Enriching Lexical Knowledge Bases with Encyclopedic Relations

by

Samuel Fernando

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

Department of Computer Science
University of Sheffield
March 2013

Abstract

Lexical knowledge bases, such as WordNet, have been shown to be useful in a wide range of language processing applications. However WordNet lacks certain information, such as topical relations between synsets. This thesis addresses this problem by enriching WordNet using information derived from Wikipedia.

The approach consists of mapping concepts in WordNet to corresponding articles in Wikipedia. This is done using a three stage approach. First a set of possible candidate articles is retrieved for each WordNet concept. This is done by searching using the article title, and also by searching the full text using an IR engine. Secondly, text similarity scores are used to select the best match from the candidate articles. Finally, the mappings are refined using information from Wikipedia links to give a set of high quality matches.

The mappings are evaluated using a manually annotated gold standard set of synset-article mappings. The annotation process indicates that the majority of synsets have a good matching article. The refined mappings are shown to have precision of 88.2%.

The mappings are then used to enrich relations in WordNet using Wikipedia links. The enriched WordNet is then used with a knowledge based Word Sense Disambiguation system. Evaluations are performed on the Semcor 3.0 corpus. Adding the new relations improves performance significantly over the WordNet baseline, demonstrating the usefulness of the mappings on an extrinsic task.

Acknowledgements

With thanks to my supervisor Mark Stevenson, and all the others within the department who gave advice and support: Mark Greenwood, Adam Funk, Kumutha Swampillai, Jon Barker, Rob Gaizauskas, and Simon Foster, among others.

Also thanks to my family who were invaluable throughout: especially my parents Luke and Winnie Fernando, and my cousin Chrisantha Fernando.

Contents

Chapter 1

Introduction

This thesis discusses methods for enriching the lexical knowledge base (LKB) WordNet with new relations by linking with the online encyclopedia Wikipedia. LKBs are resources which classify and index words with their senses and the connections that exists between them. Software implementations of LKBs allow this information to be efficiently stored and retrieved so that it can be processed readily by computer programs. LKBs have been used successfully in a wide variety of language processing tasks. WordNet in particular is the most widely used LKB in current research because of its free availability and wide coverage. It has been used for a broad range of language processing tasks including information extraction (Bagga et al., 1997), semantic search (Benassi et al., 2004), semantic annotation (Fellbaum et al., 2001), information retrieval (Flank, 1998), question answering (Harabagiu and Moldovan, 1996), natural language generation (Hongyan, J., 1998), sentence similarity (Li et al., 2006), query expansion (Voorhees, 1994), text summarisation (Carenini et al., 2008), textual entailment (Zanzotto and Moschitti, 2006), and word sense disambiguation (Agirre and Soroa, 2009).

Although WordNet has been widely used there are a number of recognized shortcomings. It has been noted that WordNet senses are too fine-grained, sometimes difficult even for humans to distinguish (Navigli, 2006). Another issue is that it is difficult for WordNet to keep up with new words which enter into common usage. Also, while WordNet covers a range of semantic relations, such as hypernymy, meronymy and synonmy, there are no topical relations; for example there is no link between concepts such as "tennis" and "racket" despite their relatedness. This has become informally known as the 'tennis' problem.

This thesis tests whether Wikipedia can be used to enrich WordNet with useful new relations between concepts. Wikipedia is a freely-editable encyclopedia which has become hugely popular since its launch in 2001, with over 3 million articles on a wide range of topics. The freely-accessible nature of Wikipedia has naturally raised concerns over the quality of articles. However studies have shown the quality of scientific articles in Wikipedia is comparable with that of the Encyclopedia Britannica - a well-established, proprietary encyclopedia written by expert contributors (Giles, 2005). Wikipedia is rich with topical links and category annotations, and generally of a higher quality than Web text. Thus information extracted from Wikipedia may go some way to addressing the 'tennis problem' since Wikipedia is rich with topical and other relations. Specifically the thesis tested is whether WordNet synsets can be successfully mapped onto Wikipedia articles. Once articles are mapped in this way, the aim is to show whether new relations derived from Wikipedia links can be used to enrich WordNet with useful new topical relations.

1.1 Novel contribution

There have been previous attempts to enrich knowledge bases such as WordNet automatically using corpora (see Section 2.2). Recently, several attempts have specifically made use of Wikipedia for this task (Ruiz-Casado et al., 2005; Suchanek et al., 2007; Ponzetto and Navigli, 2010). However this thesis offers the following novel contributions:

- A new approach to matching WordNet synsets to Wikipedia articles. Previous work has mapped articles to word senses (Ruiz-Casado et al., 2005; Ponzetto and Navigli, 2010). The approach described here performs the mapping in the other direction. This is a substantially different problem since Wikipedia is a much larger resource than WordNet (i.e. there are many more Wikipedia articles than WordNet synsets). The reasoning is that it is better to find the best article for each given synset rather than vice versa, since the aim is to enrich WordNet; and also because a much larger proportion of WordNet synsets will be mapped to Wikipedia articles than vice versa, partly due to the large difference between the sizes of the two resources. This is the first time the mapping has been attempted in this direction, to the author's knowledge.
- Novel methods are used to refine the mappings creating a smaller but more precise set of mappings.
- The manually annotated data set provides a useful evaluation resource and analysis of this set gives insight into the overlap between WordNet concepts and Wikipedia articles.
- The full WordNet-Wikipedia mappings are made available online, providing useful data for future research. The enriched WordNet using relations derived from Wikipedia links is also available online.

The work in this thesis has led to two publications (Fernando and Stevenson, 2010, 2012).

1.2 Structure of thesis

The rest of this thesis follows this structure:

- **Chapter 2** gives background to the experimental work in this thesis, including an overview of WordNet, Wikipedia, and previous related work on aligning knowledge bases with other resources.

- **Chapter 3** describes the methods for mapping WordNet synsets to Wikipedia articles. This is done using a three stage approach: first candidate articles are retrieved for each synset; then the best article mapping is selected from the candidate article set; and finally global refinements are used to eliminate incorrect mappings and improve precision.

- **Chapter 4** describes the creation of a gold standard manually annotated test set used for evaluating the mapping methods and describes how the methods are evaluated against this gold standard set.

- **Chapter 5** gives the results of experiments using the methods of Chapter 3 with the evaluation approaches in Chapter 4.

- **Chapter 6** then uses the synset-article mappings to enrich WordNet with new relations. This enriched WordNet is then used as a knowledge base for a word sense disambiguation system, which is evaluated on Semcor 3.0 and the Semeval 2007 coarse-grained task.

- Finally **Chapter 7** summarises the conclusions of the thesis and its contributions, and describes possible directions for future work in this area.

Chapter 2

Background

This chapter provides a review of relevant work from the literature. Section 2.1 gives an overview of lexical knowledge bases (LKBs), focussing on WordNet as the most widely used in language processing research. Comparison is made with other machine-processable knowledge bases and ontologies (e.g. CYC, LDOCE). The section also describes previous work using WordNet in language processing applications. Section 2.2 gives an overview of previous work on automatically enriching knowledge bases such as WordNet. Section 2.3 describes the online encyclopedia Wikipedia including previous work deriving machine-processable knowledge from Wikipedia, and work linking Wikipedia to WordNet. Finally, Section 2.4 summarizes the chapter.

2.1 Lexical knowledge bases

This section examines some of the most widely used LKBs. Various dictionaries and lexical databases have been used for language processing applications including the Oxford Advanced Learners Dictionary of Current English (Lesk, 1986), Collin's English Dictonary (Veronis and Ide, 1990), LDOCE (Boguraev and Briscoe,

1987) and Roget's thesaurus (Jarmasz and Szpakowicz, 2000). However WordNet (Fellbaum, 1998) is by far the most predominantly used for various reasons, including that it is freely licensed and specially designed for machine processing. WordNet can also be viewed as a kind of ontology and thus can be compared to other large ontologies such as for example the general purpose ontology CYC (Lenat, 1995).

2.1.1 WordNet

WordNet is a large lexical database of English (Fellbaum, 1998). The lexicon consists of synsets (short for synonym sets) which group together lexical items (nouns, adjectives, verbs, and adverbs) which are considered synonymous. This thesis focuses on noun synsets, since they have the greatest overlap with Wikipedia. An example noun synset is {car, auto, automobile, machine, motorcar}. Additionally each synset contains a short written definition or *gloss*. For the car synset, the gloss is "a motor vehicle with four wheels; usually propelled by an internal combustion engine". There is often also an example of the concept in a short sentence "he needs a car to get to work".

The words are referred to as *lemmas* since they comprise only the lemmatized root forms for each word (except irregular formations which are stored separately). When a user searches for a term, the WordNet system attempts to deduce the root form, so for example 'cars' becomes 'car'. Compound words (with whitespaces) are also allowed within synsets, such as 'railroad car' or 'elevator car' which appear in other synsets also containing the word 'car'. A given word may appear in more than one synset - these capture the different senses of polysemous words, as for the 'car' example.

In addition to the synsets in WordNet, there are also defined *relationships* between different synsets. The relationships between noun synsets include hypernymy, meronymy and others. Hypernyms are *is-a* relations, where A is a hypernym of B if B is a kind of A, so for example 'canine' is a hypernym of 'dog'. Meronyms are *part-of* relations, so A is a meronym of B if A is a part of B, for example 'finger' is a meronym of 'hand'. The hypernym relations serve to organize the synsets into a hierarchy, with very general concepts at the top level, and specific concepts and instances at the lowest level leaf nodes. The hypernym chain for the noun 'asthma' (which has only one sense in WordNet) is as follows:

> asthma → respiratory disease → disease → illness → ill health → pathological state → physical condition → condition → state → attribute → abstraction → entity

2.1.2 Other lexical knowledge bases

Other lexical databases which have been used in language processing applications are LDOCE (Procter, 1978) and Roget's Thesaurus (Chapman, 1992).

LDOCE was designed as a learners' dictionary. Entries are grouped into homographs which are each divided into sense definitions. Here are extracts from LDOCE for the entries for the word 'bank':

- bank *n* **1** land along the side of a river, lake, etc. **2** earth which is heaped up in a field or garden ... ...
- bank *n* **1** a place where money is kept and paid out on demand, and where related activities go on. **2** *(usu. in comb.)* a place where something is held ready for use, esp organic product of human origin for medical use: *Hospital bloodbanks have saved many lives...*

WordNet does not make a distinction between homographs and senses, so all the above senses are classed under different synsets with no information about homographs. However LDOCE as a dictionary does not contain semantic relationships as found in WordNet.

Roget's Thesaurus (Roget) is the most widely known thesaurus. In contrast to a dictionary, a thesaurus is organized as a hierarchy of concepts with abstract concepts at the top down to instances and more specific concepts at the leaf nodes. In Roget there are 15 top level classes such as 'Science and Technology' and 'The Body and the Senses'. Each of the top level classes contains a set of large categories which are subdomains of the class. Within each category is a set of paragraphs ordered by parts of speech. Roget does not contain definitions and examples therefore cannot be used as a dictionary or lexicon.

