on its input. It does not however introduce errors on its own.

The alignment errors might have been reduced, had GIZA++ the ability to word align all the languages at once. Better pre-processing could have also helped. But given the constraints our results did not support our expectations.

4.9 Error Analysis

There are two major steps in the generation of multilingual proto-synsets:

[6] http://www.statmt.org/europarl/

1. pair-wise word alignment for English, German, French, and Greek using GIZA++ Och & Ney (2003).

2. Collating of words into phrases using our own devised algorithm (see Algorithm 3).

The second step using algorithm 3 in the thesis is deterministic and does not produce errors on its own but propagates the errors already introduced in the previous step of word alignment.

Och, et al. Och & Ney (2003) have shown that using GIZA++ induces errors in pair-wise word alignment. We quote the precision figures for word alignement when English is used as a target language, as shown in Table 4.27.

Languages	**Corpus size**	**Precision**
German → English	0.5K	77.9
	2K	88.1
	8K	90.2
	34K	92.5
French → English	0.5K	68.5
	8K	76.0
	128K	84.6
	1470K	89.1

Table 4.27: Accuracy figures for Word Alignment when English is used as a target language

As can be seen from the table above, the maximum accuracy that is obtained when English is used as the target language, is 92.5% for German, and 89.1% for French and the accuracy tends to increase with the size of the corpus. Assuming that in our case GIZA++ worked with maximum efficiency we calculate the overall accuracy as composed of accuracies of 3 pair-wise alignments which were independent of each other. The maximum accuracy can be obtained by mul-

tiplying the numbers for accuracies for each of the individual word alignments, and is:

$$accuracy = 0.925 * 0.891 * (accuracy for Greek) \tag{4.8}$$

We do not have any accuracy numbers for Greek, however if it is assumed to lie between the accuracy figures for German and French then it is 0.908 = $\frac{0.925+0.891}{2}$, which is the estimated figure. Then the combined accuracy for the three pair-wise word alignments is:

$$accuracy = 0.925 * 0.891 * 0.908 \tag{4.9}$$

which gives us a figure of 0.692 or 69.2% accuracy. An accuracy of 69.2% would cause errors in pair-wise word alignment across the languages and hence the synsets extraction using word-alignment using GIZA++ will not perform well on evaluation unless processed to a certain degree of refinement.

4.10 SemEval Parallel Corpora and Generation of Multilingual Synsets

Later, we discovered the SemEval-2010 Task 3 on Cross-Lingual Word Sense Disambiguation[7]. We used their data in six languages viz. English, French, German, Dutch, Italian and Spanish which was sentence aligned with 1:1 sentence alignment (Lefever & Hoste 2010a). The data set consisted of 884,603 sentences per language.

We did not need to do any preprocessing of the data as it was already in adequate shape for that purpose and was directly fed to GIZA++ and then mul-

[7] http://semeval2.fbk.eu/semeval2.php?location=tasks#T8

tilingual synsets were generated using the procedures outlined above. Despite being a relatively much larger data set and being pre-processed to a higher degree of refinement, the results of alignment were not very good, as can be seen from the following example:

> i declare resumed the session of the european parliament adjourned on friday 17 december 1999 , and i would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period .
> NULL (23 24 25 26 27 28 29 30 32) ich (1) erkläre (2) die (3) am (4) freitag (5) , (6) dem (7) 17. (8) dezember (9) unterbrochene (10) sitzungsperiode (11) des (12) europäischen (13) parlaments (14) für (15) wiederaufgenommen (16) , (17) wünsche (18) ihnen ( ) nochmals (19) alles (20) gute (21) zum (22) jahreswechsel (31) und (33) hoffe (34) , ( ) daß(35) sie (36) schöne (37) ferien (38) hatten (39) . (40)

A cursory look at the alignment above would tell us that the consecutive English words *to wish you a happy new year in* have not been aligned with any German word, which is a clear case of misalignment. Using such data to generate multilingual synsets cannot possibly yield good results. That made us leave using the SemEval data at that.

Yet, we used their trial data set and the sense inventory to theoretically gauge how the use of multiple languages really help us in reducing ambiguity.

4.11 Theoretical Analysis

The SemEval (Lefever & Hoste 2010b) data provides 5 target words which need to be disambiguated viz. *bank*, *movement*, *occupation*, *passage*, and *plant*. They also provide a gold standard sense inventory for each target word.

4.11.1 Sense Inventory

For each of the target words, there is a list of possible meanings in the sense inventory. For each meaning, all possible combinations of words in the 6 languages corresponding to it are listed.

The sense inventory is created by first word aligning the sentences in the Europarl parallel corpus. That gives the set of possible translations for the set of target words. The resultant translations are manually verified by certified translators.

After manual verification of translations, one annotator manually clustered them by meaning. The clusters, thus created were organized into two levels of granularity. The top level contains the main categories. For instance, for the target word *movement*, the main sense categories are: *social movement*, *traffic/motion*, and *transport*. The next level defines the finer sense distinctions.

Meanings	English	Dutch	Italian	French	German	Spanish
1. Financial institution						
1.1 Finanlcial building/entity: general		bancair	banca	banque	bank	banco
1.2 Credit/Savings (bank)						
1.2.1 credit/savings bank		kas	cassa	caisse	kredit	caja
1.2.2 Piggy bank		spaarpot	formula di risparmio	tirelire	spar§§strumpf	bancario
1.3 between banks		interbancair	interbancario			
1.4 Bank in casino/game		bank	banca	banque	bank	banca
2. Supply/Stock	rice bank	rijst§§bank	banca	banque	reis-bank	banco
3. Sloping land beside water						
3.1 General		oever	riva	rive	weichsel§§ufer	orilla
		rivier§§oever	sponda	bord	ufer	margen
3.2 burst their banks		oever	stare strari-pato	débordement	flussufer	orilla
3.3 other side		overkant	fille	côté	reihe	lado
4. Cisjordan		bank	cisgiordania	cisjordanie	West§§jordanland	cisjordania
5. group of similar objects (row/tiers)	banks of lifts	lift§§bundel	comando	système	aufzug	fila

Table 4.28: A sample of the sense inventory for the target word *bank*.

Table 4.28 gives a sample of the sense inventory with different meanings and sub-meanings for the target word *bank*. For the purposes of this exercise we considered translations at second level of granularity to be part of the meanings at the first level of granularity. So translations belonging to sub-meanings *general*, *burst their banks*, and *other side* were all considered to be translations of *sloping land beside water*, the third meaning of the word *bank* in the inventory.

4.11.2 Gold Standard

The sense inventory was later used by annotators to annotate 20 sentences per target word and were asked to give contextually relevant translations for each of the languages considered. The sentences were extracted from JRC-ACQUIS[8] and the British National Corpus (BNC)[9]. The annotators were asked to pick the sense meaning from the sense inventory which was most contextually relevant, and from the meaning cluster they were asked to pick three or fewer preferred translations. Based on their annotations, frequency weights were assigned to each translation of the target word, for each sentence, and for each of the languages considered. As a result a gold standard was created.

Given below is an example of the gold standard for the English word *bank* in German for different sentences.

> bank.n.de 1 :: bank 4;bankengesellschaft 1;kreditinstitut 1;zentralbank 1;finanzinstitut 1;
> bank.n.de 2 :: bank 4;zentralbank 3;finanzinstitut 1;notenbank 1;kreditinstitut 1;nationalbank 1;
> bank.n.de 3 :: westjordanufer 3;westufer 2;westjordanland 2;westjordanien 2;westbank 2;west-bank 1;

[8] http://langtech.jrc.it/JRC-Acquis.html
[9] http://www.natcorp.ox.ac.uk/

Each line starts with the following pattern:

{*lexelt*}{.*language*} {*id*}, where *lexelt* contains the lemma with its Part-Of-Speech (POS) tag. In the example above, the lemma is *bank* and POS tag is *n* denoting a noun. That is followed by its translations in the corresponding language, German in this case denoted by the abbreviation *de*. The number before *::* is the sentence ID. Thus, each line is a list of translations for the target word in one of the five languages, where each translation is followed by a number which signifies how many times that translation has been used by the annotators as they see the meaning of the target word, in a given sentence, from the sense inventory.

The sense inventory and the gold standard combined are a sort of perfect data set which are not given to the errors introduced by pre-processing and the word alignment step in the creation of multilingual synsets.

4.11.3 Methodology

The aim is to see how this data can be used to gauge the effects of availability of multilingual resources on monolingual disambiguation. Here is the the outline of what we plan to do:

1. Let us see if there are any combinations of words across the languages that can correspond to more than one meaning. For example, the combination of words ⟨durchfahrt, passage, paso, transito, doorgang⟩ for the target word *passage* corresponds to two different meanings: *transition, passing from one place to another* and *way through which someone/something may pass*, which are meanings 1 and 2 respectively in the sense inventory.

   (a) If so, such a synset corresponds to more than one meaning.

2. To calculate the proportion of these ambiguous synsets and the average degree of polysemy we shall:

   (a) generate all possible combinations of words allowed for each meaning, and weigh them by the frequency with which they were suggested by the translators.

   (b) count the polysemy of each word for each sentence and then average the result over the entire set of sentences.

That led us to come up with a theoretical *lower bound* on the polysemy of the target word, signifying the fact that how much polysemy is reduced when we use translations of the target word in multiple other languages for a perfect data set. It is important since knowing this would help us identify how much promise is held by using multilingual corpora in the NLP task of Cross-Lingual WSD.

4.11.4 Experiments

For the purposes of this exercise of calculating the polysemy, we took different sets of languages: the first one comprising all the five languages viz. German, French, Spanish, Italian, and Dutch. Then we took subsets of the five languages: French, Spanish, and Italian; French and Italian; and Spanish and Italian. That is to see the effect of how much polysemy is reduced when only the Romance languages $French, Spanish,$ and $Italian$, or their subsets, are considered.

We then generated all the possible multilingual synsets as outlined above, from this trial data gold standard translations, separately for each sentence. Given below is the sample from the gold standard for sentence 1. The translations are for the target word *bank* in German (de), French (fr), Spanish (es), Italian (it), and Dutch (nl). Translations are followed by the frequency with which they have been chosen by annotators from the sense inventory.

bank.n.de 1 :: bank 4;bankengesellschaft 1;kreditinstitut 1;zentral-bank 1;finanzinstitut 1;

bank.n.fr 1 :: banque 5;institution 3;bank 3;

bank.n.es 1 :: banco 5;institución 1;institución finaniciera 1;

bank.n.it 1 :: banca 4;istituto di credito 2;istituzione 2;bank 1;istituto 1;

bank.n.nl 1 :: bank 5;kredietinstelling 1;bankinstelling 1;financiële instelling 1;

In the first step the synset generation algorithm would take the first word/phrase from each language, and put them together as below:

⟨bank, banque, banco, banca, bank⟩

The next synset generated would be based on frequencies as well, so a word/phrase with a frequency greater than one would be repeated as many times. If we just take the first translations in each language, as for the synset above, we can see that their frequencies are: 4 for bank in German, 5 for banque in French, 5 for banco in Spanish, 4 for banca in Italian and 5 for bank in Dutch. Thus the next four synsets generated would all be the same.

Since all possible combinations are considered, it would then take the same words/phrases from the first four languages, but a different one from Duth. followed by:

⟨bank, banque, banco, banca, kredietinstelling⟩

This one will not be repeated since *kredietinstelling* has a frequency of only 1.

Thus, we generated all possible multilingual synsets in the above fashion based on frequencies and then compared them against the sense inventory for the target word. If a word/phrase in any language in the synset occurred under any of the main categories, the whole synset would be considered falling under

that sense category. Thus all the synsets were checked under which sense categories they fell. From that information we calculated the measures of polysemy, polysemy ratio, Precision, Recall, and F-score.

Polysemy (Equation 4.10) is defined as the average number of senses that a synset carries, and higher the value higher is ambiguity.

$$polysemy = \frac{\textit{total number of senses for all the synsets for a sentence}}{\textit{total number of synsets for the sentence}} \quad (4.10)$$

Polysemy ratio is defined as the polysemy of the word as a ratio of number of meanings in the sense inventory for that word. Mathematically:

$$PolysemyRatio = \frac{\textit{Polysemy as calculated in equation 4.10}}{\textit{Number of meanings of the word in the Sense Inventory}} \quad (4.11)$$

Precision is defined as to how does the most prevalent sense fare among all the sense categories covered by the synsets for a particular sentence. Precision would be 1 if all the synsets had only one sense, meaning perfect disambiguation. On the other hand the minimum value that Precision could take on is $1.0/$*(number of sense categories)*, when all the synsets for a sentence share all the senses.

$$Precision = \frac{\textit{No. of majority sense hits}}{\textit{total number of senses proposed}} \quad (4.12)$$

Recall depicts what portion of the synsets have the majority sense, and is defined as below:

$$Recall = \frac{\textit{No. of majority sense hits}}{\textit{total no. of synsets}} \quad (4.13)$$

Sent.	Synsets	Sen1	%Sen1	Sen2	%Sen2	Sen3	%Sen3	Sen4	%Sen4	Sen5	%Sen5	polysemy	PR	Prec	Rec	F-score
1	49280	49280	100.00	48416	98.25	44000	89.29	47012	95.40	0	0.00	3.83	0.77	0.26	1.00	0.41
2	29106	29106	100.00	28602	98.27	23760	81.63	26460	90.91	0	0.00	3.71	0.74	0.27	1.00	0.42
3	142560	0	0.00	0	0.00	82080	57.58	142560	100.00	0	0.00	1.58	0.32	0.63	1.00	0.78
4	142560	0	0.00	0	0.00	82080	57.58	142560	100.00	0	0.00	1.58	0.32	0.63	1.00	0.78
5	43120	0	0.00	0	0.00	43120	100.00	42940	99.58	0	0.00	2.00	0.40	0.50	1.00	0.67
6	81081	0	0.00	0	0.00	81081	100.00	80121	98.82	0	0.00	1.99	0.40	0.50	1.00	0.67
7	105600	105600	100.00	98688	93.45	73920	70.00	82560	78.18	0	0.00	3.42	0.68	0.29	1.00	0.45
8	149688	122040	81.53	149688	100.00	49896	33.33	92664	61.90	0	0.00	2.77	0.55	0.36	1.00	0.53
9	120960	120960	100.00	113280	93.65	84672	70.00	96768	80.00	0	0.00	3.44	0.69	0.29	1.00	0.45
10	93555	0	0.00	6237	6.67	93555	100.00	92655	99.04	0	0.00	2.06	0.41	0.49	1.00	0.65
11	129600	0	0.00	0	0.00	69120	53.33	129600	100.00	0	0.00	1.53	0.31	0.65	1.00	0.79
12	56700	56700	100.00	54950	96.91	47700	84.13	54740	96.54	0	0.00	3.78	0.76	0.26	1.00	0.42
13	224640	224640	100.00	190080	84.62	134784	60.00	155520	69.23	0	0.00	3.14	0.63	0.32	1.00	0.48
14	4800	4800	100.00	4800	100.00	4800	100.00	3520	73.33	0	0.00	3.73	0.75	0.27	1.00	0.42
15	84700	84700	100.00	67900	80.17	23100	27.27	51700	61.04	0	0.00	2.68	0.54	0.37	1.00	0.54
16	60984	60984	100.00	16184	26.54	6776	11.11	5544	9.09	0	0.00	1.47	0.29	0.68	1.00	0.81
17	47432	47432	100.00	30968	65.29	17248	36.36	21560	45.45	0	0.00	2.47	0.49	0.40	1.00	0.58
18	79200	79200	100.00	70200	88.64	43560	55.00	46800	59.09	0	0.00	3.03	0.61	0.33	1.00	0.50
19	87318	0	0.00	6237	7.14	87318	100.00	86508	99.07	0	0.00	2.06	0.41	0.48	1.00	0.65
20	48510	48510	100.00	46890	96.66	36960	76.19	45594	93.99	0	0.00	3.67	0.73	0.27	1.00	0.43
										average		2.70	0.54	0.41	1.00	0.57

Figure 4.11: German, French, Spanish, Italian, and Dutch for the target word *bank*

Finally, the F-score is the harmonic mean of precision and recall, as given below:

$$F = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{4.14}$$

The calculated values of the parameters above for the five trial target words for each of the 20 sentences in English, where the target words need disambiguating, are given in Figures 4.11, 4.12, 4.13, 4.14, and 4.15.

Table 4.29 gives a summary of statistics from the tables. It can be observed from the table that polysemy is reduced by as much as 47% when translations of a word are used as sense tags.

Sent.	Synsets	Sen1	%Sen1	Sen2	%Sen2	Sen3	%Sen3	polysemy	PR	Prec	Rec	F-score
1	174240	14520	8.33	174240	100.00	174240	100.00	2.08	0.69	0.48	1.00	0.65
2	174240	55440	31.82	174240	100.00	165024	94.71	2.27	0.76	0.44	1.00	0.61
3	76230	72114	94.60	76230	100.00	75846	99.50	2.94	0.98	0.34	1.00	0.51
4	207360	50544	24.38	207360	100.00	205092	98.91	2.23	0.74	0.45	1.00	0.62
5	248832	132192	53.13	248832	100.00	240192	96.53	2.50	0.83	0.40	1.00	0.57
6	209088	145728	69.70	209088	100.00	209088	100.00	2.70	0.90	0.37	1.00	0.54
7	228096	120096	52.65	228096	100.00	228096	100.00	2.53	0.84	0.40	1.00	0.57
8	228096	140096	61.42	228096	100.00	227296	99.65	2.61	0.87	0.38	1.00	0.55
9	191664	113256	59.09	191664	100.00	104544	54.55	2.14	0.71	0.47	1.00	0.64
10	248832	131220	52.73	248832	100.00	131220	52.73	2.05	0.68	0.49	1.00	0.65
11	145200	145200	100.00	111936	77.09	111936	77.09	2.54	0.85	0.39	1.00	0.56
12	209088	79488	38.02	209088	100.00	176832	84.57	2.23	0.74	0.45	1.00	0.62
13	121000	121000	100.00	108040	89.29	105448	87.15	2.76	0.92	0.36	1.00	0.53
14	143748	143748	100.00	130028	90.46	124540	86.64	2.77	0.92	0.36	1.00	0.53
15	171072	142722	83.43	171072	100.00	154944	90.57	2.74	0.91	0.36	1.00	0.53
16	89100	79300	89.00	89100	100.00	89100	100.00	2.89	0.96	0.35	1.00	0.51
17	144000	99648	69.20	144000	100.00	138240	96.00	2.65	0.88	0.38	1.00	0.55
18	228096	137376	60.23	228096	100.00	219024	96.02	2.56	0.85	0.39	1.00	0.56
19	171072	105732	61.81	171072	100.00	169947	99.34	2.61	0.87	0.38	1.00	0.55
20	228096	98496	43.18	228096	100.00	228096	100.00	2.43	0.81	0.41	1.00	0.58
						average		2.51	0.84	0.40	1.00	0.57

Figure 4.12: German, French, Spanish, Italian, and Dutch for the target word *movement*

Sent.	Synsets	Sen1	%Sen1	Sen2	%Sen2	Sen3	%Sen3	Sen4	%Sen4	polysemy	PR	Prec	Rec	F-score
1	145200	145200	100.00	109360	75.32	50160	34.55	73920	50.91	2.61	0.65	0.38	1.00	0.55
2	144000	144000	100.00	111072	77.13	71424	49.60	89568	62.20	2.89	0.72	0.35	1.00	0.51
3	228096	228096	100.00	158976	69.70	103680	45.45	114048	50.00	2.65	0.66	0.38	1.00	0.55
4	13824	13824	100.00	13824	100.00	13824	100.00	13824	100.00	4.00	1.00	0.25	1.00	0.40
5	129600	129600	100.00	129510	99.93	93312	72.00	97848	75.50	3.47	0.87	0.29	1.00	0.45
6	67584	51744	76.56	62544	92.54	67584	100.00	63104	93.37	3.62	0.91	0.28	1.00	0.43
7	174240	174238	100.00	170640	97.93	100320	57.58	113256	65.00	3.21	0.80	0.31	1.00	0.48
8	12800	12800	100.00	12800	100.00	12800	100.00	12800	100.00	4.00	1.00	0.25	1.00	0.40
9	110000	110000	100.00	86480	78.62	38720	35.20	45848	41.68	2.55	0.64	0.39	1.00	0.56
10	174240	174240	100.00	146520	84.09	93600	53.72	100320	57.58	2.95	0.74	0.34	1.00	0.51
11	106920	106920	100.00	84240	78.79	50787	47.50	69498	65.00	2.91	0.73	0.34	1.00	0.51
12	14080	14080	100.00	14080	100.00	14080	100.00	14080	100.00	4.00	1.00	0.25	1.00	0.40
13	11200	11200	100.00	11200	100.00	11200	100.00	11200	100.00	4.00	1.00	0.25	1.00	0.40
14	21168	20088	94.90	20688	97.73	20880	98.64	21168	100.00	3.91	0.98	0.25	0.99	0.40
15	38880	33120	85.19	38016	97.78	38880	100.00	37728	97.04	3.80	0.95	0.26	1.00	0.42
16	99000	98970	99.97	96200	97.17	64008	64.65	64008	64.65	3.26	0.82	0.31	1.00	0.47
17	73728	73728	100.00	73472	99.65	47808	64.84	51048	69.24	3.34	0.83	0.30	1.00	0.46
18	45360	36720	80.95	44064	97.14	45360	100.00	43416	95.71	3.74	0.93	0.27	1.00	0.42
19	190080	190080	100.00	166320	87.50	103680	54.55	125280	65.91	3.08	0.77	0.32	1.00	0.49
20	24948	20196	80.95	24948	100.00	24948	100.00	24948	100.00	3.81	0.95	0.26	1.00	0.42
								average		3.39	0.85	0.30	1.00	0.46

Figure 4.13: German, French, Spanish, Italian, and Dutch for the target word *occupation*

Sent.	Synsets	Sen1	%Sen1	Sen2	%Sen2	Sen3	%Sen3	Sen4	%Sen4	Sen5	%Sen5	Sen6	%Sen6	Sen7	%Sen7	polysemy	PR	Prec	Rec.	F-score
1	223608	223608	100.00	211728	94.69	170688	76.33	165396	73.97	101640	0.00	0	0.00	0	0.00	3.90	0.56	0.26	1.00	0.41
2	143000	141824	99.18	143000	100.00	110240	77.09	111800	78.18	65000	0.00	0	0.00	0	0.00	4.00	0.57	0.25	1.00	0.40
3	101088	101088	100.00	99792	98.72	81588	80.71	85488	84.57	44928	0.00	0	0.00	0	0.00	4.08	0.58	0.24	1.00	0.39
4	116160	116160	100.00	96360	82.95	82560	71.07	82560	71.07	43560	0.00	0	0.00	0	0.00	3.63	0.52	0.28	1.00	0.43
5	89100	89100	100.00	79650	89.39	55836	62.67	58300	65.43	35640	0.00	0	0.00	0	0.00	3.57	0.51	0.28	1.00	0.44
6	239580	239580	100.00	229500	95.79	169020	70.55	179100	74.76	87120	0.00	0	0.00	0	0.00	3.77	0.54	0.26	1.00	0.42
7	223608	223608	100.00	213528	95.49	155694	69.63	160104	71.60	81312	0.00	0	0.00	0	0.00	3.73	0.53	0.27	1.00	0.42
8	4410	4410	100.00	4410	100.00	4382	99.37	4284	97.14	3150	0.00	0	0.00	0	0.00	4.68	0.67	0.21	1.00	0.35
9	108900	59400	54.55	50580	46.45	64350	59.09	29700	27.27	108900	100.00	0	0.00	0	0.00	2.87	0.41	0.35	1.00	0.52
10	58968	58968	100.00	58584	99.35	56160	95.24	49608	84.13	24024	0.00	0	0.00	0	0.00	4.19	0.60	0.24	1.00	0.39
11	311040	181440	58.33	224640	72.22	138240	44.44	311040	100.00	0	0.00	0	0.00	0	0.00	2.75	0.39	0.36	1.00	0.53
12	311040	165240	53.13	213840	68.75	179820	57.81	311040	100.00	0	0.00	0	0.00	0	0.00	2.80	0.40	0.36	1.00	0.53
13	50688	44208	87.22	39888	78.69	50688	100.00	32868	64.84	17424	0.00	13824	0.00	0	0.00	3.92	0.56	0.25	1.00	0.41
14	259200	178560	68.89	198720	76.67	138240	53.33	259200	100.00	0	0.00	0	0.00	0	0.00	2.99	0.43	0.33	1.00	0.50
15	179685	77625	43.20	106785	59.43	179685	100.00	70335	39.14	60885	0.00	0	0.00	0	0.00	2.76	0.39	0.36	1.00	0.53
16	116160	116142	99.98	112200	96.59	100480	86.50	85668	73.75	56870	0.00	0	0.00	0	0.00	4.06	0.58	0.25	1.00	0.40
17	196020	111780	57.02	123120	62.81	196020	100.00	163620	83.47	50220	0.00	0	0.00	0	0.00	3.29	0.47	0.30	1.00	0.47
18	92664	92664	100.00	91824	99.09	88452	95.45	76284	82.32	47736	0.00	0	0.00	0	0.00	4.28	0.61	0.23	1.00	0.38
19	27440	27440	100.00	27200	99.13	26360	96.06	24920	90.82	19208	0.00	0	0.00	0	0.00	4.56	0.65	0.22	1.00	0.36
20	32340	32340	100.00	32340	100.00	31020	95.92	29568	91.43	16170	0.00	0	0.00	0	0.00	4.37	0.62	0.23	1.00	0.37
														average		3.71	0.53	0.28	1.00	0.43

Figure 4.14: German, French, Spanish, Italian, and Dutch for the target word *passage*

Sent.	Synsets	Sen1	%Sen1	Sen2	%Sen2	Sen3	%Sen3	polysemy	PR	Prec	Rec	F-score
1	190080	137808	72.50	190080	100.00	0	0.00	1.73	0.58	0.58	1.00	0.73
2	158400	125472	79.21	158400	100.00	0	0.00	1.79	0.60	0.56	1.00	0.72
3	209088	209088	100.00	160704	76.86	0	0.00	1.77	0.59	0.57	1.00	0.72
4	132000	132000	100.00	88000	66.67	0	0.00	1.67	0.56	0.60	1.00	0.75
5	209088	79488	38.02	209088	100.00	0	0.00	1.38	0.46	0.72	1.00	0.84
6	129600	129600	100.00	103680	80.00	0	0.00	1.80	0.60	0.56	1.00	0.71
7	248832	199680	80.25	248832	100.00	0	0.00	1.80	0.60	0.55	1.00	0.71
8	228096	228096	100.00	126720	55.56	0	0.00	1.56	0.52	0.64	1.00	0.78
9	132000	132000	100.00	96000	72.73	0	0.00	1.73	0.58	0.58	1.00	0.73
10	144000	144000	100.00	109440	76.00	0	0.00	1.76	0.59	0.57	1.00	0.72
11	159720	159720	100.00	134772	84.38	0	0.00	1.84	0.61	0.54	1.00	0.70
12	175692	175692	100.00	104544	59.50	0	0.00	1.60	0.53	0.63	1.00	0.77
13	159720	159720	100.00	90024	56.36	0	0.00	1.56	0.52	0.64	1.00	0.78
14	209088	209088	100.00	120384	57.58	0	0.00	1.58	0.53	0.63	1.00	0.78
15	190080	190080	100.00	73440	38.64	0	0.00	1.39	0.46	0.72	1.00	0.84
16	191664	191664	100.00	125136	65.29	0	0.00	1.65	0.55	0.61	1.00	0.75
17	120960	120960	100.00	92160	76.19	0	0.00	1.76	0.59	0.57	1.00	0.72
18	76230	60060	78.79	76230	100.00	0	0.00	1.79	0.60	0.56	1.00	0.72
19	143748	143748	100.00	82764	57.58	0	0.00	1.58	0.53	0.63	1.00	0.78
20	248832	248832	100.00	165888	66.67	0	0.00	1.67	0.56	0.60	1.00	0.75
						average		1.67	0.56	0.60	1.00	0.75

Figure 4.15: German, French, Spanish, Italian, and Dutch for the target word *plant*

Word	# of Unique Synsets	Polysemy Before	Avg. Polysemy After	Reduction in Ambiguity %age
bank	17,873	5	2.7	46
movement	230,061	3	2.51	16
occupation	81,706	4	3.39	15
passage	95,363	7	3.71	47
plant	91,830	3	1.67	44
Total	516,833	4.4	2.796	36.45

Table 4.29: Summary of the tables for German, French, Spanish, Italian, and Dutch for the five target words.