Since WordNet is also organized as a hierarchy it functions as a thesaurus in a similar fashion to Roget. As it also contains definitions and examples for words it has the advantage over LDOCE and Roget of being both a dictionary and a thesaurus.

Apart from these advantages in content and structure, WordNet has become one of the most widely used lexicons in language processing for other, more pragmatic reasons. It is much easier for machines to process the information in WordNet than in Roget since WordNet was specifically designed for this purpose from the beginning. In addition, WordNet has always been freely available, where there have always been licensing issues with many other lexical resources including Roget (Jarmasz and Szpakowicz, 2000) and LDOCE.

2.1.3 WordNet as an ontology

WordNet contains entities and relationships, and therefore it is often referred to as an ontology. In computer science, the ontology for a particular domain represents the entities and relationships within that domain in such a way to allow reasoning (Gruber, 1993). The domain for many published ontologies is a specific area of interest over which reasoning applications are desired (for example genomics or earth science). However efforts have been made to create a comprehensive general ontology with CYC (Lenat, 1995). The main objective of the CYC project was to encode core commonsense knowledge into an ontology. CYC contains thousands of different relation types, while WordNet contains only a handful of semantic relations, such as synonymy, hypernymy and meronymy. The complexity of CYC means that many inference steps are intractable. Overall the focussed nature of WordNet has made it more readily usable for NLP-intensive tasks while CYC has found favour for semantic web and information retrieval applications.

Unlike word sense repositories, ontologies contain terms which do not necessarily have a natural lexicalisation. So for example there are several synsets in WordNet containing the word ‘bank’ of which one is the synset describing the concept of a financial institution. There is no exact equivalent of this concept in CYC, but the most similar term is ‘BankingOrFinanceCompany’ which does not correspond to a word that would be naturally used in text or speech. This illustrates the subtle distinction between a lexical database (like WordNet) and other more general ontologies.

2.1.4 Use in language processing applications

There has been a large body of work making use of WordNet in various language processing applications. This section gives a brief overview of recent work for several types of applications.[1]

Lexical similarity metrics

The purpose of lexical similarity metrics is to give a quantitative measure of the similarity of two word senses. Measures of similarity can be based on information in

[1] More information on related projects can be found at the web site for the Global WordNet association `http://www.globalwordnet.org/` or from the WordNet web site `http://wordnet.princeton.edu/`

a *is-a* hierarchy or other information such as the definitions of the senses. We consider 'car' and 'boat' to be more similar to each other than 'boat' and 'tree' since 'car' and 'boat' have a more specific common ancestor, the 'vehicle' concept. WordNet only contains *is-a* hierarchies for verbs and nouns, so similarities can only be found where both words are in one of these categories, for example the nouns 'dog' and 'cat', and the verbs 'run' and 'walk'. However concepts can be related in many ways apart from being similar to each other. These include *part-of* relationships ('wheel' and 'car'), as well as opposites ('night' and 'day') and so on. Measures of *relatedness* make use of this additional, non-hierarchal information in WordNet, including the gloss of the synset. As such they can be applied to a wider range of concept pairs including words that are from different parts of speech, for example 'murder' and 'gun'.

The *lesk* metric (Banerjee and Pedersen, 2003) uses the glosses of the two words and measures relatedness as a function of the overlaps between these definitions. For example, the concepts 'drawing paper' and 'decal' have the glosses 'paper that is specially prepared for use in drafting' and 'the art of transferring designs from specially prepared paper to a wood or glass or metal surface' respectively.

The similarity of two glosses is computed by the function $score(G_1, G_2)$ which works by finding the longest overlapping sequence of words between the sentences that does not start or end with a function word (pronoun, preposition, article or conjunction). In the above examples this would be 'specially prepared'. The score given to an overlap is n^2 where n is the length of the sequence, so this two-word sequence would have a score of 4. The algorithm then removes this sequence from both texts and then finds the longest remaining subsequence, and accumulates the score. This continues until there are no remaining overlaps.

The *lesk* metric also takes into account all concepts which are directly related to the concept via explicit relations in WordNet (hypernyms, hyponyms etc.). $RELS$ is defined as a subset of relations in WordNet. For each relation, a function is defined of the same name which returns the gloss of the synset related to the synset by that relation. If more than one synset is returned the glosses are concatenated and returned. So for example $hype(A)$ returns the gloss of the hypernyms of A.

$RELPAIRS$ is defined as a closed reflexive set of pairs of relations:

$$\begin{aligned} RELPAIRS = \{(R_1, R_2) \mid R1, R2 \in RELS;\ \text{if } (R_1, R_2) \in RELPAIRS \\ \text{then } (R_2, R_1) \in RELPAIRS\} \end{aligned} \tag{2.1}$$

The reflexive constraint is imposed to ensure that the relatedness function is itself reflexive so that $relatedness(A, B) = relatedness(B, A)$

Finally, the relatedness of two synsets A and B is given by

$$relatedness(A, B) = \sum_{\forall (R_1,R_2) \in RELPAIRS} score(R_1(A), R_2(B)) \tag{2.2}$$

For example, if the set of relations RELS = {gloss, hypo, hype} and RELPAIRS = {(gloss, gloss), (hypo, hypo), (hype, hype), (gloss, hype), (hype, gloss)} then:

$$\begin{aligned} relatedness(A, B) = score(gloss(A), gloss(B)) + score(hypo(A) + hypo(B)) + \\ score(hype(A) + hype(B)) + score(gloss(A) + hype(B)) + \\ score(hype(A) + gloss(B)) \end{aligned} \tag{2.3}$$

The *lch* metric (Leacock and Chodorow, 1998) determines the similarity of two nodes by finding the path length between them in the *is-a* hierarchy. The similarity is computed as:

$$sim_{lch} = -log\frac{N_p}{2D} \tag{2.4}$$

where N_p is the distance between the nodes and D is the maximum depth in the *is-a* taxonomy.

The remainder of the methods use the notions of least common subsumers (LCS) and information content (IC).

Given two concept nodes $C1$ and $C2$ in a *is-a* hierarchy, the LCS (Wu and Palmer, 1994) is defined as the most specific node which both share as an ancestor. For example if $C1$ was 'car' and $C2$ was 'boat', then the LCS would be 'vehicle'. This is illustrated in Figure 2.1.

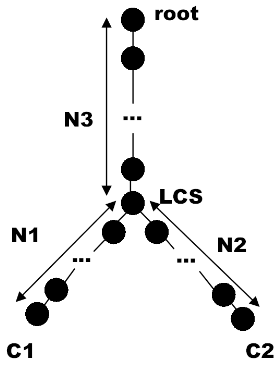


Figure 2.1: Part of a WordNet *is-a* hierarchy illustrating the LCS of two concepts $C1$ and $C2$.

The information content (Resnik, 1995) of a node is an estimate of how informative the concept is. Concepts which are more general or which occur frequently are deemed to have low information content, while concepts which are specific or occur rarely are defined as having a high information content. Formally the information content of a concept c is defined as:

$$IC(c) = -logP(c) \tag{2.5}$$

where $P(c)$ is the probability of finding c in a large corpus.

The *wup* metric (Wu and Palmer, 1994) computes the similarity of the nodes as a function of the path length from the LCS of the nodes.

The similarity between nodes $C1$ and $C2$ is:

$$sim_{wup} = \frac{2 * N3}{N1 + N2 + 2 * N3} \tag{2.6}$$

where $N1$ is the number of nodes on the path from the LCS to $C1$, $N2$ is the number of nodes on the path from the LCS to $C2$, and $N3$ is the number of nodes on the path from the root node to the LCS. These are shown in Figure 2.1.

The *resnik* metric (Resnik, 1995) uses the information content of the LCS of the two concepts. The idea is that the amount of information two concepts share will indicate the degree of similarity of the concepts, and the amount of information the two concepts share is indicated by the information content of their LCS.

Formally:

$$sim_{res} = IC(LCS) \tag{2.7}$$

The *lin* metric (Lin, 1998) builds on the *resnik* measure by normalising using the information content of the two nodes themselves.

$$sim_{lin} = \frac{2 * IC(LCS)}{IC(N1) + IC(N2)} \tag{2.8}$$

The *jcn* metric (Jiang and Conrath, 1997) also uses the information content idea:

$$sim_{jcn} = \frac{1}{IC(N1) + IC(N2) - 2 * IC(LCS)} \tag{2.9}$$

All of the above similarity metrics have been packaged together as set of Perl modules in the WordNet::Similarity package. Budanitsky and Hirst (2006) evaluate several of these similarity metrics on gold-standard data and also on an external NLP task (detecting spelling errors). Recent work has enriched the WordNet lexical knowledge using information derived from Wikipedia to improve performance of the measurements (Ponzetto and Strube, 2007). The WordNet similarity metrics have been used for a wide range of language processing applications including text summarisation systems (Carenini et al., 2008), and for determining textual entailment (Zanzotto and Moschitti, 2006).

Word Sense Disambiguation

Word sense disambiguation (WSD) is the task of identifying which one of the senses of a word is used in a particular context, when the word has multiple meanings (i.e. is *polysemous*). This is an open problem in natural language processing, and it is considered AI-complete (Navigli, 2009) (i.e. it is at least as hard as the most difficult problems in AI). WSD is one of the most straightforward language processing applications of WordNet since WordNet itself comprises a sense inventory which can be readily used for this purpose. Currently there are two main kinds of approaches to the problem[2]. The first are *supervised* approaches, which required some hand-labelled data, in which the senses of the words have been manually identified. Supervised approaches then attempt to learn from this hand-labelled data how to identify the correct senses, using various features in the context of the ambiguous words. Supervised systems have often achieved the best results on the commonly used evaluation sets such as the Senseval or Semeval tasks (Pradhan et al., 2007). However these require large amounts of hand-tagged data which is expensive to create. Currently there is only a small amount of training data available, with

[2] Another kind of approach is *unsupervised* - however pure unsupervised approaches make no use of sense inventories or dictionaries, and aim instead to identify sense clusters rather than identify sense labels (Navigli, 2009).

SemCor[3] being a commonly used corpus.

The second set of approaches are *knowledge-based*. These approaches use the information in a lexical knowledge base (such as WordNet) for disambiguation without using labelled training data. One knowledge-based approach is to use overlap of sense definitions to obtain the best sense. This approach is named *gloss overlap* or the *Lesk* algorithm after its author. This approach requires computing the pairwise overlap of all word senses within the context - which gives rise to an exponential number of steps relative to the number of context words and senses. A variant of this approach addresses this problem by only finding the overlap between each word sense and the context words themselves. Given a target word w the following score is computed for each sense S of w:

$$score_{Lesk}(S) = |context(w) \cap gloss(S)| \tag{2.10}$$

where $context(w)$ is the bag of all content words in a context window around the target word w.

The other main type of knowledge-based approaches are structural approaches: these use structural information from computational lexicons such as WordNet. Some of these approaches use similarity measures to find the best sense for a particular word. Given a scoring function to evaluate the similarity of two word senses:

$$score : Senses_D \times Senses_D \rightarrow [0, 1] \tag{2.11}$$

a target word w_i in a text $T = (w_1, \ldots, w_n)$ is disambiguated by choosing the sense S of w_i which maximizes the following sum:

[3] http://www.cs.unt.edu/∼rada/downloads.html

$$S = \operatorname*{argmax}_{S \in Senses_D(w_i)} \sum_{w_j \in T : w_j \neq w_i} \max_{S' \in Senses_D(w_j)} score(S, S') \tag{2.12}$$

Given a sense S of the target word w_i, the formula sums the contribution of the most appropriate sense of each context word $w_j \neq w_i$. The sense with the highest sum is chosen. Many different scoring functions have been used for the disambiguation, including all the WordNet similarity metrics described previously (Patwardhan and Pedersen, 2006). However a major drawback with this approach is that the number of computations grows exponentially with the number of words to disambiguate as every pair of words must be checked.

More recently there has been a surge of interest in graph-based methods. These have the advantage of being able to find globally optimal solutions much more efficiently than the pairwise methods and these have been shown to outperform the state of the art supervised approaches. Sinha and Mihalcea (2007) uses a combinations of semantic similarity and graph-based measures. Graphs are constructed by using a window of a few words before and after the word to be disambiguated. All the senses of each word are listed. The weighting of edges between the word nodes are then computed using WordNet-based similarity measures, *lesk*, *jcn* and *lch* as described above. The graph centrality measures used are then as follows:

- **indegree** of a vertex in an undirected weighted graph G = (V,E) is defined as the sum of the weighted edge scores coming into that node:

  $$Indegree(V_a) = \sum_{(V_a, V_b) \in E} w_{ab} \tag{2.13}$$

  where w_{ab} is the weight on the edge between V_a and V_b.

- **closeness** of a vertex is defined as the reciprocal of the sum of the shortest paths between the vertex and all other vertices in the graph:

$$Closeness(V_a) = \frac{1}{\sum_{V_b \in V} s(V_a, V_b)} \tag{2.14}$$

where $s(V_a, V_b)$ is used to denote the "shortest path" or "shortest geodesic distance" between the nodes V_a and V_b.

- **betweenness** of a node is defined in terms of how "inbetween" a vertex is among the other vertices in the graph. Formally:

$$Betweenness(V_a) = \sum_{V_b \in V, V_c \in V} \frac{\delta_{V_b,V_c}(V_a)}{\delta_{V_b,V_c}} \tag{2.15}$$

where δ_{V_b,V_c} is the total number of shortest geodesic paths between V_b and V_c while $\delta_{V_b,V_c}(V_a)$ is the number of such paths that pass through V_a.

- **PageRank** (Page et al., 1999) uses the idea that a link from one vertex to another is casting a vote or recommendation for that vertex. The PageRank score is defined as :

$$PageRank(V_a) = (1 - d) + d \times \sum_{(V_a,V_b) \in E} \frac{PageRank(V_b)}{degree(V_b)} \tag{2.16}$$

where $degree(V)$ is the number of outlinks from V, and E is the set of edges.

The weighting of the graph edge are also taken into account in Sinha and Mihalcea (2007):

$$PageRank(V_a) = (1 - d) + d \times \sum_{(V_a,V_b) \in E} \frac{w_{ba}}{\sum (V_c, V_b) \in E w_{bc}} PageRank(V_b) \tag{2.17}$$

The best results are used by combining all three semantic similarity metrics and using a voting scheme combination of the graph based measures, achieving 57.6% F-measure on the SENSEVAL 2 test set.

Navigli and Lapata (2007) use a different approach to graph-based WSD using WordNet. Given a sentence to disambiguate, a graph is induced from WordNet by using the word senses in the sentence as nodes, and relations as the edges. The graph is extended using a depth-first search through the WordNet relations, with a limit of 6 edges for the path length. Both local and global graph-based metrics are then used to find the best sense for each word. The local measures compute independently the degree of relevance of a single vertex v in a graph G. The global connectivity measures are concerned with the structure and properties of the graph as a whole. However the best perfoming metric was found to be a local one, the KPP (Key Player problem), which finds vertices which are relatively close to other neighbours:

$$poKPP(v) = \frac{\sum_{u \in V : u \neq v} \frac{1}{d(u,v)}}{|V| - 1} \tag{2.18}$$

where the $d(u, v)$ is the length of the shortest path between u and v. This achieves a F-1 score of 40.5% on the Senseval 3 test set.

Agirre and Soroa (2009) describe an approach which adapts PageRank for the task of word sense disambiguation, giving a new algorithm dubbed 'Personalized Page Rank' or *ppr*. The graph used is derived from the relations in WordNet plus links derived from gloss disambiguations for each synset. Then for each sentence to be disambiguated the context words of the sentence are inserted into the graph as nodes, and linked to the respective concepts in WordNet. The initial probability mass is then concentrated over the newly introduced word nodes. This has the effect of influencing the PageRank metric so that the PageRank value for each of the nodes in the LKB is effectively a measure of the structural relevance of that concept in the presence of the input context. A further refinement is motivated by the problem of related different senses of a word reinforcing each other, thus dampening the effect of other senses of that word. This is addressed by building the graph for each target word in the context: for each target word W_i, the initial probability mass is concentrated over the senses of the words surrounding W_i, but not in the senses of the target word itself. The aim is to let the surrounding words decide which concept associated to W_i has more relevance. This refined approach is dubbed *ppr_w2w* and does not disambiguate all context words in a single run, which makes it much less efficient that *ppr*. This achieves performance of 57.4% on the Senseval 3 test set.

Information retrieval

Many IR systems retrieve only documents that contain the words in the query, but not those containing words which are similar or related in meaning. So for example if the user enters a query containing the word 'car' then documents containing the word 'automobile' will not be matched, despite the strong semantic similarity between the two terms.

There are different ways in which WordNet can be used to address this problem. Query expansion using related terms in WordNet was used in Moldovan and Mihalcea (2000). More recently, Hliaoutakis et al. (2006) presents a comprehensive solution using reweighting of query terms according to semantic similarity and construction of a similarity matrix based on WordNet similarity metrics. Fang (2008) experiment with different semantic similarity metrics to expand information retrieval queries. The most effective similarity metric for this purpose is found to be a gloss overlap metric which calculates the overlap between the glosses of two synsets.

Another interesting application is cross-lingual information retrieval. Work has been carried out using EuroWordNet (Vossen, 1998) for this purpose. Result showed that WSD is useful for CLIR using EuroWordNet (Clough and Stevenson, 2004).

More recently Peters et al. (2006) has developed a multi-lingual legal WordNet which is then used to allow cross-lingual queries.

2.1.5 Alignment with other lexical resources

Work has been done on aligning WordNet with other lexical resources. This includes alignment with Roget's thesaurus and the LDOCE (Kwong, 1998). This was done by finding the overlap between sense definitions in LDOCE and the WordNet synset, hypernyms and gloss words. For ROGET, the overlap is computed using the synset, hypernyms and co-ordinate terms. This was tested on a small sample of 36 words, divided equally into 3 groups based on the polysemy in WordNet: low ($1-5$ senses), medium ($6-10$), and high (> 11). Results vary from 64.8% accuracy for the LDOCE to WordNet mapping, to 78.9% for the WordNet to Roget mapping for the low polysemy words. For the highly polysemous words accuracy drops to 53.0% for LDOCE to WordNet and 69.8% for the Roget to WordNet mapping.

For the first Senseval WSD competition (Kilgarriff, 1998), the Hector corpus was used as the sense inventory, in addition to the WordNet sense-annotated Semcor corpus. The HECTOR database consists of a tree of senses, containing definitions, syntactic properties, example usages and "clues" (collocational information about the syntactic and semantic environment in which a word appears in a specific sense). To adapt their systems for the competition therefore, it was necessary for some participants to create a mapping between WordNet and Hector senses. Litkowski (1999) test two different approaches to achieve this mapping. First is word overlap, which achieved accuracy of 36.1% on a test set of 86 cases. The second uses 'componential analysis' which involves parsing definitions and using patterns to identify semantic relations present in the definitions. To improve performance, values in the relations are relaxed to allow synonymic substitution (using WordNet). Using this approach achieves 40.7% accuracy on the same test set.

Mapping to a domain-specific terminology database was tested by Burgun and Bodenreider (2001), which tested the mapping between WordNet and the Unified Medical Language System (UMLS). The mapping was done both for terms (corresponding to individual words within a synset), and concepts (corresponding to whole synsets). The mapping from WordNet to UMLS was done using the Knowledge Source Server (McCray et al., 1996). The mapping from UMLS terms was done using the standard *wn* interface to WordNet. Terms were considered equivalent if they mapped successfully using these methods. Concepts were determined to be equivalent if at least one term of the WordNet synset was equivalent to at least one term from the UMLS concept. Two semantic classes were used to compare WordNet and the UMLS :ANIMAL, a general class, and HEALTH DISORDER, typical of the medical domain. For the WordNet to UMLS mapping, for the ANIMAL class 51% of the 3984 synsets and 36% of the 7961 terms were mapped succesfully. For the HEALTH DISORDER class, 83% of the 1379 synsets and 77% of the 2194 terms were mapped successfully. Therefore the overlap is higher between WordNet and UMLs for concepts than for terms. For the UMLS to WordNet mapping, for the ANIMAL class 19% of the UMLS concepts were found in WordNet. For the HEALTH DISORDER class, 2% of more than 140,000 concepts were found in WordNet. It was concluded that terms represented in WordNet are sometimes absent from medical vocabularies. For example, a synonym of "infectious mononucleosis" in WordNet is "kissing disease", which does not exist in the UMLS. This kind of lay terminology may be of interest for some applications in consumer health projects for example.

2.1.6 Related projects

Possibly the largest single project directly related to WordNet is EuroWordNet (Vossen, 1998), a large multi-lingual lexical database based on the original WordNet. This consists effectively of separate wordnets for each of the individual languages

(Dutch, Spanish, Italian, and more), which broadly follow the same kind of structure as the original WordNet. These are then linked together by equivalence relations via a central set of concepts named the Inter Lingual Index, which is based on the original WordNet. The resulting resource has been used for various language processing research applications including cross lingual information retrieval.

Another related project is WordNet Domains, an effort to annotate WordNet synsets with subject field codes, which describe the broad subject area to which the synset belongs. So for example the MEDICINE label groups together nouns such as doctor and hospital, together with verbs such as operate. The subject field codes are based on the Dewey Decimal Classification. The annotation is performed manually for a small number of high level synsets. An automatic procedure then exploits the WordNet relations (hyponymy, meronymy, etc.) to extend assignments to all reachable synsets. The work has mainly focussed on the noun hierarchy to date, with 96% of noun synsets having been annotated.