For other set of languages, the calculated average figures for polysemy, precision, recall and F-score for 20 sentences per target word, are summarized below (Tables 4.30, 4.31, and 4.32):

Word	**Synsets**	**Polysemy**	**Polysemy Ratio**	**Precision**	**Recall**	**F-score**
bank	812	2.45	0.49	0.45	1.00	0.60
movement	3,476	2.40	0.80	0.42	1.00	0.59
occupation	1,554	3.25	0.81	0.32	1.00	0.48
passage	2,226	3.64	0.52	0.28	1.00	0.44
plant	1,420	1.65	0.55	0.61	1.00	0.76

Table 4.30: Average figures for French, Spanish, and Italian for the target words.

Word	**Synsets**	**Polysemy**	**Polysemy Ratio**	**Precision**	**Recall**	**F-score**
bank	167	2.01	0.40	0.52	1.00	0.68
movement	456	2.16	0.72	0.47	1.00	0.64
occupation	197	2.75	0.69	0.38	1.00	0.55
passage	295	2.92	0.42	0.36	1.00	0.52
plant	235	1.49	0.50	0.68	1.00	0.81

Table 4.31: Average figures for French and Italian for the target words.

Word	**Synsets**	**Polysemy**	**Polysemy Ratio**	**Precision**	**Recall**	**F-score**
bank	179	1.99	0.40	0.53	1.00	0.69
movement	429	2.18	0.73	0.47	1.00	0.63
occupation	225	3.01	0.75	0.35	1.00	0.51
passage	341	3.20	0.46	0.33	1.00	0.49
plant	235	1.62	0.54	0.62	1.00	0.77

Table 4.32: Average figures for Spanish and Italian for the target words.

4.11.5 Baseline Comparison for Extraction of Multilingual Synsets

We designed an experiment where we annotated the target words in the SemEval sentences. We have 5 target words (bank, movement, occupation, plant, and passage), and 20 sentences per target word.

The target words in these sentences were annotated by their suggested meanings in the sense inventory. Two native and one non-native speakers of English were asked to carry out the annotation. Out of a total of 100 sentences, only in 2 cases was there complete lack of consensus among the annotators as all three of them suggested different meanings. We removed them from the evaluation.

For cases where 2 annotators agreed on 1 sense and the third one on the other, the majority vote was taken. Ultimately, we only considered the majority sense for each target word in each sentence for evaluation purposes.

We treated the annotated sentences as the Gold Standard (GS) data for evaluation of our synsets created from the SemEval data.

Most Frequent Sense (MFS) was chosen as the baseline. For each set of sentences MFS was considered. MFS from annotations of the SemEval sentences. So among all the senses assigned to the sentences by the annotators, the most frequent sense was taken and it was assumed that all the occurrences of the target word in the sentences had the same sense. We also took the top sense for each target word from WordNet, which ranks them by their frequency of use.

We then took the majority senses for each target word in each sentence from the synsets generated from the SemEval data and compared them against the GS senses. We also compared the MFS from the annotated SemEval (GS) sentences and the top senses from the WordNet, with the senses proposed in the GS. We calculated accuracy as a measure of how many of the suggested senses match those in the GS, as proposed by the synsets, by the MFS from the SemEval (GS) sentences, and by the MFS from WordNet. Our results show that the accuracy

of senses proposed by the synsets is 86%, 52% for WordNet MFS and 59% for SemEval MFS. It shows the clear benefits of our approach.

4.12 Discussion

Here we have the perfect data set without any errors introduced in the pre-processing steps or during word alignment by GIZA++. And results clearly show that given a perfect (or improved) word alignment tool, the original hypothesis stands. In total we generated 516,833 multilingual synsets from this perfect data set. Taken on its own it constitutes a perfect parallel corpus where all the words are ambiguous words and there are no function words but only content words.

In our previous experiments on Europarl, we could have taken the multilingual proto-synsets and after refinement, by merging proto-synsets that are either synonymous or morphological variations of each other, and we could have come up with a set of refined synsets just as created from the perfect data set, and then used them for word sense disambiguation. In that case results might have been better. We could not achieve it due to time constraints.

4.13 Conclusion

We wanted to see if the parallel corpus could be used to build a resource useful for carrying out NLP/IR tasks on the same corpus. So starting from a parallel corpus, Europarl in this case, we word aligned them. Once, the word alignments were there we grouped them into words, depending on if a word in one language aligned with two or more words in another language. We put those phrases together, as translations, in the languages of the parallel corpus, in the form of 4-tuples.

We used the notion of multilingual synsets to describe these 4-tuples, as opposed to the notion of synsets used in the PWN where a synset is a set of synonyms in English. There are similarities between them, where both refer to a semantic concept. While a WordNet synset is the set of synonyms of a word in the same language, in the case of multilingual synsets, it is the translations of a word in other languages with the same semantics, or we can say they are synonymous with each other in the multilingual context. Using that translation in the parallel corpus will not alter the context in which it is used. The notion of multilingual synsets also help in narrowing down the meaning of a word/phrase to fewer alternatives.

However, there are some crucial differences between the resource that we have created and a WordNet. The WordNet creates a hierarchical structure between synsets employing concepts such as synonymy, antonymy, hypernymy and meronymy. Different approaches to creating a new WordNet basically map to these hierarchical structures so that a new WordNet created in any language is basically an extension of PWN. Our resource, however, is not a *WordNet* per se, since there are no such hierarchical structures and there is no mapping between it and the PWN.

It has been clear from the start that a number of additional tools, resources, and processing steps would make the success of this approach more likely, i.e., by mapping word forms to their lexical entries or recognizing named entities.

However, this could increase the complexity of the task beyond what is viable in the time available for this PhD, and also our focus has always been on the relative benefits of using multilingual synsets and not on the performance of the ultimate tool that incorporates them.

The evaluation of the benefits of using multilingual synsets has been a perennial challenge throughout the project, since we did not have a semantically dis-

ambiguated parallel corpus at hand.

However, we carried out baseline comparison of our synsets with the word-based multilingual lexicons generated by other people. Our approach also produces some word-based synsets, so there is an overlap between what our technique does and what other people have done. Barring the intersection of the two, our approach produced lesser number of phrase-based synsets than the word-based synsets. That shows the clear benefits of our approach over what other people have done.

Thus, we decided to evaluate them indirectly by measuring the benefits of employing multilingual synsets on the task of document clustering with the unsupervised *ignorant* setup. There is no annotation. We also adopted the rather extreme assumption that we do not make use of any useful resources in the languages, such as lexicons, morpho-lexical analysers, or gazeeteers.

When we realized that the parallel corpus we used contained information which could be used to label the contents of our documents, we carried out the second set of experiments in which the benefits of multilingual synsets were evaluated on a classification task, that is on a supervised learning task, in which a decision tree was trained to classify a document on the basis of the words and the multilingual synsets they contained.

Because of the variance in word forms, or because of morphological variances, there are a lot more multilingual synsets than corresponding meanings. The gap can be reduced by merging word forms that are synonymous, for individual languages. That forms the basis of our work on morphology discussed in the next chapter.

The multilingual synsets we produced can contribute to the mono-lingual methods for the learning of semantic relationships by providing an independent point of reference. Ultimately, one can conceive an iterative process in which

mono-lingual and parallel corpus techniques progressively reinforce each other and refine their results.

The work was based on Europarl, but a very pertinent question is whether it can be used beyond the parallel corpus from which they were derived. They can be, since any parallel corpus could be processed in the same manner to create multilingual synsets and then to use them for NLP/IR tasks such as document clustering and WSD.

Towards the end of this work, we became aware of a resource that would make possible the evaluation of the potential of multilingual synsets for WSD, independent from the effects of any other NLP step, such as morpho-lexical analysis, word alignment, and so on, albeit on a relatively smaller scale.

We also carried out the baseline comparison of the senses proposed by our synsets generated from the SemEval data, when compared to the gold standard sense annotation of the target words in the sentences. We used the Most Frequent Sense (MFS) baseline, proposed by the gold standard and also by the WordNet. Our synsets had more accuracy than either of the other two MFS based baselines, which shows the clear benefits of our approach.

CHAPTER 5

Morphology and Lexical Distances

As discussed in the previous chapter, the multilingual proto-synsets generated, are more numerous in number than their meanings would suggest. That is due to syntactic (in word form) or semantic variances, such as synonymy, which are inherent to any natural language.

5.1 Main Idea

This gap in the number of proto-synsets and their meanings can be reduced by identifying such variations and merging word forms or synonyms for each language. We have indicated how word forms could be merged through the use of either existing mono-lingual lexicons or through unsupervised learning of word morphology. It is also clear that existing techniques for the learning of synonyms in any given language can be employed to the same purpose.

On the basis of the work done so far we realized that: firstly, there was a

need to map word forms to lexical entries in order to show the true benefits of multilingual synsets; secondly, that the word aligned parallel corpus can be used to learn the word paradigms of a given language without any additional expert input.

We therefore developed a methodology and carried out experiments to create such paradigms and to compare them to the output of other approaches for the unsupervised morphology learning.

5.2 Morphological Analysis

Languages are rich in morphology and some of them are more equal than others in this respect. Turkish is an example of a morphologically rich language where words can be immensely long with a number of morphemes composing a single word. Such morphologically rich languages are hard to analyse but demonstrate the art of brevity by putting morphemes together in shorter space, conveying the message with fewer number of words.

Morphology essentially deals with morphemes, the shortest form of the word that carries semantic information. For instance, the word *dog* that corresponds to a mammal is a free morpheme that can exist on its own and conveys enough semantic information without any help from other words. But such free morphemes can have inflectional forms, such as *dogs* that though still corresponds to the same mammal but is the plural form of the original morpheme. They might also be compounded with other words to form new words, for instance *doghouse*. Still other morphemes can be used to change the word POS, e.g., *biology* can be converted into *biological*. While the first one is a noun the second one is an adjective.

Analysing such morphological relationships is not trivial. We have carried

out morphological analysis of the text in the multilingual context.

We will show that the concept of edit distance (see section 5.4) can be used with benefit in calculating the morphological variation of the extracted multilingual proto-synsets, as discussed in the last chapter. They can be used to refine the proto-synsets by considering to merge the ones that are lexicographically similar (i.e., with small edit distance between them. That would help us in reducing some of the word forms to their lexical entries.

5.3 Experiments

We base our approach on the observation that word forms in a given language could belong to the same lexical entry if they share a common root (begin with the same substring). This hypothesis would be reinforced by the following factors:

1. the stem is of sufficient length;
2. the two proposed endings are not too long, and are frequently encountered;
3. the two word forms in question appear in a pair of synsets which contain identical words for some of the other three languages.

We shall describe results for the following setups:

Sect. 5.5 : segmentation of pairs of English word forms into a common stem and two endings is proposed regardless of how they are translated in the other three languages. Then we find all the stems that share the same pair of endings, and put them in the form of 'paradigms', where we have the set of stems along with the pair of endings. Finally, we go through the list of paradigms one by one, finding another paradigm with which it has

most stems in common and creating a new paradigm with the merger of the two. The resulting two sets of stems and endings can be seen as a bi-partite graph defining all word forms of a particular morphological paradigm.

Sect. 5.6 : segmentation of pairs of word forms, in any language, into a common stem and two endings is only proposed if they appear in synsets which completely overlap in the other 3 languages (a situation which we describe as "support of three" (languages).

The result of these is a lexical resource matching a stem to a list of possible endings[1]. We have further considered how roots sharing the same set of endings could be combined together to form a class of roots that take the same set of endings, i.e., to form morphological paradigms. In doing so, we have considered two cases, that of roots sharing exactly the same endings, and another, where the roots share 'sufficiently many' endings, i.e., a certain, large, percentage. Note that the result of the latter case is the ability of our morphological lexicon to analyse some unseen word forms, although, of course, there is the possibility of an error in allowing this generalisation. We also evaluate our results through a comparison with a gold standard corpus (lexicon segmented into morphemes), and with another unsupervised method for word segmentation.