2.2 Enriching WordNet

The standard method for new words or relations to be added to WordNet is by lexicographers looking through concordance lists for words and manually adding new relations to the database. We would expect new relations added in this way to be accurate. However the main disadvantage is that this method is very laborious and time consuming (Church and Hanks, 1990). Therefore there is a strong incentive to find automatic methods to find new words and relations in WordNet. This section organizes previous work on this task by the type of method used to find novel entities and relations:

- Lexical co-occurrence looks at keywords in the context of novel words to help identify the location in which to insert these words into the taxonomy.

- Lexico-syntactic patterns (either manually or automatically created) can be used to identify patterns indicating possible relations in text. This goes beyond lexical co-occurrence by taking into account syntax.

- Gloss disambiguation. This exploits the existing manually written glosses in each WordNet synset. By disambiguating the glosses this enriches WordNet with many new possible relationships.

2.2.1 Lexical co-occurrence

The concept that the sense of a word depends crucially on the surrounding context was famously drawn to attention by Firth (1957) who said:

> You shall know a word by the company it keeps.

Agirre et al. (2001) uses this idea to enrich WordNet with topic signatures (lists of topically related words) for each synset. These are found by searching the Web, and selecting the most relevant words using a χ^2 distribution. In Agirre et al. (2001) the topic signatures are used to cluster similar word senses together to address the sense proliferation problem of WordNet. However the topic signatures could be used to enrich WordNet with new relations. This idea was used in Widdows (2003) where a large corpus was used to find semantic neighbours of an unknown word using latent semantic analysis. This captures the co-occurence of frequently occuring meaningful words in a large matrix. The word is then attached into the taxonomy by finding the node where the semantic neighours are most concentrated.

A similar method is used by Pantel (2005) which introduces a framework for inserting *co-occurence vectors* into an ontology such as WordNet. This essentially derives a list of significant related words for each synset. This was then used to add new nodes into the ontology by comparing the feature vectors with each of the possible attachment points. Accuracy of 73.9% is achieved at finding the correct attachment point for unknown words.

2.2.2 Lexico-syntactic patterns

The idea of finding new WordNet relations by searching for lexico-syntactic patterns in large corpora was originally proposed by Hearst (1992). The method can be illustrated with the following example from Grolier's *Academic American Encyclopedia*:

> Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use.

From this sentence it can be inferred that Gelidium is a kind of red algae. The semantics of the lexicosyntactic pattern "NP_0 *such as* NP_1" implies that NP_1 is a hyponym of NP_0.

Another example pattern is illustrated by the following text:

> Bruises, wounds, broken bones or other injuries ...

Here the pattern "NP_1, NP_2, NP_3... *or other* NP_0" implies the following:

- NP_1 is a hyponym of NP_0
- NP_2 is a hyponym of NP_0
- NP_3 is a hyponym of NP_0
- etc.

Altogether 10 such patterns are defined in Hearst (1992). These patterns have been widely used and extended since their proposal and are often referred to as *Hearst patterns*. Work by Cimiano et al. (2004) used the Hearst patterns and a few other manually created patterns to categorize proper nouns with the correct concepts to extend existing ontologies. Given a candidate proper noun their system instantiated patterns using this noun with each concept from the ontology to generate hypothesis phrases which were then used as search phrases into the Google$^{\mathrm{TM}}$ search engine. Counting the results then allowed the system to find the most likely concept with which to categorize the noun.

The work described above created the patterns manually by inspecting corpora and finding patterns which seemed to be indicative of some relationship. An obvious question is whether such patterns could instead be derived automatically. This was also investigated in Hearst (1992) where a standard pattern discovery procedure was outlined:

1. Choose the relation of interest (hypernymy or meronymy etc.)

2. Find existing word pairs where this relation holds using entities in WordNet (e.g. *car* is-a *automobile*.

3. Find sentences from the corpus which contain these word pairs and record the lexical and syntactic context.

4. Find the commonalities among these contexts and use these to derive the patterns.

Using this approach several new productive patterns were discovered. The whole set of patterns generated 152 new relations in total. Evaluating against those terms that already appeared in WordNet it was found that 61 out of 106 possible relations were discovered by the patterns.

This approach was extended in Snow et al. (2005). This used a similar approach to that of Hearst (1992). Dependency paths were used as a general purpose representation of the lexico-syntactic patterns. It was shown that this space includes the hand crafted Hearst patterns, with each of the patterns having a corresponding dependency path formalization. Evaluation showed the best results for hypernym classification using the Wikipedia corpus as training data. Evaluations on a manually annotated test set showed an improvement in the F-score over using WordNet data alone.

2.2.3 Gloss disambiguation

Work by Harabagiu et al. (1999) recognized that the glosses for each synset provide useful information and potential extra links and relationships for each synset. To fully exploit this information the problem then becomes that of disambiguating the words in the glosses to identify the sense of each word used. In Harabagiu et al. (1999) this is done using lexical and semantic heuristics and statistical methods. The resulting disambiguated glosses then allow extra links to be added between synsets where they are used in each other's glosses.

This work has recently been superseded by efforts to manually annotate the glosses in WordNet with the sense tags and this work is now included with the latest release, WordNet 3.

2.3 Wikipedia

Wikipedia is a freely accessible online encyclopedia. Any internet user can create or edit a page on Wikipedia (hence the *wiki* in the name). This approach has allowed rapid expansion of the encyclopedia, going from around 1,000 articles in the weeks after its creation in February 2001 to currently over 3 million English articles. However the approach naturally attracts questions over the quality of the articles. The main safeguard for quality assurance is the collaborative nature of the wiki; if an error is found within a page anyone else can correct it. Deliberate vandalism can be reported to an adminstrator and users can be blocked. Despite news stories over the last few years highlighting particular problems over controversial articles and personal attacks, the wiki approach has worked (perhaps surprisingly) well. An expert comparison of Wikipedia against the more established Encyclopedia Britannica on a sample of science articles found that over 42 articles there were only 8 serious errors found, 4 from Wikipedia and 4 from Brittanica (Giles, 2005). Many more minor errors or omissions were found, 162 from Wikipedia compared to 123 in Brittanica (a ratio of approximately 4:3). The key advantage of Wikipedia is that errors could in theory be much more quickly corrected than for a more traditional volume like Brittanica.

The main content of Wikipedia consists of *articles* or *pages.* These are hypertext documents which can link to other articles both within and outside Wikipedia. Articles are uniquely identified by the title. Where possible ambiguity could occur titles contain an explanation in parentheses, for example *Kent (band)* refers to a Swedish rock band while *Kent* refers to the county in England. As in this example, the parentheses are usually reserved for the more obscure concepts although this is of course a subjective judgement.

Also of interest are *redirect* pages, which have no content themselves, but point to other articles. These are used where many different names can refer to the same concept. For example *Cambridge University* and *Cambridge Uni* both point to the article *University of Cambridge.*

Another important type of page in Wikipedia is the disambiguation page which are created for ambiguous names and consist of lists of links to articles defining the different meanings for the name. These pages are sometimes defined by the word disambiguation in the title, as in *family (disambiguation).* In other cases there are tags within the document indicating that the page is a disambiguation page.

Additionally Wikipedia pages can belong to one or more *categories*, which are decided on by contributors and editors. The types of categories can vary considerably. Some categories, the *conceptual* categories, do indeed identify the class for the entity of the page (e.g. Zidane is in the category *French football players*). Others serve administrative purposes (e.g. Zidane is in the category *Articles with unsourced statements*), while others indicate some other relation (*1972 births*) or thematic vicinity (*Football*).

2.3.1 Creating a taxonomy from Wikipedia categories

Ponzetto and Strube (2007) uses the Wikipedia category system to derive a taxonomy. This is done by using connectivity in the category network, and also using lexico-syntactic methods. The approach starts by taking the full categorisation network consisting of approximately 166,000 nodes and 349,000 links between them.

The first step is to filter out category nodes which are for administration or management purposes. This leaves 127,000 nodes and 267,000 links. The second step then identifies two common patterns Y X and X BY Z e.g. (MILES DAVIS ALBUMS and ALBUMS BY ARTIST). For all categories containing BY in the name, all subcategories links are labelled with an *is-refined-by* relation. This labels 55,000 category links, leaving 213000 unlabelled. The third step is then to apply syntax-based methods. Category labels are parsed using the Stanford parser (Klein and Manning, 2003). Two methods are then used to find *isa* relations. The first is head matching - to label pairs of categories sharing the same lexical head e.g. BRITISH COMPUTER SCIENTISTS *isa* COMPUTER SCIENTISTS. The second method is modifier matching - to label as *notisa* if the lexical head occurs in the non-head position in the other category. This is to rule out thematic categorisation links such as CRIME COMICS and CRIME or ISLAMIC MYSTICISM and ISLAM. A total of approximately 73,000 *isa* relations are found by head matching and 38000 *notisa* relations are found by modifier matching. The fourth step uses the structure and connectivity of the categorisation network. Two methods are used. The first is instance categorisation, using the heuristic of Suchanek et al. (2007) that if the head of the page category is plural, then the *isa* relation can be applied i.e. ALBERT EINSTEIN belongs to the NATURALIZED CITIZENS OF THE UNITED STATES category. In Ponzetto and Strube (2007) this is further extended to find *isa* relations between categories as well. So for the page MICROSOFT being categorised as COMPANIES LISTED ON NASDAQ, evidence is derived that Microsoft is a company and specifically MICROSOFT *isa* COMPUTER AND VIDEO GAME COMPANIES. The second method is redundant categorisation. This uses the idea that if a page falls into two categories, then one subsumes the other. So for example ETHYL CARBAMATE is both an AMIDE and and ORGANIC COMPOUND - implying by transitivity that one category is subsumed by another - in this case AMIDE is a ORGANIC COMPOUND. Using the instance categorisation and redundant categorisation methods finds 10000 and 11000 *isa* relations respectively.

After applying steps 1-4, there are still 82,000 unclassified relations. The next step is to apply lexico-syntactic patterns to identify *isa* relations as in Hearst (1992). Patterns are also used to improve precision by identifying *notisa* relations. These methods find approximately 15000 *isa* relations and filter out 3000 previously idenitified positive links. The last set of methods propagate the previously found relations by means of multiple inheritance and transitivity. The resulting taxonomy is evaluated by comparing with ResearchCyc (Guha et al., 1990) - achieving a recall of 89.1% and precision of 86.6%.

2.3.2 Linking Wikipedia categories to WordNet synsets

Similarly Suchanek et al. (2008) uses heuristic methods to link Wikipedia categories to synsets in the WordNet hierarchy, thus creating a new ontology which is named YAGO (Yet Another Great Ontology). The structure of the ontology is a slight extension of RDFS which forms the basis of OWL. All objects (e.g. cities, people) are represented as *entities* in YAGO. Two entities can stand in a *relation*. So for example the fact that Albert Einstein won the Nobel Prize can be stated by saying the entity `Albert Einstein` is in the HASWONPRIZE relation with the entity `Nobel Prize`:

`AlbertEinstein` HASWONPRIZE `NobelPrize`

Words are also regarded as entities. This makes it possible to express that a certain word refers to a certain entity. This is done using the MEANS relation. For example:

'Einstein' MEANS `AlbertEinstein`

This allows for ambiguity as well, so the following line says that 'Einstein' may also refer to Alfred Einstein the musicologist.

'Einstein' MEANS `AlfredEinstein`

Similar entities are grouped into classes. For example the class `physicist` comprises all physicists, and the class `word` comprises all words. Each entity is an instance of at least one class. This is expressed by the TYPE relation:

`AlbertEinstein` TYPE `physicist`

Classes are themselves entities, instances of the class `class`. Classes are arranged in a taxonomic hierarchy, expressed by the `subClassOf` relation.

`physicist` SUBCLASSOF `scientist`

In YAGO, relations are entities as well. This allows properties of relations to be expressed within the model. For example to express that the SUBCLASSOF relation is transitive by making it an instance of the class `transitiveRelation`:

`subClassOf` TYPE `transitiveRelation`

Relation triples are referred to as facts. Each fact is given a *fact identifier*. Fact identifiers are entities as well in YAGO allowing us to store information about the fact. For example suppose the fact (`AlbertEinstein`, BORNINYEAR, `1879`) had the fact identifier `#1` then the following line would say this fact was found in Wikipedia:

`#1` FOUNDIN `http://www.wikipedia.org/Einstein`

Creating an ontology

The system extracts a YAGO ontology from WordNet and Wikipedia. All facts are tagged with a confidence value between 0 and 1. Currently they are tagged using the empirical confidence estimation value which lie between 0.90 and 0.98. Since Wikipedia has many more articles than WordNet synsets, the candidate individuals for YAGO are taken from Wikipedia. So for example the article about Albert Einstein is a candidate to become the individual `AlbertEinstein` in YAGO. The page titles in Wikipedia are unique.

The classes for each individual are established using the category system in Wikipedia. As mentioned earlier, some categories such as the *conceptual* categories, do indeed identify the class for the entity of the page (e.g. Albert Einstein is in the category *Naturalized citizens of the United States*). Others are irrelevant since they exist only for administrative purposes (e.g. Albert Einstein is in the category *Articles with unsourced statements*), while others indicate some other relation (*1879 births*) or thematic vicinity (*Physics*). To filter out unwanted categories and keep only the conceptual categories the system runs a shallow linguistic parser over the category names. Heuristically it was found that if the head word of the category name is a plural then it is likely to be a conceptual category. So for example the head word of *Naturalized citizens of the United States* is *citizens*. A stemmer was used to identify plurals. An additional benefit of applying this method is that articles that do not describe individuals (for example hub pages) do not have conceptual categories. This means that the conceptual categories yield as its domain the set of individuals and as its range the set of classes.

Wikipedia categories are organized in a hierarchy. However this hierarchy reflects thematic structure rather than a taxonomy. So as mentioned earlier Zidane is in the category *Football in France*. Hence only the leaf categories of Wikipedia are used as classes in YAGO. Instead WordNet is used to establish the taxonomy of classes. Each synset of WordNet becomes a class of YAGO. Proper nouns from WordNet (which would be individuals) are excluded from YAGO, to avoid duplication of entities. So for example although Albert Einstein is an synset in WordNet this is not included in YAGO. There are about 15,000 cases where an individual is known to both WordNet and Wikipedia. In some of these cases the Wikipedia page describes an individual that has a common noun as its name. For example 'Time exposure' is a common noun for WordNet but an album title in Wikipedia. In the overwhelming majority of cases however the Wikipedia page is about the common noun (the Wikipedia page 'Physicists' is about physicists). To be on the safe side preference is given to WordNet and the Wikipedia individual is discarded in case of conflict. This means information about individuals that have a commmon noun as a name are lost, but it ensures that all common nouns are classes and that no entity is duplicated.

The subClassOf hierarchy of classes is taken from the hyponymy relation in WordNet: a class is a subclass of another one if the first synset is a hyponym of the second. The lower classes extracted from Wikipedia have to be connected to the higher classes extracted from WordNet. For example the Wikipedia class *American people in Japan* has to be made a subclass of the WordNet class *person*. To achieve this the category name is parsed, and the head compound, pre-modifier and post-modifier of the name is found. So for example the Wikipedia category

American people in Japan has head-compound ‘people’, pre-modifier ‘American’ and post-modifier ‘Japan’. The head compound is stemmed to its singular form (i.e. ‘person’ in the example). The algorithm first checks if there is a WordNet synset with the name *pre + head*, i.e. *American person* from the example. If there is then the Wikipedia class becomes a sub class of this WordNet class. If not, then the head compound (*person*) needs to be mapped to the appropriate synset. It was found that mapping to the most frequently occuring synset of the word had the best results (i.e. the most frequent synset containing ‘person’). A dozen prominent exceptions were manually corrected, e.g. *capital* in Wikipedia means capital city, but in WordNet the most frequent sense is *financial asset*.

WordNet synsets contain synonymous words within each synset. For example the synset *city* contains ‘urban center’ and ‘metropolis’. In YAGO a new means relation is added for each word in each synset e.g. (‘metropolis’, means, *city*). Wikipedia has redirect pages which serve to redirect users to the correct page. So for example, ‘Einstein, Albert’ redirects to the page for ‘Albert Einstein’. A new means relation is added for each redirect e.g. (‘Einstein, Albert’, means, `Albert Einstein`). If the words referring to individuals uses the pattern of given name, following name, the YAGO system deduces they refer to people. The relations givenNameOf and familyNameOf are established. These are subrelations of the means relation.

Other relations include bornInYear, diedInYear, establishedIn, writtenInYear etc. These are all derived from processing the category name in some way. Although a huge number of facts are extracted, the process is very fast because only the category names are examined and not the pages themselves. Meta-relations are also stored, including the links to other pages within the article page (context), and the URL where facts were found (describes).

Evaluating the ontology

The ontology was evaluated manually. The portions of YAGO obtained directly from WordNet were excluded since human accuracy could be assumed for these cases. Likewise, non-heuristic relations such as DESCRIBES, MEANS and CONTEXT were also excluded. The evaluation thus concentrates on the potential weak points of the ontology. The evaluation showed very good results betwen the range of 90.8% and 98.7% for all relations evaluated. The crucial TYPE relation and the link between WordNet and Wikipedia SUBCLASSOF turned out to be very accurate, achieving accuracy of 94.5 and 97.7% respectively. Some errors were introduced by erroneous Wikipedia categories (for example an article about a person born in 1802 in the category *1805 births*), and vagueness or ambiguity (is an economist who works in France a French economist even if he was born in Ireland). To give some indication of the size of YAGO there were 143,000 SUBCLASSOF facts and 1.9 million TYPE facts. Altogether there were 5 million ontological facts. There were 907,000 individuals (not including words) and 149,000 classes. YAGO is far larger than other publicly available ontologies: WordNet has 207,000 facts, and OpenCyc has 306,000.

Similar work

Ponzetto and Navigli (2009) also links categories to synsets using a graph based approach. Once this mapping is done it is used to restructure the Wikipedia category taxonomy. This allows many Wikipedia instances to be added into the WordNet hierarchy.

2.3.3 Mapping articles to synsets

Using text similarity metrics

Ruiz-Casado et al. (2005) use text similarity to link articles to synsets. This was done using the Simple English Wikipedia, a much smaller resource than the full Wikipedia. For each article in Wikipedia, the approach attempts to find the best matching synset. The first step is to find all synsets which contain the title of the Wikipedia article. A variable N is set to 1. Each synset is then represented by the set of words in its gloss definition, the words in the synset, and hypernyms to level N. Terms are weighted in comparison with the glosses for the other senses. Two weighting functions were tested: $tf{-}idf$ and χ^2. The sense with the highest similarity is chosen; if there is a tie between two or more senses, then N is incremented and the process repeated. Use of the dot product with stemming and $tf - idf$ weighting was found to be most effective. This mapping was then used in Ruiz-Casado et al. (2007) to learn lexical patterns with which to extract new relations to add to WordNet. So for example if 'Lisbon' and 'Portugal' were both mapped to WordNet synsets, and the 'Lisbon' article contained the sentence 'Lisbon is part of Portugal' the 'is part of' would be used to identify new meronym relationships not already present in WordNet.

Recent work by Ponzetto and Navigli (2010) maps Wikipedia articles to WordNet synsets. This is done by creating a context for the article and the synset. This is a set of words intended to represent the item. For the article this comprises the following:

- Sense labels. The words in the parentheses after the title. So for the article SODA (SOFT DRINK), the words SOFT and DRINK are added to the context.
- Links. The titles of the pages linked from the article. This would include SODA,

LEMONADE, SUGAR etc.

- Categories. For example SODA is categorised as SOFT DRINKS. Since categories can often be very specific, only the syntactic head is used in the context. So for the categories SWEDISH WRITERS or SCIENTISTS WHO COMMITTED SUICIDE, only WRITER and SCIENTIST respectively are used in the context.

For a particular word sense in WordNet, the following information is used in the context:

- Synonym words in the synset. So for the word soda all synonyms are included: tonic, soda pop, pop.

- All words in the hypernym or hyponym synsets of the word sense. So for the word soda the words from the hypernym soft drink are included.

- Words from sister synsets are included. Sister synsets are those that share a direct hypernym, i.e. bitter lemon and soda are sisters. The words bitter and lemon are added to the context.

- Content words from the gloss are added to the context. For instance the gloss of soda is “a sweet drink containing carbonated water and flavoring”. Thus the words sweet, drink, contain, carbonated, water and flavoring are added to the context.

The mapping algorithm then selects the word sense for the article whose context has the greatest overlap with the article context. For example for the SODA article, there are two candidate word senses, the sodium carbonate and drink senses. The context for the drink sense has the greatest overlap with the article context and therefore is chosen as the word sense to map the article to. The mapped articles are then used to add new links to WordNet. Where a link exists between two mapped articles in Wikipedia a new relation is added in WordNet. This enriches WordNet with many new links. This enriched WordNet is then used with an extended Lesk and a graph-based degree centrality approach for coarse-grained WSD on the Semeval 2007 task, and is found to give better results than using either WordNet or Wikipedia alone.

Other approaches

Medelyan and Milne (2008) use a similar approach to link Wikipedia articles to the domain-specific agricultural ontology Agrovoc, which again allowed additional synonyms and topical relations to be added. Reiter et al. (2008) links articles in Wikipedia to a domain-specific music ontology using keywords to choose amongst ambiguous articles to match to each class in the ontology.