To recapitulate the above account: in most experiments, we start by grouping together synsets which contain the same English word or phrase. This means that the group will contain all word forms corresponding to the same English word or phrase in all other languages. We then proceed to:

- Create all possible synset-pairs within a group, if it has at least two synsets in it.

[1] which could be used to map word forms onto a single root / lexical entry

- We look at English monolingual data and propose segmentations, organizing them into paradigms.
- Finally, we consider carrying out word segmentation in all four languages, provided there is an overlap between synset-pairs in 3 of the 4 languages, or in other words each segmentation has support of exactly 3. We did not create any paradigms, but compared the results segmentation with the gold standard data and analogy, an unsupervised segmentation technique.

5.4 Edit Distance

Gusfield (1997) gives an introduction to edit distances and describes it as an inexact matching problem where given any two strings, the minimum number of steps in which starting from one string and coming up with the other is ascertained.

As per Gusfield, it is defined as: "The *edit distance* between two strings is defined as the minimum number of edit operations - insertions, deletions, and substitutions - needed to transform the first string into the second. For emphasis, note that matches are not counted." Gusfield gives an example where starting from string *vintner*, one converts it into *writers*.

```
RIMDMDMMI
v intner
wri t ers
```

Four edit operations are permitted: *insertion* (I) of a character into a string, *deletion* (D) of a character from the string, *replacement* (R) (or *substitution* (S)) of a character with another character, or a non-operation of *match* (M). The minimum number of such operations is the distance between any two strings, also

known as the Levenshtein distance in recognition of the paper written by V. Levenstein (Levenstein 1966), who probably first discussed the concept.

5.4.1 Calculating Edit Distances

The edit distance problem, as defined by Gusfiled is "to compute the edit distance between two given strings, along with an optimal edit transcript that describes the transformation." (Gusfield 1997) It can be viewed as simultaneously doing edit operations on the two strings, which might yield a third string, which is the desired solution. Since insertion in one string can be taken as a deletion in the other.

The edit transcript is a way to represent the sequential set of operations applied to a string, thus the sequence RIMDMDMMI in the example given for words *vintner* and *writers* is the edit transcript. There might be more than one such optimal transcripts.

Finding the edit distance is basically a string alignment process, whereby either spaces are introduced corresponding to either the insertion or the deletion operations or characters are mismatched to indicate substitution. Take the example of two strings *qacdbd* and *qawxb*, as given in Gusfield's. When put together they are aligned as such:

```
q a c _ d b d q a w x _ b _
```

The characters that match (q, a, and b) are put opposite to each other. c is put opposite a w signifying a substitution operation. A space (dash) in the first string signifies insertion and in the second signifies deletion. The edit distance is given by minimizing the number of mismatched characters and the number of characters opposite spaces (dashes).

Dynamic programming technique has been used to compute the edit distances, as defined in (Wagner & Fischer 1974), and as given in Algorithm 6:

Algorithm 6 Calculating Edit Distances

$D[0,0] \leftarrow 0$
for $i = 1 \rightarrow |A|$ **do**
  $D[i,0] \leftarrow D[i-1,0] + \gamma(A < i > \rightarrow \lambda)$
end for
for $j = 1 \rightarrow |B|$ **do**
  $D[0,j] \leftarrow D[0,j-1] + \gamma(\lambda \rightarrow B < j >)$
end for
for $i = 1 \rightarrow |A|$ **do**
  for $j = 1 \rightarrow |B|$ **do**
    $m_1 \leftarrow D[i-1,j-1] + \gamma(A < i > \rightarrow B < j >$
    $m_2 \leftarrow D[i-1,j] + \gamma(A < i > \rightarrow \lambda)$
    $m_3 \leftarrow D[i,j-1] + \gamma(\lambda \rightarrow B < j >)$
    $D[i,j] \leftarrow min(m_1, m_2, m_3)$
  end for
end for

Algorithm 6 is recursive. γ gives the cost of different edit operations. $\gamma(A < i > \rightarrow \lambda)$ represents the cost of deletion. $\gamma(\lambda \rightarrow B < j >)$ gives the cost of insertion. And $\gamma(A < i > \rightarrow B < j >)$ gives the cost of substitution. The non-operation of match has no cost associated with it. Thus the cost of starting with a string and ending up with an empty string is the length of the first string, since all the characters need to be deleted. The cost of starting with an empty string and ending up with some string is the length of the second string since all the characters need to be inserted. These two form the base cases of recursion. The aim is to find the minimum cost of all such operations which starts from one string and ends up with another as represented by $min(m_1, m_2, m_3)$.

5.4.2 Edit Distances between Multilingual Proto-Synsets

The multilingual proto-synsets generated in the previous step are in a relatively crude form with two or more separate proto-synsets for words that might be inflections of each other (Table 5.1).

abolish	abschaffen	abolir	καταργήσει
abolished	abgeschafft	aboli	καταργήθηκε

Table 5.1: An example of morphological syntactic variation

In Table 5.1 the two proto-synsets are for the same basic lexeme *abolish*, but the second proto-synset is for the inflectional form of the word and is the past tense form. Such inflections are relatively easy to spot since, as a rule, they have relatively small edit distances between them. This, of course, is not always the case, e.g., for irregular verbs, *is* and *are* the small edit distance does not imply that they are morphological variation of each other. The more difficult to spot are the synonyms, where two different words can substitute each other in a context without changing it. Synonymous words may have large edit distances since they might be totally different words and a number of operations might be required to start with one and convert it into another. So large distances can mean either that the two words are synonymous or might be the result of misalignment. The translations of such two words in other languages can indicate if they hold the synonymy relationship with each other or not. If their translations are the same or close inflections of each other then there is a higher chance that they are synonymous. Table 5.2 gives one example of such a case.

administration	verwaltung	administration	διοίκηση
administration	verwaltung	administration	διαχείριση

Table 5.2: Pair of Greek synonyms

As can be seen from the example, English, German and French have been translated in the same way for the two Greek words $\delta\iota o\acute{\iota}\kappa\eta\sigma\eta$ and $\delta\iota\alpha\chi\varepsilon\acute{\iota}\rho\iota\sigma\eta$, which have an edit distance of 5, and could also be translated as *administration* and *management* respectively, are synonymous with each other, as WordNet confirms. Since the Greek words are translated using the same word forms in other languages, we can say that they have support of 3.

We have devised the following experiment. We took the original proto-synsets created, as explained in Chapter 4. We then separated the proto-synsets into groups where in each group all the proto-synsets shared the same English word/phrase. Some of the proto-synsets are for English words/phrase that have only one synset associated with them. But edit distances can only be measured between pairs of proto-synsets. Thus, all those groups of proto-synsets were dropped which had only proto-synset in them .

After that within each group all possible pairs of proto-synsets were created and edit distances were calculated for them. Edit distances were calculated for each of the languages in a synset pair and then the total edit distance was calculated by adding values of edit distances for each individual languages in the synset pair. Finally, the pairs with in each group were ordered with respect to their total edit distances in ascending order.

americans	amerikanische	américaine	$\alpha\mu\varepsilon\rho\iota\kappa\alpha\nu\iota\kappa\acute{\eta}$
0	5	1	3
americans	amerikanern	américains	$\alpha\mu\varepsilon\rho\iota\kappa\alpha\nu\acute{\omega}\nu$

Table 5.3: An example of synsets with the same English word/phrase but different translations and the corresponding edit distances for each of the translated words/phrases

As Table 5.3 shows the English words are the same, the edit distance between German words is 5, French words is 1, Greek words is 3 and in total (0 + 5 + 1 + 3) is 9. Apparently the words in the same language are just inflectional variations

of each other, depicting closer proximity in terms of semantics and only syntactic differences.

There were a total of 441,163 proto-synsets with unique English phrases, out of which 89,234 occurred with a frequency of 1 and the rest had frequency more than 1.

5.5 Looking for Word Paradigms

The aim here is to segment word forms in any of the four languages, but we only created paradigms for English. Word segmentation in other three languages were not considered for creating paradigms. We do it by comparing the word forms in each language and taking the maximum common prefix of size at least 4. We do

administer	verwalten	administrent	*χαράσσοιυν*	0 13 9 12 34
administer	haushaltsvorgänge	facilité	*διευκολύνεται*	0 13 9 12 34
administered		géré	*ποσόστωσης*	0 0 4 9 13
administered			*διεγερτικές*	0 0 4 9 13
administered	hiervon	géré	*διαχειρίζονται*	0 7 4 10 21
administered			*διεγερτικές*	0 7 4 10 21
administered	hiervon	géré	*διαχειρίζονται*	0 7 0 14 21
administered		géré	*ποσόστωσης*	0 7 0 14 21
administered	unbürokratische	définissant	*συναλλαγή*	0 15 10 10 35
administered		géré	*ποσόστωσης*	0 15 10 10 35
administered	unbürokratische	définissant	*συναλλαγή*	0 15 11 11 37
administered			*διεγερτικές*	0 15 11 11 37
administered	unbürokratische	définissant	*συναλλαγή*	0 14 10 14 38
administered	hiervon	géré	*διαχειρίζονται*	0 14 10 14 38
administering	verwaltung	cofinancement	*διαχείρισης*	0 0 12 1 13
administering	verwaltung	allégé	*διαχείριση*	0 0 12 1 13

Table 5.4: A sample of groups of synset pairs with lexical distances for individual languages and for the entire synsets.

it separately for each synset-pair.

Table 5.4 gives a sample of the data used for word segmentation. We have divided the data into groups, where each group contains set of synsets where English word forms are the same. The groups are demarcarted by horizontal lines here.

We experimented with this data for each of the three languages, but decided to concentrate on experiments using data with support of 3 (see section 5.6). Before we present these results, we shall report experiments with English data which did not make any use of the multilingual information, as they can provide a feel of what one can expect from a comparable monolingual corpus.

Within a group, we take all possible synset pairs which share the same English word forms. We can say that translations in these languages in each synset-pair have support of at least 1. Since English word forms are same in each synset pair, we can not use them to segment English words. However, they can be used to segment word forms in German, French, and Greek.

However, English word forms can be segmented by comparing English word forms between groups. English word forms may or may not have any support in a synset pair, since translations in other languages may be different. But in some synset pairs, the English word forms may have support of 1 or 2. Support of 3 is not possible for English within a group, since it would mean that the two synset-pairs are exactly the same. Such synset-pairs can not exist because only the unique synsets were considered for calculating edit distances, to begin with. We can use the same criterion to segment English word forms, that the size of the common prefix (stem) should be at least 4 letters.

Table 5.4 gives a snapshot of the set of proto-synsets with their edit distances for all the languages individually, and the total. As can be observed even though English words and phrases remain the same in one particular set of proto-synsets

Algorithm 7 Algorithm for separating stems from endings

```
Get two words in the same language from which to extract stems and endings
Start comparing the letters in each word from the left
Find the minimum number of common letters, that is the stem
the rest is the ending
if length of stem is at least 4, then output the stem and the ending
```

or a group, they do change from one set (group) to the next.

In Table 5.4 we have proto-synsets for three English words administer, administered and administering. Carrying out the morphological analysis for them would yield the stem as *administer* and the set of endings as ⟨*ed,ing*⟩, which is also called as a *signature*.

The algorithm for separating stems from endings is listed as Algorithm 7. That is a simple algorithm that carries out unsupervised morphological analysis of the proto-synsets. As a result we ended up with a total number of 23,935 unique English stem-endings, 118,559 German, 96,395 French and 153,061 unique Greek stems and endings. But they contained a lot of redundancy in the case of German, French and Greek since all proto-synset pairs were considered for their extraction which shared the same English phrase, and there was a high chance that phrases in other languages were repeated. A lot of entries also contained 2 nulls, due to the fact that many times two proto-synset pairs had exactly the same two phrases in any of the non-pivotal languages. Some also contained multiple word phrases. Some entries were also alpha-numeric and even purely numeric. All these problems can be resolved but would require more pre-processing before applying the segmentation algorithm.

After processing the obtained word segmentations to remove such problems, we were left with only 4,929 English, 50,961 German, 35,040 French and 89,345 Greek stems with their signatures. From now on, we only considered the English word segmentations and we created paradigms for them.

It is important to look for support for each signature, in other words how many stems share the same set of endings. To ascertain the level of support we sorted the entries in ascending order with respect to the endings, thus *phenomen* ⟨*a on*⟩ would come before *referend* ⟨*a um*⟩. Then we removed all the entries with signatures having support of just one stem. After this step we were left with 3,023 pairs of stems and their endings.

5.5.1 Merging Paradigms

A paradigm is the set of stems that share the same set of endings. Certain paradigms may share certain number of stems and they have the potential to be merged by taking a union of their endings and putting the common stems in the new paradigms. Merging paradigms helps in creating more generalized paradigms which may cover more words but may also have the potential of increasing noise by putting stems and endings together which do not form valid words when combined together.

We use a bottom-up approach where we start from a point where every stem with a pair of endings is a paradigm. Then we start putting stems together in the same paradigm if they share the same set of endings. Each stem increases the support that a paradigm enjoys. The more the stems are in a pardigm, the more support it has. It is easy to carry out this step with just one run through the entire list of stems with their signatures, since they are sorted by the endings.