Medelyan and Legg (2008) align CYC entities with Wikipedia articles by matching the titles of the Wikipedia articles against the entities, and also by using the surrounding context of the entities - hypernyms and hyponyms in CYC, and linked articles in Wikipedia. A suggested application of this mapping is to enrich CYC with additional synonyms for entities, exploiting the redirect system in Wikipedia.

Bunescu and Pasca (2006) detect and disambiguate named entities in text against Wikipedia articles. This recognizes that certain named entities are ambiguous, for example ‘John Williams’ refers to a wrestler, a composer and a winner of the Victoria Cross. A kernel similarity function is used to organise the named entities into a dictionary. This would allow web search queries to return results which were organised by the named entities, allowing the user to select the relevant one.

Mihalcea (2007) creates a sense tagged corpus of ambiguous words in Wikipedia. This is done by first extracting all paragraphs containing the ambiguous words. Then all possible labels for each word are collected using the words in the links. For example these paragraphs from Wikipedia link to two different senses of the word 'bar':

> In 1834, Sumner was admitted to the [bar(law)|bar] at the age of twenty-three, and entered private practice in Boston.
>
> It is danced in 3/4 time (like most waltzes), with the couple turning approx. 180 degrees every [bar (music)|bar].

The different senses of the word 'bar' are then manually mapped onto WordNet senses. A word sense disambiguation system is then trained using this corpus.

For further information Medelyan et al. (2009) give a comprehensive summary of ways in which machine-readable knowledge has been gleaned from Wikipedia.

2.4 Summary

WordNet is a lexical knowledge base, containing information about synonyms, hypernyms and meronyms. WordNet is richer in content than other machine-processable dictionaries and thesauri, is easier to use for external applications, and is freely available. This has resulted in WordNet being widely used in language processing applications.

Wikipedia is an openly accessible online encyclopedia of categorised and hyperlinked articles. Although far smaller than the Web as a whole, the open, collaborative style of editing means mistakes are usually corrected quickly, resulting in a high overall quality. The encyclopedic nature and the quality of Wikipedia has often proved more useful in language processing tasks than general news corpora, or large Web collections.

There has been a body of previous work on enriching knowledge bases such as WordNet using automatic methods over natural language text. This has used various approaches such as lexico-syntactic patterns and lexical co-occurence. Several approaches have used information from Wikipedia to enrich WordNet with some success.

Chapter 3

Mapping WordNet to Wikipedia

This chapter describes the methods used to create a mapping between WordNet noun synsets and Wikipedia articles. Section 3.1 defines the problem in more detail, and gives an overview of the three stage approach used to generate the mappings. Section 3.2 gives more background information about synsets and articles as relevant for the task. The rest of the chapter then describes each of the mapping stages in more detail. The first stage is candidate article retrieval (Section 3.3) which aims to reduce the search space by identifying a small (but high recall) set of candidate articles for each noun synset using various methods to search Wikipedia. The second stage is the selection of the best mapping (Section 3.4) from the candidate article set (or deciding that there is no appropriate match) using text similarity methods. Finally the third stage is refinement of the mappings (Section 3.5) where a global approach making use of Wikipedia links eliminates spurious synset-article matches, thus selecting a more precise set of mappings. The chapter is summarised in Section 3.6.

3.1 Overview of approach

The set of 82115 noun synsets in WordNet is denoted S, and the set of 3 million+ Wikipedia articles as A. For each synset s in S the goal is to find the best matching article a in A or decide that no appropriate article exists. It is reasonable to question whether *more* than one article match might exist for a given synset. However this is very unlikely when the nature of WordNet and Wikipedia are taken into consideration. Some WordNet synsets are abstract or obscure and have *no* matching article (as we will discover in Section 4.1). However if a WordNet synset does match with a Wikipedia article then both the synset and the article are describing the same specific entity or concept (note that Wikipedia articles about specific instances are not considered as matches for general synsets - so for example articles about particular films are not considered good matches for the Film synset, but only the FILM article itself). Therefore it is very unlikely that *another* Wikipedia article exists that covers that same concept; if there was then the Wikipedia editors would quickly merge the two articles together and add a redirect page from one of the titles. Therefore it is safe to limit to at most one matching article for each synset.

To explain the mapping process, assume that there is some idealised similarity function that returns a value from 0 to 1 based on the semantic similarity of a given synset and article:

$$sim_{ideal}(s,a) \rightarrow [0,1] \tag{3.1}$$

where 1 represents a perfect match, and 0 completely unrelated. A further assumption is that there is a threshold t which separates good matches from others (i.e. if $sim_{ideal}(s,a) > t$ we have a good match otherwise we do not.)

Then ideally an implementation of the following function is required:

$$match_{ideal}(s) = \begin{cases} \underset{a}{\operatorname{argmax}}(sim_{ideal}(s,a)) & \text{if } sim_{ideal}(s,a) > t \\ null & \text{otherwise} \end{cases} \tag{3.2}$$

which returns the most similar article for each synset, or *null* if none of the articles exceed the similarity threshold.

Since there are 3 million articles in Wikipedia and 82115 noun synsets in WordNet computing the similarity for every synset-article pairing would be extremely computationally intensive. The entire Wikipedia text is over 14G in size which is too large to retain in memory and therefore database retrievals are required, which is relatively slow. This brute force approach is practically infeasible due to these memory and computation time requirements.

The approach used in this thesis is to reduce the number of articles considered for each synset using an efficient initial search method. This is **Stage 1** of the process, which uses title searching and information retrieval methods and is described in Section 3.3. The end result of this stage is a small set of candidate articles for each synset, ready to be processed in the further stages. Let *cand*(s) be the candidate article set for a synset s. The aim is that the candidate set contains the best matching article (if there is an appropriate article), as expressed here:

$$\forall s.\ match_{ideal}(s) \neq null \rightarrow match_{ideal}(s) \in cand(s) \tag{3.3}$$

The best performing methods are used to select candidate articles for each synset. The next stage is then to select the best match from this set of candidates. Now the brute force approach becomes feasible.

Once the candidate articles have been retrieved the task in **Stage 2** is to find the best article amongst the candidate set for each synset. This is done by using similarity functions which aim to approximate the idealised similarity function in (3.1). It is also necessary to determine a threshold value to distinguish good from bad matches. Once this is done it is straightforward to implement a mapping function to select the best match from the candidate articles as in (3.2). The similarity functions are described in Section 3.4. The best performing similarity functions are used to select the best mapping for each synset (or decide that no good matches exist).

Finally in **Stage 3** the aim is to find a set of more precise mappings from the whole set. This uses the global structure of the mappings and Wikipedia links to refine the mappings. This final stage is described in Section 3.5.

3.2 Noun synsets and articles

Chapter 2 gave a description of WordNet and Wikipedia. This section gives a more detailed account of the information present in a WordNet synset and of the methods available for searching Wikipedia to find articles. This sets a context for the methods described in the subsequent sections.

3.2.1 Synsets

Section 2.1.1 gave a description of WordNet. This section reviews the main sources of information within a synset in WordNet and how these might be used when mapping with Wikipedia articles.

The lemmas in the synset form the most important features of the synset when it comes to searching for relevant articles in Wikipedia since they capture the concept most distinctively. The gloss contains useful information which may help the searching; however it also contains noisy information, which may result in wrong matches. For example the gloss for the car synset contains the text "usually propelled by a combustion engine" which may result in the synset being matched up with an article about engines rather than the car.

The other main source of data we can derive from a synset are the *related* synsets. From the car synset we have hypernyms, or is-a related synsets, which are illustrated here:

> car → motor vehicle → self-propelled vehicle → wheeled vehicle → vehicle → conveyance → instrumentation → artifact → whole → object → physical entity → entity

A few of the hyponyms (inverse of hypernym) are shown here:

> ambulance, beach wagon, bus, cab, compact, convertible, coupe, cruiser, electric, gas guzzler . . .

Likewise some of the meronyms (part-of relation):

> accelerator, air bag, auto accessory, automobile engine, automobile horn, buffer, bumper, car door, car mirror . . .

There is very rich potential resource of data to be extracted from the related synsets. The data could be extended using glosses of related synsets. More distant relations could be used such as hyponyms of hypernyms (or sibling terms), extending to hundreds or thousands of synsets. However there will again be a tradeoff between useful information and noise; adding distant relations to the search queries seems likely to result in erroneous matches.

3.2.2 Articles

Section 2.3 gave a description of Wikipedia. This section gives more detailed information about Wikipedia articles, and how they are accessed through the usual

Wikipedia web interface. This helps to put in context the mapping methods described in the subsequent sections

Every article in Wikipedia is uniquely identified by its title. To help resolve ambiguities, parentheses distinguish between different meanings, for example Bar (establishment) and Bar (unit). It may be expected that the title would be the single most important feature with which to identify the best matching article for a given synset.

When end-users query Wikipedia using the standard Web interface, the titles are searched for matches. Given a query X, the following cases are possible:

1. If X unambiguously matches an article title, then that article will be returned.

2. If X *redirects* to an article Y (as described in Section 2.3) then the article Y will be returned with a note 'Redirected from X'. So for example Cambridge Uni redirects to the University of Cambridge article. The redirect system thus effectively captures possible synonyms for Y (including X).

3. If X is an ambiguous term then it redirects to a disambiguation page which lists the various possible meanings for X. (If none of these apply, the search will return possible spelling variations and a list similar to that of a search engine.) The disambiguation pages thus capture the polysemy of terms.

4. If none of these cases apply then the system will revert to a search-engine style output, which gives the most relevant articles using the query words as search terms. Otherwise it will simply state 'No articles found'.

The first three cases can be considered title matching methods, since the articles are searched on title alone. The last case (where no article exists with title X), then the system reverts to an IR approach, searching the whole text of the articles. Users might also arrive at articles through standard web search engines such as Google or Yahoo, which also use IR methods to retrieve the appropriate article.

3.3 Stage 1: Candidate article retrieval

The aim of this stage is to select a set of candidate articles which may be good matches for each synset. Two approaches are used. The first is **title search** where the article titles in Wikipedia are searched using the lemmas in WordNet. The second is **full text search** which forms queries from words in the synset and searches the full Wikipedia article text with an IR engine. These mirror the different end-user search methods described in the previous section (3.2). The IR method was found to give high recall, but the top result was not always the best (see Section 5.1.2); therefore it was decided to have subsequent stages to select the best from the top candidate articles.

In the following sections, the letters in parentheses provide shorthand reference points for each of the methods to link to the experimental results later in the thesis.

3.3.1 Title search

The title matching approach comprises different methods which search the Wikipedia database for articles with titles matching the words in the synset. The titles are searched using each of the lemmas in the synset. The hot dog synset is used as an example. This synset consists of the lemmas {frank, frankfurter, hotdog, hot dog, dog, wiener, wienerwurst, weenie}.

- Return all Wikipedia articles (A) where the title exactly matches one of the lemmas in the synset. Matching is case-insensitive. So for example the dog lemma will retrieve the article DOG about the domestic animal.

- Return articles which are redirected (R) from one of the lemmas. Here the word frankfurter redirects to Hot dog.
- Return articles linked to from the disambiguation (D) pages for one of those lemmas. For example the disambiguation page for dog contains links to many different articles containing the word dog, including:
  - Dog (film)
  - Hot dog (band)
  - Dog (domestic animal)
  - Dog (character from video game Half Life 2)
  - Dog (engineering tool)
  - etc..

The result of these searches are as follows:

- Exact title matches (A): {Hot dog, Dog}.
- Redirects (R): {Hot dog} (redirected from ‘frankfurter’ and ‘weenie’).
- Disambiguation links (D): {Dog (film), Hotdog (band) $\dots$}.

The advantage of these methods is that they can be executed with low computational cost, since the titles, redirects and disambiguation links can be indexed efficiently within a database. However a drawback with these methods is that they only consider the title and none of the other information within the article.

3.3.2 Full text search

The methods in the previous section used only the Wikipedia titles to find good matching articles for each synset. To make use of the article text, information retrieval (IR) can be used find articles using queries formed from the information in the synset. The advantage of this IR approach is that all the text within the article is considered in the search, not just the titles.

Experiments are performed using queries formed from combinations of different parts of information from the synsets. The following are features from the 'hot dog' synset:

- Lemmas (L) e.g. {frank, frankfurter, hotdog, hot dog, dog, wiener, wienerwurst, weenie}.
- Gloss (G) e.g. 'a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll'.
- Lemmas of related synsets (RL), such as hypernyms (hot dog is-a sandwich), hyponyms (chili dog is a hot dog), meronyms and holonyms (hot dog is part-of a hot dog (including bun)).

Wikipedia is then searched using queries formed by concatenating combinations of these features.[1]

An example query using just lemmas (L): 'frank frankfurter hotdog hot dog dog wiener wienerwurst weenie'. Example query using lemmas of related synsets (RL): 'sandwich chili dog hot dog'. Finally, an example query of lemma plus gloss (L+G): 'frank frankfurter hotdog hot dog dog wiener wienerwurst weenie a smooth-textured sausage of minced beef or pork usually smoked often served on a bread roll'.

3.3.3 Output

The end-result of running the title matching and IR approaches is a set of candidate articles for each synset, from which the subsequent methods find the best matching

[1] No additional processing of the queries, such as stopword removal is carried out since this is provided automatically by the IR system used for the implementation. See Section 5.1.2.

article. So for example the candidate article set for the 'hot dog' synset might include: {Hot dog, Dog, Vienna sausage, Hot wiener, Chicago-style hot dog, etc.}.

3.4 Stage 2: Selecting the best mapping

The next step is to try to find the best article match for each synset from the candidate article sets. This is done by assigning a similarity score to each article in the candidate article set based on the similarity with the synset. The most similar article is then chosen as the best match. Two methods were used. The first estimates similarity using the whole text of the article with information from the synset. The second uses just the title of the article.

3.4.1 Text similarity

This method works by calculating how many terms are shared between the synset and the article, and dividing by the number of terms in the smaller of the two (which will usually be the synset). The synset and article are each represented as a set of words. This is similar to the candidate article IR approach from Section 3.3, with different combinations of features included in the set:

- Lemmas (L) e.g. {frank, frankfurter, hotdog, hot dog, dog, wiener, wienerwurst, weenie}.
- Gloss (G) e.g. a frankfurter served hot on a bun.
- Lemmas of related synsets (RL), such as hypernyms (sandwich), hyponyms (chili dog), meronyms and holonyms (hot dog (including bun)).

The similarity is then computed using the overlap metric (Manning and Schütze, 1999) as:

$$text\_sim(A, B) = \frac{|A \cap B|}{min(|A|, |B|)} \tag{3.4}$$

where A represents the WordNet feature set, and B represents the Wikipedia feature set.

Considering an example synset A {hotdog, dog, frankfurter} and an article represented by the set B {hotdog, frankfurter, sausage, weiner} the similarity would be:

$$text\_sim(A, B) = \frac{2}{min(3, 4)} = \frac{2}{3} \tag{3.5}$$

since the sets share 2 terms in common and the synset is the smaller set with 3 items.

3.4.2 Title similarity

The previous method used the whole Wikipedia article for comparison. However the title of the article is the single most important feature when considering similarity with a synset. Therefore a further method assigns a similarity score using the title alone. For a synset $S = \{w_1, w_2, ...w_n\}$ the $title\_sim$ is computed as:

$$title\_sim(S, A) = \max_{w_i \in S} \begin{cases} 1 & \text{if } title{=}w_i \\ \frac{len(title)}{len(w_i)} & \text{if substr}(title,\ w_i) \\ \frac{len(w_i)}{len(title)} & \text{if substr}(w_i,\ title) \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

where $len(string)$ is the length of a string and $substr(a, b)$ is true iff a is a substring of b. This metric computes the substring overlap between the article title and the most similar word in the synset. The reason for this is that sometimes there is not an exact match between an article title and the lemmas. For example, using the synset: {frank, frankfurter, hotdog, hot dog, dog, wiener, wienerwurst, weenie}, and the (fictional) article HOT WIENERWURST, two lemmas qualify as substrings of the article title, wiener and wienerwurst. The word wiener has a score of len(wiener)/len(HOT WIENERWURST) = 6/15, whereas wienerwurst has a score of 11/15, and so the title similarity would be the maximum value, 11/15. Another example is the article DOG in which case there is an exact match with the lemma word dog so the title similarity would be 1.

3.4.3 Output

The similarity metrics described here are used to determine the best matching article for each synset. The metrics are instantiations of the idealised function described earlier (equation 3.1), and thus can be substituted into the idealised match function (equation 3.2). So for example using the text similarity function gives:

$$match_{text\_sim}(s) = \begin{cases} \underset{a}{\text{argmax}}(text\_sim(s,a)) & \text{if } text\_sim(s,a) > t \\ null & \text{otherwise} \end{cases} \tag{3.7}$$

In practice the similarity metric will only be computed over the candidate articles and not the whole of Wikipedia, for reasons discussed in Section 3.1. The procedure for computing the threshold t is explained in Section 5.2.

For simplicity the function output from this stage will henceforth be referred to as the *match* function, regardless of the specific similarity function used (the experiments in Chapter 5 are used to determine the best performing similarity function). The *match* function returns for each synset the best matching article (judged by the similarity metric) or null if no matches above the threshold could be found. For example the function might contain mappings such as these:

Synset: s	Article: match(s)
horticulturist	PLANTSMAN
hotdog	HOT DOG
house guest	*null*

3.5 Stage 3: Mapping refinement

The mapping approaches described in Section 3.4 included decisions that some synsets may not have a good matching article (differing from the candidate selection stage where the intention was simply to retrieve all possible matches). This raises the inevitable trade-off between precision and recall. The aim of this stage is to improve the precision of the overall synset-article mapping by removing incorrect matches. The idea is that a smaller, but more precise set of matches may be more useful than a large number of less precise matches. This is because typically the mapping is not an end in itself, but rather will be used for some other purpose, such as enriching WordNet for some external application. Thus inputting a small amount of high quality data would be preferable to a large amount of less reliable data (this hypothesis is tested in Chapter 6).

Therefore the aim of this stage is to refine by removing incorrect mappings. Two methods are used, both of which consider global information about the mappings between synsets and articles, rather than just consider each synset in isolation. The first method removes many-to-1 mappings from the set leaving only 1-to-1 mappings. The second method uses Wikipedia links as evidence of high quality mappings, removing those mappings which do not have links between each other.

3.5.1 1-to-1 mappings

In this stage, the global structure of the mappings are considered. The *match* mapping function is not an injective function; more than one synset may match the same article. It was found by inspection that in these cases many of the matches were incorrect. Figure 3.1 shows an example in which several synsets containing the word ‘tongue’ that are mapped to the ‘Tongue’ article in Wikipedia.

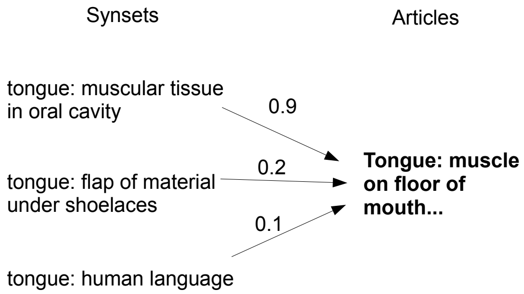

Synsets
Articles
tongue: muscular tissue in oral cavity
0.9
tongue: flap of material under shoelaces
0.2
0.1
tongue: human language
Tongue: muscle on floor of mouth...

Figure 3.1: Multiple synsets matching a single article.

Only one of these synsets, with the gloss ‘muscular tissue in oral cavity’, represents a correct match. One way to perform the reduction from many-to-1 to 1-to-1 is to keep only the article mapping with the highest similarity score. However initial experiments showed that this was not very effective, since different articles would often have the same similarity score, and therefore one mapping would have to be randomly chosen. Therefore a new mapping function is instead derived by simply eliminating all the many-to-1 matches, leaving a 1-to-1 mapping between synsets and articles. Note that this has the unwanted side effect of sometimes removing correct matches, thus lowering the overall recall performance; however the aim here is solely to improve precision performance.

The $1to1$ mapping can be formally defined in terms of an existing *match* mapping function (as defined in Section 3.4.3).

$$1to1(p) = \begin{cases} null & \text{if } \exists q \neq p :\ match(p) = match(q) \\ match(p) & \text{otherwise} \end{cases} \tag{3.8}$$

where $p, q : synset$, $a : article$ and $1to1 : synset \rightarrow article$.

3.5.2 Linked mappings

The next approach in refining the mappings is to exploit the links in Wikipedia to determine which of the synset-article mappings represent good matches. The idea behind this is that the links provide good evidence of which of the mappings are accurate. The hypothesis is that a synset where the mapped article is linked to (and from) another mapped article is more likely to be accurately mapped than not. The reasoning behind this can be explained as follows.

Consider the mapping function *match* as computed from the approaches in Section 3.4. Let S_{match} be the domain of this function, the synsets which are mapped to an article, and A_{match} be the range of the function. Furthermore, let $A_{correct}$ be the subset of A_{match} containing articles that have been *correctly* mapped from the corresponding synset. Since there are far fewer synsets than articles (82000 compared to 3 million), $|A_{match}|$ and is much smaller than $|A|$, i.e. most articles are not mapped to in the function, and of course $|A_{correct}|$ is smaller still.