The paradigms are then put in the descending order of their support, and from the list any two paradigms are chosen for merger which share the maximum number of common stems. A new paradigm is created by taking a union of the endings in the original paradigms and putting them and the common stems, between the two original paradigms, into the new paradigm. These common stems are removed fom the original paradigms. Old paradigms are removed if

they are left with no stems in them since they loose any manner of support. The new paradigms created and the paradigms from which they have been created, if they still have support of at least one stem, are then made part of the original set of paradigms. The rest retain their existence as long as they have the support of at least one stem. Algorithm 8 outlines the process.

Algorithm 8 Looking for paradigms

```
sort the list of stems and endings in descending order of endings
merge the stems that share the same set of endings
for each paradigm in the list of paradigms
    find a paradigm with max. no. of common stems
        create a new paradigm
        take a union of their endings
        put the common stems in the new paradigm
        remove the common stems from the original paradigms
        remove a paradigm if no more stems left in them
```

That yields a total of 454 paradigms with 182 of them, 40%, that enjoy the support of only one stem. 107 of them enjoy the support of 2 stems each while the rest enjoy the support of at least 3 stems. The paradigm with the highest support has a support of 399 stems where the endings are <null,s>. That makes sense since many words, 399 in this case, when suffixed with *s* either correspond to the plural of it or makes it into a 3rd person singular present tense. For instance, *abduction* affixed with *s* becomes the noun *abductions* which is the plural form of *abduction*. But an *s* affixed with say *accede* would yield the 3rd person singular present tense of the word. Both of them exist in the paradigm with the signature *null,s*.

Endings	Support	Stems
⟨∅ ,s⟩	399	abduction,abstention,academic,accede,accusation,...
⟨null,d⟩	127	accelerate,accommodate,acquire,advocate,allocate,announce,...
⟨es,ing⟩	112	abolish,acced,acknowledg,address,advocat,allocat,analys,appreciat,...
⟨ed,ing⟩	97	accelerat,accommodat,acquir,anticipat,assess,assum,assur,astound,...
⟨ies,y⟩	81	abilit,accompan,agenc,ambiguit,appl,bankruptc,beneficiar,capabilit,...
⟨null,ly⟩	78	accidental,according,acute,admitted,alleged,anonymous,apparent,approximate,...
⟨ng,on⟩	76	accelerati,accessi,accommodati,allocati,anticipati,appreciati,associati,...
⟨d,s⟩	57	abolishe,aggregate,allie,annexe,argue,believe,breache,challenge,clarifie,...
⟨null,ing⟩	54	allay,benchmark,bend,bind,bolster,boycott,burn,constrain,dawn,deny,dock,...
⟨null,d⟩	49	abuse,aggravate,analyse,appreciate,authorise,cite,couple,criticise,delegate,...
⟨null,ed⟩	42	access,amend,annex,applaud,breach,concert,connect,crush,curtail,deem,defeat,...
⟨ility,le⟩	30	acceptab,accessib,accountab,admissib,affordab,applicab,availab,capab,comparab,...
⟨ing,s⟩	29	accept,affect,attempt,await,conflict,detail,extend,farm,flood,function,guarantee,...
⟨ed,ing,null⟩	29	adjust,administer,alter,broaden,conduct,construct,contest,convert,convey,curb,...
⟨ce,t⟩	28	absen,coheren,convenien,deterren,dominan,equivalen,excellen,ignoran,inciden,...
...	...	...
⟨ly,s⟩	6	essential,friend,month,objective,official,year
⟨er,ing⟩	6	bann,clean,join,remind,waiv,warn
⟨null,er⟩	5	bold,campaign,cheap,mann,rich
⟨es,ing,ation,e⟩	5	combin,determin,realis,restor,subsidis
...	...	...
⟨null,ful,s,ual⟩	1	event
⟨null,ance⟩	1	resist
⟨null,ed,ors,s,ments,or⟩	1	invest
⟨ing,m⟩	1	centralis

Table 5.5: A sample of the paradigms created.

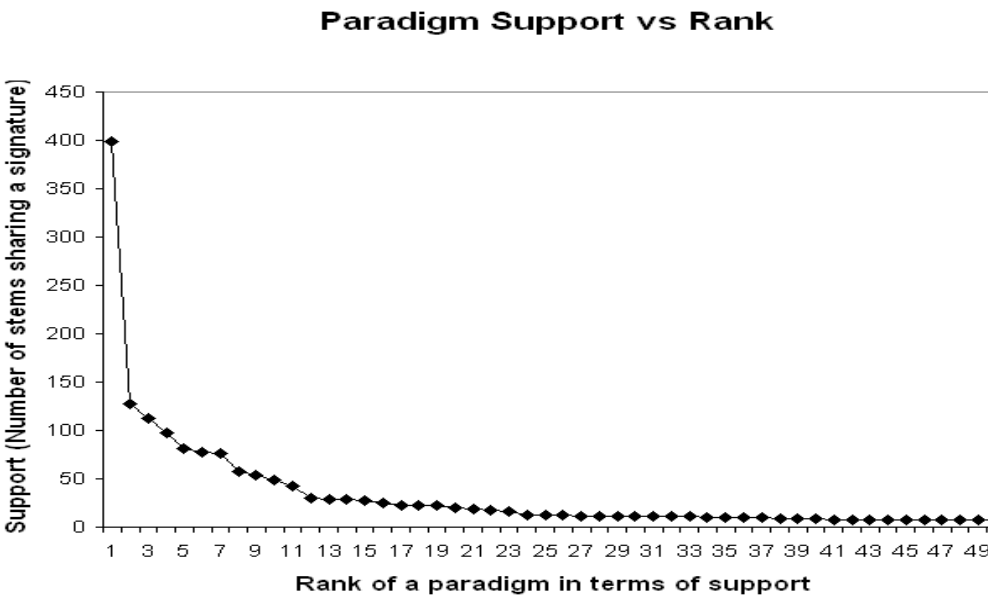


Figure 5.1: Support enjoyed by a paradigm vs. its rank, giving rise to a curve similar to the one for Zipf's Law.

Table 5.5 gives a sample of paradigms where the first column corresponds to the set of endings shared by the stems in the 3rd column. Column 2 gives support enjoyed by the set of endings, in terms of how many stems have been found sharing the same set of endings. The endings with the highest support of 399 stems are $\{\emptyset, s\}$. We plotted the support enjoyed by each paradigm vs. its rank, for the top 50 paradigms ordered by their support, and the resultant chart looks like the familiar Zipf's Law (see Figure 5.1).

5.5.2 Merging Paradigms based on Common Number of Stems

Another set of experiments was carried out where two parameters were defined: *Common Factor* (CF), and *Percentage of Compared Stems* (PCS). We take the pervious set of merged paradigms and carry out comparisons between them and merge paradigms that have more stems in common than a predefined threshold.

CF is defined as the ratio of number of stems in common between two paradigms, that are compared, to the length of the original set of stems.

$$Common\ Factor = \frac{\#\ of\ Stems\ in\ Common\ Between\ Paradigms\ Compared}{\#\ of\ Stems\ in\ the\ First\ Paradigm} \tag{5.1}$$

PCS is defined as the ratio of number of stems in common between two paradigms, that are compared, to the length of set of stems with which it is compared.

$$\%age\ of\ Stems = \frac{\#\ of\ Stems\ in\ Common\ Between\ Paradigms\ Compared}{\#\ of\ Stems\ in\ the\ Second\ Paradigm} \tag{5.2}$$

We only merge paradigms if they are different and CF is at least greater than and equal to a certain threshold. Here, merging means we take the union of both stems and endings in the paradigms to be merged, and put them together as a new paradigm. Table 5.6 gives how many new paradigms have been created for different merge thresholds out of a total of 454 paradigms originally created.

Table 5.7 presents a sample of new paradigms created as a result of merging paradigms based on pre-defined thresholds of 0.33, 0.50, and 0.67.

Merge Threshold	**Number of New Paradigms**
0.33	35
0.5	28
0.67	9

Table 5.6: How Number of new Paradigms created changes with the Merge Threshold?

Endings	support	Stems
⟨r,st,ty⟩	13	bigge,cheape,deepe,functionali,newe,poore,quicke,riche,safe,simple,stricte,...
⟨ed,ment,s⟩	7	align,enjoy,fulfil,imprison,indict,involve,replace
⟨ary,s,null⟩	5	element,precaution,reaction,revolution,vision
⟨ary,s,null⟩	5	element,precaution,reaction,revolution,vision
⟨ants,ed,ers,ing,s⟩	5	account,assist,defend,export,protest
⟨ation,ations,ed,ing,null⟩	8	alleg,civilis,condemn,confront,expect,generalis,install,vari
⟨ary,s,null⟩	5	element,precaution,reaction,revolution,vision
⟨ary,s,null⟩	5	element,precaution,reaction,revolution,vision
⟨ability,ably,ed,ing,null⟩	5	account,favour,regrett,remark,sustain
⟨ability,ants,ed,ing,null⟩	4	account,assist,defend,sustain
⟨ation,ations,ed,ing,null⟩	8	alleg,civilis,condemn,confront,expect,generalis,install,vari
⟨es,ing,s,y⟩	7	german,part,read,risk,speed,unit,victor
⟨ability,ed,ing,ment⟩	5	agree,disappoint,employ,enlighten,punish
⟨en,est,ness,null⟩	3	happi,rich,weak
⟨able,al,ed,ing,null⟩	3	deferr,deni,renew

Table 5.7: A sample of new paradigms created after merging paradigms based on the thresholds of 0.33, 0.50, and 0.67 without signature refinement

5.5.2.1 Signature Refinement and Merging Paradigms based on Common Number of Stems

Some paradigms also show more interesting patterns where the last letter in each stem is the same, implying that it could be combined with the endings to refine them. For instance, in the following paradigm:

"⟨es,ng⟩" 5 "[citi,counti,deliveri,polici,treati]"

the last letter in each stem is *i*, which can be combined with the two endings to yield a new paradigm, as given below:

"⟨ies,ing⟩" 5 "[cit,count,deliver,polic,treat]"

We look at paradigms and see if the stems in a particular paradigm share the same last letter. If they do, we refine the signatures by prefixing the common last letter to the endings and removing it from the end of the stems, as shown above.

But since even after carrying it out once we may have cases where there is a common last letter shared by the stems, we need to iterate over it a few times.

We carried out a series of experiments where the paradigms were first refined and then merged based on a certain merge threshold. We did four iterations of signature refinement so the output of refinement in one iteration would become the input for signature refinement in the next iteration.

During each iteration we used three merge thresholds viz. 0.33, 0.5, and 0.67 for merging. So first we performed signature refinement and then merged paradigms which were different and had CF greater than the merge threshold.

Table 5.8 demonstrates how signatures and corresponding stems are refined over four iterations. For this paradigm, the process converges in the 3rd iteration, and thus in the 4th iteration we get the same results as in the 3rd iteration.

Iteration	Signature	Stems
1	⟨le,ly⟩	commendab,inevitab,notab,preferab,probab,regrettab,remarkab,undeniab,...
2	⟨ble,bly⟩	commenda,inevita,nota,prefera,proba,regretta,remarka,undenia,...
3	⟨able,ably⟩	commend,inevit,not,prefer,prob,regrett,remark,undeni,...
4	⟨able,ably⟩	commend,inevit,not,prefer,prob,regrett,remark,undeni,

Table 5.8: An example of signature refinement over four iterations.

Table 5.9 shows how the number of new paradigms created as a result of merging varies with the threshold and the iteration.

Threshold	Iteration	Number of new Paradigms
0.33	1st	52
0.33	2nd	53
0.33	3rd	55
0.33	4th	54
0.5	1st	52
0.5	2nd	53
0.5	3rd	55
0.5	4th	54
0.67	1st	52
0.67	2nd	53
0.67	3rd	55
0.67	4th	54

Table 5.9: The results of experiments on refining signatures and merging paradigms

The process of refinement and merging of paradigms is described below as Algorithms 9 and 10, one for refinement and the other for paradigm merging.

5.5.3 Discussion

We have shown how proto-synsets can be used to first carry out word segmentation in an unsupervised manner, and then to create paradigms in English. These paradigms can be useful in deciding to which paradigm a new unseen instance would belong to, and what could be its possible endings.

However, we did not evaluate the paradigms since we were interested in morphological analysis from the multilingual aspect. Looking at support from other languages would provide that multilingual aspect to our morphological analysis.

Algorithm 9 Ending refinement

$Paradigm[< set\ of\ endings,\ support,\ set\ of\ stems >]$
$commonLastLetter \leftarrow TRUE$
for $i = 2 \rightarrow (set\ of\ stems).length$ **do**
  if $!(set\ of\ stems)[i].lastLetter == (set\ of\ stems)[i-1].lastLetter$ **then**
    $commonLastLetter \leftarrow FALSE$
  end if
end for
if $commonLastLetter == TRUE$ **then**
  for $j = 1 \rightarrow (set\ of\ endings).length$ **do**
    $endings[j] \leftarrow endings[j] + lastCommonLetter$
  end for
  for $k = 1 \rightarrow (set\ of\ stems).length$ **do**
    $stems[k] \leftarrow stems[k].substring(1, stems[k].length - 1))$
  end for
end if

Algorithm 10 Paradigm Merging based on common number of stems

$commonFactor \leftarrow -1$
$numberOfCommonStems \leftarrow -1$
First carry out Signature Refinement as given in Algorithm 7
for $i = 0 \rightarrow (set\ of\ refinedparadigms).length$ **do**
  $paradigmStemsOriginal[] \leftarrow paradigmsOriginal[i][2].split('delimiter')$
  for $j = 0 \rightarrow (set\ of\ refined\ paradigms).length$ **do**
    $paradigmStemsForComparison[] \leftarrow$
    $paradigmsForComparison[j][2].split('delimiter')$
    if $i \neq j$ **then**
      calculate numberOfCommonStems
      $commonFactor = numberOfCommonStems/paradigmStemsOriginal.length$
    end if
  end for
end for
if commonFactor $\geq$ mergeThreshold **then**
  $newEndings = paradigmsOriginal.endings \bigcup paradigmForMerging.endings$

  $newStems = paradigmsOriginal.stems \bigcup paradigmsForMerging.stems$
end if

Still, we showed that there is a nice relationship similar to Zipf's Law between the support enjoyed by each paradigm and its rank.

5.6 Further Experiments in Multilingual Morphology

One way to ascertain the validity of attained stems and endings is through the support that a word gets from its translations in other languages. The more the translations of a word agree in other languages, more support its morphological analysis has.

Thus we took all the synsets that only had alphabetic words/phrases in them, leaving us with 241,590 synsets in total. They were searched for synset pairs where for three of the languages they contained identical entries, as in Table 5.2. This yielded a set of 660,272 synset pairs.

Extracting stems and endings and catering for the requirement of at least 4 letters in a stem, we ended up with 983 English, 17,156 German, 3,667 French and 15,853 Greek 'paradigms'. Here each paradigm is just one stem with two endings, one of which can be null.

Table 5.10 gives a sample of word segmentations in all languages.

5.7 Evaluation of Morphological Analysis

Here we want to evaluate how the word segmentations, stems and their endings, proposed by our algorithm compare with the word segmentations proposed by a gold standard van den Bosch et al. (1996).

In order to carry out this evaluation we need to bring our own word segmentations in the same format as that of the gold standard. Table 5.11 gives a sample

English			
	adopt	ed	ing
	barbari	sm	ty
	chair	man	person
	implement	ation	ing
German			
	altersgr	enzen	uppen
	verträge	∅	n
	alkohol	frage	genuß
	änderungsantr	ag	ägen
French			
	administrati	fs	on
	adopt	er	ée
	afri	cain	que
	modifi	ant	cation
Greek			
	$\delta\iota\omicron\iota\kappa\acute{\eta}\sigma\epsilon$	$\iota\varsigma$	$\omega\nu$
	$\epsilon\gamma\kappa\rho\acute{\iota}\nu$	$\epsilon\iota$	$\omicron\upsilon\mu\epsilon$
	$\epsilon\gamma\kappa\rho\acute{\iota}\nu$	$\epsilon\iota$	$\omicron\upsilon\mu\epsilon$
	$\upsilon\iota\omicron\theta\epsilon\tau\acute{\eta}$	$\theta\eta\kappa\epsilon$	$\sigma\epsilon\iota$

Table 5.10: A sample of word segmentations in the four languages

of how words and their segmentations look like in the gold standard.

In Table 5.11, the first column is the word to be segmented, the second column is the POS tag of the word and in the third column we have the proposed segmentation in the form of a binary code.

In a word a segmentation point is where a cut is made to partition the word into a *prefix*, the leading part, and a *suffix*, the trailing part. Thus, the word *walked* can be segmented between the letters *k* and *e*, yielding the prefix *walk* and the

Word that is segmented	POS tag	Segmentation
abandon	VB	10000001
abandoned	VBN	1000000101
abandoning	VBG	10000001001
abandonment	NN	100000010001
abandons	VBZ	100000011
abasement	NN	1000010001
abatement	NN	1000010001
abbey	NN	100001
abbot	NN	100001
abbreviation	NN	1000001001001
abbreviations	NNS	10000010010011
abdominal	JJ	1000000101
abduction	NN	1000001001
abed	RB	11001
aberrant	JJ	101001001
abetted	VBN	10001001

Table 5.11: A sample of words, their POS tags and the proposed segmentations in the gold standard.

suffix *ed*. *walked* itself has the binary code *1000001*. 1's identify the split points, and 0's identify that there is no split at that point. 1's on both sides with 0's in the middle, signify that the word is not segmented, since the leading and trailing 1's identify the word boundaries.

In the sample in Table 5.11, the word *abandoned* has the segmentation *1000000101*, which means it is segmented at the point between letters *n* and *e*, and thus the prefix and suffix are *abandon* and *ed* respectively.

5.7.1 Evaluation

To evaluate the performance of segmentation algorithms, we used *Precision* and *Recall*. In the context of morphological analysis the two terms are defined in

terms of number of segmentation points in the gold standard and our data. Since a cut is denoted by a 1, we count the number of 1's in the binary code for segmentation for a word form.

Thus, precision is defined as the ratio of number of shared segmentation points, or 1's, between the binary code for gold standard (GS) word form segmentation and the binary code suggested by our algorithm for word form segmentation, to the number of 1's in our proposed segmentation. Mathematically:

$$Precision = \frac{\textit{\# of shared 1's between GS segmentation and our segmentation}}{\textit{\# of 1's in our proposed segmentation}} \tag{5.3}$$

Recall has the same numerator as for Precision, but the denominator is for the gold standard segmentation. Mathematically:

$$Recall = \frac{\textit{\# of shared 1's between GS segmentation and our segmentation}}{\textit{\# of 1's in the GS segmentation}} \tag{5.4}$$

5.7.2 Segmentation with Support

The paradigms created for English in the last step have support from other languages. Thus, we used them to compare against the gold standard to see how well our word segmentation algorithm performed. By bringing the paradigms to the format of the gold standard, we ended up with 1,483 words with their segmentations.

5.7.3 Analogy Principle

We compared our method for segmentation with the one based on the principle of analogy as has been discussed in section 2.2.2.1. We built a lexicon of word forms to be segmented.

Algorithm 11 Algorithm for implementation of the Analogy principle

```
for all LW1 ∈ Lex do
  if LW1 = P1 + S1 then
    if LW2 ∈ Lex & LW2 ≠ LW1 & LW2 = P1 + S2 then
      if LW3 ∈ Lex & LW3 = P2 + S2 then
        if P2 ≠ P1 then
          if LW4 = P2 + S2 & LW4 ∈ Lex then
            Split LW1 into P1 and S1
          end if
        end if
      end if
    end if
  end if
end for
```

We used the analogy principle (see Algorithm 11) to get segmentations for the same words on which we applied our segmentation algorithm as outlined in section 5.6 and Algorithm 7.

Analogy principle gives a lot more segmentations than our algorithm, which only cuts at one point because it carries out the exhaustive search in a search space where every point is one particular segmentation of a word form.

5.7.4 Results

We calculated the results for segmentation based on the analogy principle and our own method. And then we took an intersection of the two by only taking the segmentations proposed by both the methods and also compared them against the gold standard segmentations.

After evaluating the methods we got the following results:

Method	Precision	Recall	F-Score
Analogy	0.449	**0.870**	0.592
Segmentations extracted from Synsets	0.856	0.711	0.777
Intersection of both	**0.887**	0.706	**0.786**

Table 5.12: Precision and Recall for segmentations proposed by the analogy principle, our own method and for the intersection of both.

Table 5.12 gives a summary of results. The segmentation based on the analogy principle gives much worse performance than our method. Even though recall is high, because it proposes a lot more segmentation points, but precision is very low, as a lot of them do not appear in the gold standard.

Our method has good precision, 86% which is much higher than analogy, with low recall of 71%. However, the intersection of the two scores slightly lower on the recall measure than any of the other two, but precision goes up even further to 89%. It is reflected in the F-Score measure, which goes up from 0.59 for Analogy to 0.79 for the intersection.

What it means is that our method is better than the one based on the analogy principle but the combination of both gives even better results with F-Score rising to 0.79. Precision improves by 3% to 89% with a slight drop in recall of 1% to 70.6%. That makes sense since in the combined segmentation data the number of 1's can not be more than the number of 1's in either the one based on analogy or our method.

5.8 Conclusion

This chapter gives details of the work that has been carried out in the direction of refinement of proto-synsets by identifying morphological variations between different wordforms in all the languages, which could be later used to merge the

proto-synsets. Lexical distances have been measured for pairs of proto-synsets. While a small lexical distance might indicate an inflectional variation, larger distances might mean either the two words are synonyms of each other or are wrong translations. Such information can come in handy while merging proto-synsets that have low inlfectional variation.

Then on the proto-synsets data with lexical distances calculated, we carried out morphological analysis by separating stems from endings. For English word segmentation we did not look for support in other languages. The English word segmentations, thus created, formed the basic 'paradigms' with a stem sharing a pair of endings. These paradigms were later merged that yielded a total of 454 paradigms with support for each paradigm measured as the number of stems sharing the same endings. The pair of endings $\langle null, s \rangle$ has the maximum support of 399 stems.

For the English paradigms created in the previous step, we defined the measure of *CF* to see how many stems were common between any two paradigms. Two paradigms would be merged if they had the CF greater than a pre-defined threshold. Three threshold values taken were 0.33, 0.5, and 0.67. So any two paradigms would be merged if they had at least one third, one half, or two third of stems in common between them. With a threshold of 0.33, 35 new paradigms were created after merging. While with a threshold of 0.67 only 9 new paradigms were created (Table 5.13).

We then discovered that in many paradigms all the stems shared the same last letter, and we decided to refine those paradigms by taking that last letter out of all the stems and concatenating that letter at the beginning of all the endings in that paradigm. We also merged any pair of refined paradigms, if they shared at least a minimum portion of their stems. We merged them by taking a union of endings and stems, and putting them in a new paradigm.

Endings	Stems
ation,ations,ed,ing,null	alleg,civilis,condemn,confront,expect,generalis,...
en,est,ness,null	happi,rich,weak
ability,ed,ing,ment	agree,disappoint,employ,enlighten,punish
able,al,ed,ing,null	deferr,deni,renew
es,ing,s,y	german,part,read,risk,speed,unit,victor
able,ably,ed,s,null	favour,regrett,remark,sustain
er,ing,s	bann,clean,join,remind,waiv,warn
ation,e,es,ing,m,tic	combin,determin,realis,restor,subsidis
e,m,tic	enthusias,idealis,optimis

Table 5.13: 9 new paradigms created with a threshold of 0.67.

Four iterations were carried out for refinement of paradigms, each time using the thresholds of 0.33, 0.5, and 0.67. Interestingly, the new paradigms thus created are the same for each threshold though their number varies over the 4 iterations. In the first iteration 52 new paradigms were created, 53 in the 2nd, 55 in the 3rd and 54 in the 4th.

The new paradigms thus created also gave us a new set of words, which were not covered by the old paradigms. That can help in generalizing the paradigms and a new unseen word could be segmented by using the new paradigms and then verified by comparing against a large corpus.

We repeated the experiments for segmenting word forms into stems and endings, but this time taking into account that how much support two word forms in one language have from their translations in other languages. We only segmented word forms in any language, whose translations matched in the other three languages, and who had at least the first four letters in common. Or we can say that all the word segmentations done in any language had the support of 3. With this approach we ended up with 983 ‘paradigms’ in English, 17,156 in German, 3,667 in French and 15,853 in Greek. We only used English segmentations for

evaluation. After calculating segmentation binary codes and collapsing them, for the word form segmentations in English, we ended up with 1,483 segmentations.

We also used the algorithm based on the analogy principle to calculate segmentations for the same word forms in English. For the application of analogy principle, at each step we took four different word forms, based on the criterion that when segmented two of the four pair would have the same prefix among them and the other two would also have the same prefix between them. And also that the two stems (prefixes) would have the same two endings. We looked for all possible segmentations for each word form. Thus the total number of segmentations is relatively very high as compared to our method where only one segmentation point is taken.

Evaluation is carried out based on the fact that how many of the segmentations proposed by the analogy principle and our method, are also shared by the gold standard data. We observed that our method performed much better than the one based on the analogy principle. Recall for our method was 86% as compared to only 45% for the one based on the analogy principle. The intersection of the two, however, yields better results than either of the two with precision of almost 89%, though recall in that case goes down a bit.

Multilingual proto-synsets created earlier are in crude form and there are more proto-synsets than the number of meanings that they correspond to. That has to do with morphological and semantic variation in the synsets. If we can refine the synsets by merging ones that are morphological variations of other or where a word is a synonym of the other, the number of proto-synsets can be reduced and the word forms would start converging towards their lexical form. Such synsets can give better results on document clustering and classification tasks. The work on morphology can lead us into that direction.

CHAPTER 6

Conclusion

Here is the summarized account of work done in this thesis and the results obtained. The results have been analyzed and conclusions drawn. Section 6.1 gives the summary of the entire work. Section 6.2 outlines the contributions made. Finally, Section 6.3 outlines some of the future directions that this work can take.

6.1 Summary

This work comprises three different parts. In the first, we demonstrated that a multi-lingual lexicon can be extracted from online resources in an automated way. In the second, we showed that word aligning parallel corpora can help remove lexical ambiguity. In the third, we demonstrated that the word-aligned parallel corpus can be used to carry out morphological analysis in an unsupervised manner.

6.1.1 Automatic Generation of Multilingual Lexicons

Generation of multilingual lexicons, is a straightforward, but valuable task. In it a crawler is used to search through Wikipedia pages, using the HTML link structure. A starting point is defined but then the search is done using BFS as the search technique, where webpages are visited in the order in which they are discovered. The crawler extracts the title of the starting Wikipedia webpage in English, and puts it together with translations of the same title in other languages. The process if repeated for all the URLs explored during search.

We created both general and domain specific lexicons, covering different language families, writing styles and topics. The general dictionary covers topics ranging from politics, to sports, to religion. The domain specific dictionaries cover domains of Computer Science and Artificial Intelligence. We also used the domain specific lexicons to build relationships between different languages, based on how many concepts any two languages have in common on Wikipedia. The results were evaluated for HeptaLex: a general lexicon in seven languages viz. English, German, French, Greek, Polish, Bulgarian and Chinese. Native speakers of the languages were asked to verify the translations. Results look promising for the languages evaluated. Chinese entries were the most reliable with 97% of them being correct, while French had 93% of its entries as correct.

These lexicons can come in handy for translators and interpreters. They can also be used in the classroom environment where there are students with diverse backgrounds who may speak different languages. In order for them to discuss ideas with their class mates and to contribute productively to the topic under discussion, they need to understand others and know what words to used to express themselves. Such a single source of multilingual lexicon can provide the required vocabulary. Domain specific dictionaries are specially useful in such circumstances since the domain specific jargon does not constitute a massive vo-

cabulary, and thus domain specific lexicons are easy to build.

The category translations can also be used to build taxonomic hierarchical structures. Categories are already implicitly defined as graph structures in Wikipedia, and thus graph traversal algorithms can be used for the said purpose.

6.1.2 Extraction of Multilingual Proto-Synsets from Parallel Corpora

We used Europarl (Koehn 2002), an un-annotated parallel corpus for the automatic creation of multilingual proto-synsets. We first word aligned the parallel corpus using GIZA++ (Och & Ney 2003), pair-wise in English, German, French and Greek, with English as the pivotal language. The resultant word alignments were used to build phrases. The phrases in English with translations in other languages, constitute the proto-synsets.

We used the resultant proto-synsets as multilingual sense tags to disambiguate the original corpus in English. Since we did not have any sense disambiguated parallel corpus, we evaluated it indirectly by evaluating how good it performed at document clustering and classification tasks.

The results of this evaluation did not show any benefits of the use of multilingual synsets. We believe the reasons were twofold: firstly, the word alignment was far from perfect; secondly, the corpus contained a much greater number of word forms than lexical entries. We did not perform any morphological analysis that would collapse all such variant multilingual synsets onto a single one, and therefore could not evaluate the full potential of our idea. This was done later through the use of a dataset that did not need word alignment or morphological analysis, and therefore allowed us to measure the benefits of our approach in its pure form.

The results clearly showed that using our approach can reduce substantially

the lexical ambiguity of the corpus, if paired with efficient alignment and morphological analysis algorithms.

6.1.3 Morphological Analysis

The proto-synsets thus created are in crude form and there are lot more synsets than the meanings or concepts suggested by them. The solution for this discrepancy is to refine the proto-synsets by merging the ones that have word forms that are morphological or semantic variation of each other. Morphological variation refers to inflectional and derivational variation. While semantic variation refers to two word forms having the same sense, or in other words being synonymous.

We did some initial experimentation in this direction by calculating edit distances between all possible pairs of proto-synsets where each of them had the same word in English.

We later used these pairs of proto-synsets for word segmentation of English wordforms into a common stem and a pair of endings. These were later used to create paradigms. The paradigms were then merged together by going through each paradigm at a time, comparing it against all the others, finding one with which it had maximum number of stems in common, and then creating a new paradigm consisting of the common stems and the union of their endings. Old paradigms that were merged were removed from the list of paradigms if they lost support of any stems. These new paradigms, along with the paradigms that were used for merging and had some stems removed, were later made part of the set of paradigms created earlier.

We also carried out another series of experiments where we first refined the endings and then taking each paradigm at a time, finding another one with which it shared stems more than a certain pre-defined threshold, and merging them together by taking a union of stems and endings. This would give rise to more

generalized paradigms, which could also be used to segment unseen wordforms. The paradigms were not evaluated because we were more interested in morphology from the multilingual perspective.

We also conducted another series of experiments where we took all the original proto-synsets, and created all possible combinations of them, and finding any pair of them that overlapped in any of the three languages. That is what we called support of 3 languages. Then we segmented the wordforms in the fourth language, where the wordforms were different, into a common stem and a pair of endings. We also segmented these wordforms using another unsupervised technique based on the analogy principle.

Then we compared the two sets of segmentations with the gold standard and found that our method performed much better than the one based on the analogy principle, with a precision of 85.6% for the former as compared to 44.9% for the latter. Though recall reduced for our method since they only have one segmentation point per word form. On the other hand the analogy principle cuts a word form at many points, since it considers all possible segmentations. It is constrained by the fact that it looks for another word form, with a different prefix that share the same set of endings. Recall is high for it, but only a few of these options can be correct so precision is low.

Then we took the intersection of the segmentations produced by the two methods. We found that recall decreased a bit but precision improved even further to 88.7%. It shows that our method works well and we can use them for morphological analysis.

6.2 Contributions

The main contributions that have been made in this thesis are summarized as follows:

1. We have demonstrated how to build both general and domain specific multilingual lexicons using Wikipedia as a resource, using standard search techniques embedded in a crawler.

2. We have demonstrated how parallel corpora can be used to extract lexical semantic information in the form of multilingual synsets. We have also studied the benefits of using multilingual synsets for the WSD task. We showed that taking cues from other languages to disambiguate a sense of a polysemous word in one of the languages, can substantially reduce ambiguity. We also proposed and studied ways of evaluating multilingual synsets in the absence of a gold standard, through their use in supervised and unsupervised learning tasks.

3. We have demonstrated how multilingual synsets can be used for the unsupervised learning of morphology. We have shown that our approach outperforms substantially, by a factor of almost two, another popular unsupervised approach, and that the two can be combined in a useful way, as measured on a gold standard dataset.

6.3 Future Work

There is a lot of room for further work. This work has also lead us to think of new ideas to work on.

1. We can refine the proto-synsets by using the work we did on calculating edit distances and segmentation of word forms.
   (a) The work on calculating edit distances can be used to merge proto-synsets based on edit distances.
   (b) Edit distances can also be used along with other synonymy detection methods to find proto-synsets that have different word forms that share the same meaning. They can also be merged.
   (c) The paradigms we created can help us in figuring out the lexical forms of words and that can help us in refining the proto-synsets to the point where the word forms in them start converging to their lexical forms.
2. The paradigms we created can also be used to segment unseen words, which can then be verified by looking for their instances in a large parallel corpus.
3. We can also target the next SemEval multilingual cross-lingual word sense disambiguation task in 2013.

References

Adafre, S. & de Rijke, M. (2006). Finding Similar Sentences across Multiple Languages in Wikipedia. In *Proceedings of the EACL Workshop on New Text*.

Adler, M. & Elhadad, M. (2006). An Unsupervised Morpheme-based HMM for Hebrew Morphological Disambiguation. In *Proceedings of the ACL/CONLL*, (pp. 665–672).

Agirre, E. & Rigau, G. (1995). A Proposal for Word Sense Disambiguation using Conceptual Distance. In *Proceedings of the First International Conference on Recent Advances in Natural Language Processing*, Bulgaria.

Ahn, D., Jijkoun, V., Mishne, G., M̀uller, K., de Rijke, M., & Schlobach, S. (2004). Using Wikipedia at the TREC QA Track. In *Proceedings of TREC 2004*.

Alfred, R., Kazakov, D., Bartlett, M., & Paskaleva, E. (2007). Hierarchical Agglomerative Clustering for Cross-Language Information Retrieval. *International Journal of Translation*, *19*(1), 139–162.

Banerjee, S. & Pedersen, T. (2002). An Adapted Lesk Algorithm for Word Sense

Disambiguation Using WordNet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, (pp. 136–145).

Berton, A., Fetter, P., & Regel-Brietzmann, P. (1996). Compund Words in Large-Vocabulary German Speech Recognition Systems. In *Proceedings of The Fourth International Conference on Spoken Language Processing (ICSLP '96)*, (pp. 1165–1168)., Philadelphia, PA, USA.

Black, P. (2006). Dictionary of Algorithms and Data Structures. `http://www.nist.gov/dads/HTML/manhattanDistance.html`.

Braschler, M. & Schäuble, P. (1998). Multilingual Information Retrieval based on Document Alignment Techniques. In Nikolaou, C. & Stephanidis, C. (Eds.), *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL '98)*, (pp. 183–197)., London.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. California: Wadsworth International.

Brin, S. & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th International World Wide Web Conference*, (pp. 110–117)., Brisbane, Australia. Elsevier Science.

Brown, P., Cocke, J., Pietra, S., Pietra, V., Jelinek, F., Lafferty, J., Mercer, R., & Roossin, P. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, *16*(2), 79–85.

Brown, P., Pietra, S., Pietra, V., & Mercer, R. Word-Sense Disambiguation Using Statistical Methods. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, (pp. 264–270).

Brown, P., Pietra, S., Pietra, V., & Mercer, R. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, *19*(2), 263–311.

Brown, P., Pietra, V., deSouza, P., Lai, J., & Mercer, R. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, *18*(4), 467–479.

Buckley, C. (1985). Implementation of the SMART Information Retrieval System. Technical report 85-686. Cornell University.

Buckley, C. & Voorhees, E. (2000). Evaluating Evaluation Measure Stability. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, (pp. 33–40)., Trento, Italy.

Bunescu, R. & Pasca, M. (2006). Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, (pp. 9–16)., Trento, Italy.

Burnard, L. & Aston, G. (1998). The BNC Handook: Exploring the British National Corpus. Edinburgh: Edinburgh University Press.

Can, B. & Manandhar, S. (2009). Unsupervised Learning of Morphology by using Syntactic Categories. In *Working Notes CLEF 2009 Workshop*.

Charitakis, K. (2007). Using Parallel Corpora to Create a Greek-English Dictionary with Uplug. In *Proceedings of the 16th Nordic Conference on Computational Linguistics - NODALIDA'07*.

Chew, P., Bader, B., Kolda, T., & Abdelali, A. (2007). Cross-Language Information Retrieval Using PARAFAC2. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Church, K., Gale, W., Hanks, P., & Hindle, D. (1991). *Lexical Acquisition: Using On-Line Resources to Build a Lexicon*, volume 4592, chapter Using Statistics in Lexical Analysis. Lawrence Erlbaum.

Clark, A. (2000). Inducing Syntactic Categories by Context Distribution Clustering. In *The Fourth Conference on Natural Language Learning (CoNLL)*, (pp. 91–94).

Collins, M. & Brooks, J. (1995). Prepositional Phrase Attachment through a Backed-Off Model. In *Third Workshop on Very Large Corpora, Association for Computational Linguistics, ACL*.

Creutz, M. & Lagus, K. (2007). Unsupervised Models for Morpheme Segmentation and Morphology Learning. *ACM Transactions on Speech and Language Processing, 4(1)*.

Dagan, I. & Itai, A. (1994). Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics, 20*, 563–596.

Dagan, I., Itai, A., & Schwall, U. (1991). Two Languages are More Informative than One. *ACL, 29*, 130–137.

Dasgupta, S. & Ng, V. (2007). Unsupervised Part-of-Speech Acquisition for Resource-Scarce Languages. In *Proceedings of the EMNLP-CoNLL*, (pp. 218–227).

Davies, D. & Bouldin, D. (1979). A Cluster Separation Measure. *IEEE Transactions and Pattern Analysis and Machine Intelligence, 1/2*, 224–227.

de Saussure, F. (1959). *Course in General Linguistics*. Philosophical Library, New York.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*.

Diab, M. (2000). An Unsupervised Method for Multilingual Word Sense Tagging using Parallel Corpora: A Preliminary Investigation. In *ACL-2000 Workshop on Word Senses and Multilinguality*, (pp. 1–9)., Hong Kong.

Diab, M. & Resnik, P. (2002). An Unsupervised Method for Word Sense Tagging

using Parallel Corpora. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dimitrova, L., Ide, N., Petkevic, V., Erjavec, T., Kaalep, H., & Tufis, D. (1998). Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and Eastern European Languages. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1 (COLING '98)*, Stroudsburg, PA, USA.

Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc.

Fellbaum, C. (1998). *WordNet An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Ferrández, S., Toral, A., Ferrández, O., Ferrández, A., & Muñoz, R. (2007). *Lecture Notes in Computer Science*, volume 4592, chapter Applying Wikipedia's Multilingual Knowledge to Cross-Lingual Question Answering. Springer.

Fišer, D. (2007). Leveraging Parallel Corpora and Existing WordNets for Automatic Construction of the Slovene Wordnet. In *Proceedings of (L&TC 2007)*, Poznań, Poland.

Francis, W. (1964). A Standard Sample of Present-Day English for use with Digital Computers. Report to the U.S. Office of Education on Cooperative Research Project No. E-007.

Fung, P. & Wu, D. (1995). Coerced Markov Models for Cross-Lingual Lexical-Tag Relations. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, (pp. 240 – 255)., Leuven, Belgium.

Gabrilovich, E. & Markovitch, S. (2006). Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In *Association for the Advancement of Artificial Intelligence, AAAI'06*.

Gale, W., Church, K., & Yarowsky, D. (1992). A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, *26*(5-6), 415–439.

Giles, J. (2005). Internet Encyclopaedias go Head to Head. *Nature, 438(7070):900-901*.

Goldsmith, J. (2001). Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics, 27(2):153-198*.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences*. Cambridge: The Press Syndicate of the University of Cambridge.

Harris, Z. (1955). From Phoneme to Morpheme. *Language*, *31(2)*.

Hartigan, J. (1975). *Clustering Algorithms (Probability & Mathematical Statistics)*. Cambridge: John Wiley & Sons Inc.

Hull, D. & Grefenstette, G. (1996). Experiments in Multilingual Information Retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jain, A. & Dubes, R. (1988). *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall.

Jardino, M. & Adda, G. (1993). Automatic Word Classification Using Simulated Annealing. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, volume 2, (pp. 41 – 44)., Minneapolis.

Jones, G., Fantino, F., Newman, E., & Zhang, Y. (2008). Domain-Specific Query Translation for Multilingual Information Access using Machine Translation Augmented With Dictionaries Mined from Wikipedia. In *Proceedings of the 2nd International Workshop on Cross Lingual Information Access Addressing the Information Need of Multilingual Societies*, Hydrabad, India.

Kaplan, A. (1950). An Experimental Study of Ambiguity in Context. *Mechanical Translation*, *1*(1-3).

Katz, S. (1987). Estimation of Probabilities for Sparse Data for the Language Model Component of a Speech Recogniser. *IEEE Transactions on Acoustics, Speech and Signal Processing, 35(3)*, 400–401.

Kawaba, M., Nakasaki, H., Utsuro, T., & Fukuhara, T. (2008). Cross-Lingual Blog Analysis based on Multilingual Blog Distillation from Multilingual Wikipedia Entries. In *Proceedings of International Conference on Weblogs and Social Media, ICWSM'08*.

Kazakov, D. (2000). Achievements and Prospects of Learning Word Morphology with Inductive Logic Programming. In Cussens, J. & Dzeroski, S. (Eds.), *Learning Language in Logic*, (pp. 89–109). Springer.

Kazakov, D., Cussens, J., & Manandhar, S. (2006). On The Duality of Semantics and Syntax: The PP Attachment Case. Technical report YCS 409. Department of Computer Science, University of York, UK.

Kazakov, D. & Manandhar, S. (2001). Unsupervised Learning of Word Segmentation Rules with Genetic Algorithms and Inductive Logic Programming. *Machine Learning, 43*(1-2), 121–162.

Kazakov, D. & Shahid, A. (2008). Extracting Multilingual Dictionaries for the Teaching of CS and AI. In *4th UK Workshop on AI in Education*.

Kilgarriff, A. & Rosenzweig, J. (1999). Framework and Results for English Senseval. *Computers and the Humanities, 34(1)*, 15–48.

Koehn, P. (2002). Europarl: A Multilingual Corpus for Evaluation of Machine Translation. `http://www.isi.edu/~koehn/publications/europarl/`.

Kvålseth, T. (1987). Entropy and Correlation: Some Comments. *IEEE Transactions on Systems, Man and Cybernetics, SMC-17*, 517–519.

Lancaster, F. (1968). *Information Retrieval Systems: Characteristics, Testing and Evaluation*. New York: Wiley.

Landauer, T., Foltz, P., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes*, *25*, 259–284.

Landauer, T. K. & Littman, M. L. (1990). Fully Automatic Cross-Language Document Retrieval using Latent Semantic Indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, (pp. 31–38).

Lee, L. (1999). Measures of Distributional Similarity. In *Proceedings of the 37th ACL*.

Lefever, E. & Hoste, V. (2009). SemEval-2010 Task 3: Cross-Lingual Word Sense Disambiguation. In *Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, (pp. 82–87)., Boulder, Colorado.

Lefever, E. & Hoste, V. (2010a). Construction of a Benchmark Data Set for Cross-lingual Word Sense Disambiguation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Malta.

Lefever, E. & Hoste, V. (2010b). SemEval-2010 Task 3: Cross-Lingual Word Sense Disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, (pp. 15–20)., Uppsala, Sweden.

Lesk, M. (1986). Automatic Sense Disambiguation using Machine Readable Dictionaries: How to tell a Pine Cone from a Ice Cream Cone. In *Proceedings of SIGDOC'86*.

Levenstein, V. (1966). Binary Codes Capable of Correcting Insertions and Reversals. *Soviet Physics Doklady*, *10*, 707.

Li, X., Szpakowicz, S., & Matwin, S. (1995). A WordNet-based Algorithm for Word Sense Disambiguation. In *Proceedings of IJCAI-95*, (pp. 1368–1374)., Montreal, Canada.

Luhn, H. (1959). The Automatic Creation of Literature Abstracts. *IBM Journal*

of Research and Development, 2, 159–165.

Manning, C. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics, 19(2)*.

Martin, S., Liermann, J., & Ney, H. (1998). Algorithms for Bigram and Trigram Word Clustering. *Speech Communication, 24(1)*, 19 – 37.

McEnery, A. (2003). *Lecture Notes in Computer Science*, chapter Corpus Linguistics, (pp. 448–463). Oxford University Press.

Mihalcea, R. (2007). Using Wikipedia for Automatic Word Sense Disambiguation. In *In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York.

Mihalcea, R. & Moldovan, D. (1999). A Method for Word Sense Disambiguation of Unrestricted Text. In *Proceedings of the 37th Meeting of ACL*, College Park, MD.

Miller, D., Leek, T., & Schwartz, R. (1999). A Hidden Markov Model Information Retrieval System. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 214–221)., Berkeley, California, United States.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Introduction to WordNet: An On-line Lexical Database. *Journal of Lexicography, 3(4):235-244*.

Mitchell, T. (1997). *Machine Learning*. MIT Press and McGraw-Hill.

Najork, M. & Wiener, J. (2001). Breadthfirst Crawling Yields High-quality Pages. In *Proceedings of the 10th International Conference on World Wide*

Web.

Ng, H., Wang, B., & Chan, Y. (2007). Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, (pp. 455–462)., Sapporo, Japan.

Oard, D. & Dorr, B. (1996). A Survey of Multilingual Text Retrieval. Technical report. University of Maryland at College Park College Park, MD, USA.

Och, F. (1999). An Efficient Method for Determining Bilingual Word Classes. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics (EACL '99)*, Bergen, Norway.

Och, F. & Ney, H. (2000). Improved Statistical Alignment Models. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, Hong Kong.

Och, F. & Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics, 29(1):19-51*.

Pianta, E., Bentivogli, L., & Girardi, C. (2002). MultiWordNet: Developing an Aligned Multilingual Database. In *Proceedings of the First International Conference on Global WordNet*, Mysore, India.

Pirelli, V. (1993). Morphology, Analogy and Machine Translation. PhD Thesis. Salford University, UK.

Pirkola, A. (1998). The Effects of Query Structure and Dictionary Setups in Dictionary-Based Cross-Language Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, New York, USA.

Potthast, M., Stein, B., & Anderka, M. (2008). Wikipedia-based Multilingual Retrieval Model. In *Proceedings of the 30th European Conference on IR Research, ECIR'08*, Glasgow.

Quinlan, J. (1986). Induction of Decision Trees. *Machine Learning, 1(1)*, 81–106.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Richman, A. & Schone, P. (2008). Mining Wiki Resources for Multilingual Named Entity Recognition. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL'08*, (pp. 1–9)., Columbus, Ohio.

Robertson, S. (2006). On GMAP and Other Transformations. In *CIKM '06 Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, New York, USA.

Ruiz-Casado, M., Alfonseca, E., & Castells, P. (2005). Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets. In *Proceedings of Advances in Web Intelligence*, Lodz, Poland.

Russell, S. & Norvig, P. (1995). *Artificial Intelligence*. Prentice Hall.

Sagot, B. & Fišer, D. (2008). Building a Free French WordNet from Multilingual Resources. In *Proceedings of OntoLex 2008*, Marrackech.

Salton, G. (1970). Automatic Processing of Foreign Language Documents. *Journal of the American Society for Information Science, 21*, 187–194.

Salton, G. (1989). *Automatic Text Processing*. Addison-Wesley.

Salton, G., Wong, A., & Yang, C. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM, 18(11)*, 613–620.

Sato, S. (2009). Crawling English-Japanese Person-Name Transliterations from the Web. In *Proceedings of the 18th International Conference on World Wide Web*, Madrid, Spain.

Schäuble, P. (1997). *Multimedia Information Retrieval*. Kluwer Academic Publishers.

Schone, P. & Jurafsky, D. (2000). Knowledge-free Induction of Morphology

using Latent Semantic Analysis. In *Proceedings of the CoNLL*, (pp. 67–72).

Sedding, J. & Kazakov, D. (2004). WordNet-Based Text Document Clustering. In *3rd Workshop on Robust Methods in Analysis of Natural Language Data (ROMAND)*, Geneva, Switzerland.

Shahid, A. & Kazakov, D. (2009). Automatic Multilingual Lexicon Generation using Wikipedia as a Resource. In *Proceedings of the International Conference on Agents and Artificial Intelligence, ICAART*, Porto, Portugal.

Shahid, A. & Kazakov, D. (2010). Retrieving Lexical Semantics from Multilingual Corpora. *Polibits, 5*, 25–28.

Shahid, A. & Kazakov, D. (2011). Using Multilingual Corpora to Extract Semantic Information. In *Proceedings of the Symposium on Learning Language Models from Multilingual Corpora, AISB'11 Convention*, York, UK.

Sheridan, P. & Ballerini, J.-P. (1996). Experiments in Multilingual Information Retrieval using the SPIDER System. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 58–65).

Snyder, B. & Barzilay, R. (2008). Unsupervised Multilingual Learning for Morphological Segmentation. In *The Annual Conference of the Association for Computational Linguistics*.

Sparck Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Applications in Retrieval. *Journal of Documentation, 28(1)*, 11–21.

Specia, L., Nunes, M., & Stevenson, M. (2005). Eploiting Parallel Texts to Produce a Multilingual Sense Tagged Corpus for Word Sense Disambiguation. In *Proceedings of the Conference on Recent Advances on Natural Language Processing (RANLP-2005)*, Borovets, Bulgaria.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A Comparison of Document Clustering Techniques. In *6th ACM SIGKDD, World Text Mining Conference*,

Boston, MA, USA.

Strehl, A. (2002). Relationbship-based Clustering and Cluster Ensembles for High-dimensional Data Mining. PhD Thesis. The University of Texas at Austin.

Talvensaari, T., Juhola, M., Laurikkala, J., & Järvelin, K. (2007). Corpus-based Cross-language Information Retrieval in Retrieval of Highly Relevant Documents. *Journal of the American Society for Information Science and Technology*, *58(3)*, 322–334.

Tiedemann, J. (1999). Uplug - a Modular Corpus Tool for Parallel Corpora. In *In the Parallel Corpus Symposium (PKS99)*, Uppsala University, Sweden.

Tiedemann, J. (2004). The OPUS Corpus - Parallel & Free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.

Tufis, D. (2000). Design and Development of a Multilingual Balkan WordNet. *Romanian Journal of Information Science and Technology Special Issue*, *7:1-2*.

Tufis, D., Ion, R., & Ide, N. (2004). Fine-Grained Word Sense Disambiguation based on Parallel Corpora, Word Alignment, Word Clustering, and Aligned WordNets. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, *59*, 433–460.

Tyers, F. & Pienaar, J. (2008). Extracting Bilingual Word Pairs from Wikipedia. In *Proceedings of the SALTMIL Workshop at Language Resources and Evaluation Conference, LREC'08*, (pp. 19–22).

van den Bosch, A., Daelemans, W., & Weijters, T. (1996). Morphological Analysis as Classification: an Inductive-Learning Approach. In *Proceedings of the Second International Conference on New Methods in Language Processing*

(NeMLap-2), (pp. 79–89)., Bilkent University, Ankara, Turkey.

van der Plas, L. & Tiedemann, J. (2006). Finding Synonyms using Automatic Word Alignment and Measures of Distributional Similarity. In *Proceedings of ACL/COLING 2006*, Sydney, Australia.

van Rijsbergen, C. (1979). *Information Retrieval*. Butterworth-Heinemann.

Voorhees, E. & Harman, D. (1999). Overview of the Seventh Text REtrieval Conference (TREC-7). In Voorhees, E. & Harman, D. (Eds.), *In NIST Special Publication 500-242*, (pp. 1–23).

Vossen, P. (1996). Right or Wrong: Combining Lexical Resources in the EuroWordNet Proejct. In *Proceedings of Euralex-96 International Congress*.

Vossen, P. (1998). *EuroWordNet: a Multilingual Database with Lexical Semantic Networks for European Languages*. Kluwer.

Wagner, R. & Fischer, M. (1974). The String-to-String Correction Problem. *Journal of the Association for Computing Machinery, 21:1*, 168–173.

Wong, W. & Fu, A. (2000). Incremental Document Clustering for Web Page Classification. In *IEEE 2000 International Conference on Information Society in 21st Century: Emerging Technologies and New Challenges*.

Xu, J., Weischedel, R., & Nguyen, C. (2001). Evaluating a Probabilistic Model for Cross-Lingual Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*, New York, USA.

Yarowsky, D. (1992). Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, (pp. 454–460)., Nantes, France.

Yarowsky, D. (1994). Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In *Proceedings of the*

32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivalling Supervised Methods. *ACL, 33*, 189–196.

Young, P. (1994). Cross Language Information Retrieval Using Latent Semantic Indexing. Master's Thesis. University of Knoxwille, Tennessee: Konxwille.

Yvon, F. (1996). Prononcer par analogies: motivations, formalisations et évaluations. PhD Thesis. ENST Paris, France.

Zesch, T. Muller, C. & Gurevych, I. (2008). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.

Zesch, T., Gurevych, I., & Muhlhauser, M. (2007). Comparing Wikipedia and German Wordnet by Evaluating Semantic Relatedness on Multiple Datasets. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT'07*, (pp. 205–208).