Now consider a synset p that is correctly mapped to an article a (so therefore $a \in A_{correct}$) and another synset q that is *incorrectly* mapped to an article b ($b \in A_{match}$ but $b \notin A_{correct}$). The hypothesis is that a is more likely to link to other articles in A_{match} than b. This is because all synsets are related to at least one other synset, so we would expect the correctly mapped article a to also link to other mapped articles. In contrast the incorrectly mapped article b will be less likely to have these

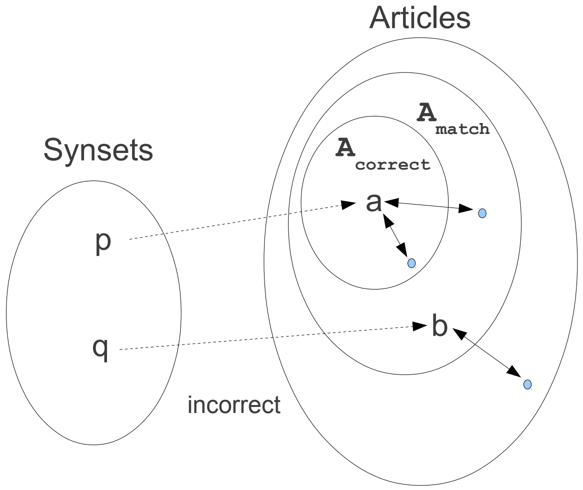


Figure 3.2: Hypothesis that correctly matched articles are more likely to be linked to other mapped articles than incorrect ones.

connections. This idea is illustrated in Figure 3.2.

Following from this hypothesis, the link refinement function *link* can be specified in terms of an existing *match* function (as defined in Section 3.4.3):

$$link(p) = \begin{cases} match(p) & \text{if } \exists x : \; x \in A_{match} \wedge link(match(p), x) \\ null & \text{otherwise} \end{cases} \tag{3.9}$$

where p : *synset*, a, b : *article*, *link* : *synset* → *article*, and *link* : *article* × *article* → *boolean*, with $link(x, y) = true$ iff there is an link from x to y.

The refinement approach eliminates mappings where there are no links between the mapped article and other mapped articles. The requirement can be further strengthened by requiring bidirectional links, i.e. the article must be linked both to and from another article in A_{match}. The *bilink* mapping is defined similarly to *link* except it requires the reciprocal link:

$$bilink(p) = \begin{cases} match(p) & \text{if } \exists x : \; x \in A_{match} \wedge \\ & link(match(p), x) \wedge link(x, match(p)) \\ null & \text{otherwise} \end{cases} \tag{3.10}$$

with $bilink : synset \rightarrow article$.

Figure 3.3 illustrates examples. There are no links to or from any of the mapped articles (including the thousands not shown in the figure) to the 'Exhumation' article - therefore the *exhumation* synset mapping is excluded from the *link* and *bilink* functions.

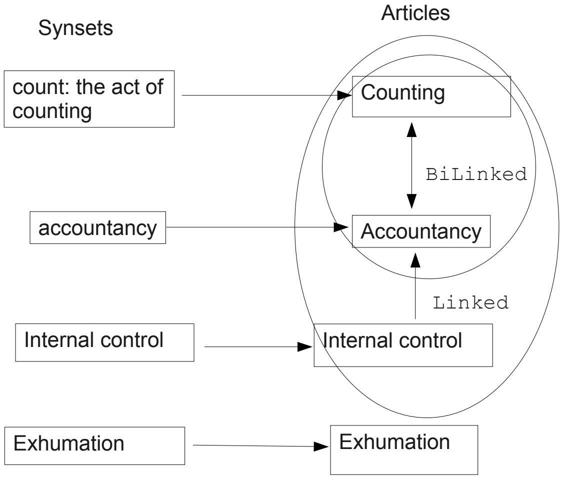


Figure 3.3: Links between articles

For the 'Internal control' article, there is a link to the 'Accountancy' article, but this is not reciprocated. This means that the *internal control* article would be included in the *link* mapping. However, assuming that no other bi-directional links exist with any of the other mapped articles, the *internal control* synset mapping would be excluded from *bilink*.

In contrast, there is a link from ‘Counting’ to ‘Accountancy’, and vice versa. Therefore the mappings from ‘count’ and ‘accountancy’ would be present both in the *link* and *bilink* mappings since the associated articles link to each other.

3.5.3 Output

As before the output of this stage is a mapping function from synsets to articles. These are derived from the *match* function from the previous section but with some mappings removed if they do not meet certain properties i.e. there will be more null mappings. The aim is that the remaining matches are of a higher quality than in the original mapping function.

3.6 Summary

An approach is described for mapping WordNet synsets to Wikipedia articles. The first stage uses title matching and information retrieval to find a set of candidate articles for each synset. This effectively reduces the search space allowing further methods to select the best matching article. The second stage uses text similarity methods to find for each synset the best matching article from the candidate article set. Methods for finding the similarity of the titles are also described. The result is a set of mappings from synsets to at most one article. Finally the third stage uses a global approach to refine the mappings. This eliminates many-to-1 mappings and uses Wikipedia links as evidence for good quality mappings. The aim of this refinement is to select a high quality set of precise mappings from the full set of mappings.

Chapter 4

Evaluation Methodology

A gold standard data set was created to evaluate the mapping methods of the previous chapter. This was done by manually annotating a random sample of 200 synsets with the appropriate matching article (if any), to create a gold standard **200NS** data set. Section 4.1 describes this annotation process and the 200NS set. The mapping methods were then measured against the gold standard set using standard metrics of accuracy, precision, recall and F-1 measure. The application of these metrics is described in Section 4.2.

For further evaluations, the gold standard of Ponzetto and Navigli (2010) was also used. This comprises 1000 articles which have been associated with 0 or more synsets as appropriate. The mapping methods proposed here are also tested on this gold standard data, which thus provides a completely independent evaluation of the methods. This gold standard and the evaluation methods are discussed in Section 4.4.

4.1 Annotation

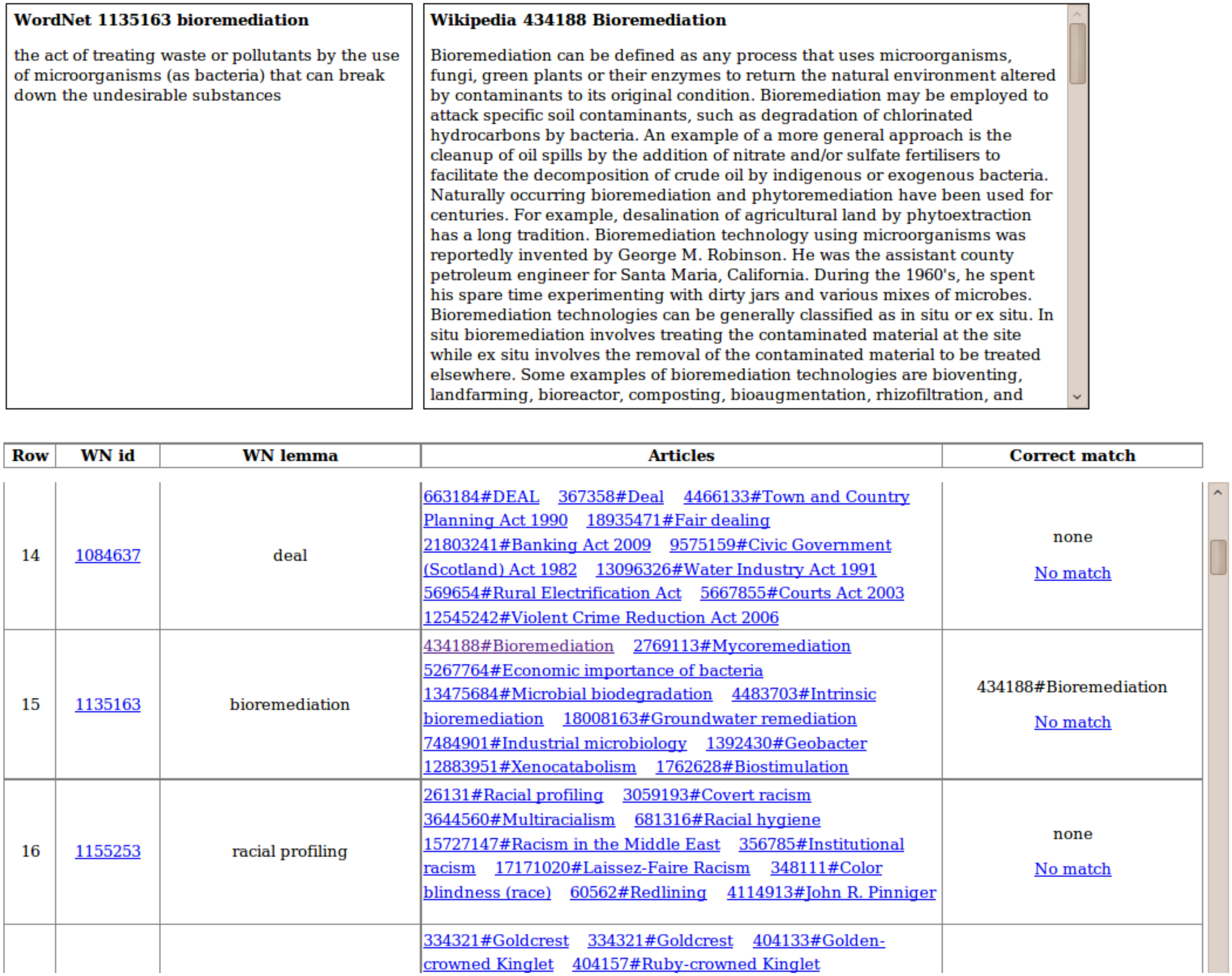
A set of 200 noun synsets were randomly chosen from WordNet. This was done by randomly choosing 200 distinct numbers from 1 to 82115 (the number of synsets) and then selecting the corresponding synsets.

Two annotators[1] were then asked to find for each synset the best matching article in Wikipedia. Both annotators were native English speakers. A web interface (Figure 4.1) was provided which gave for each synset a set of possible article matches, of which the annotators could select one as the best match. For each synset the interface shows one of the words in the synset[2] and the full gloss description. The article matches were generated using the title and article search approaches described in the Section 3.3. The candidate articles were listed next to the corresponding synset, and the annotators could then click each title to be shown the full text of the article in the main window. The interface would then record which article was eventually selected by the annotator. If none of the articles was a good match the annotators could then search Wikipedia manually. If an article was found outside the given candidate set this was noted separately by the annotator. Finally, if no appropriate article could be found then this could be noted on the interface by clicking on the 'No match' link.

Annotators were instructed to find the best matching article for each synset. In many cases this choice was straightforward. However, as discussed in the previous chapters, Wikipedia and WordNet have very different scopes and intentions. Typically articles are much more detailed and comprehensive than a synset, and will give a much broader context for the concept being described. This results in difficult borderline cases, where the decision on which (if any) is a good matching article is quite subjective. The key principle used was that in order to qualify as a match

[1] The thesis author and supervisor.

[2] If a similar annotation were conducted in the future it might be beneficial to display all words in the synset not just one. However in this case no problems were encountered.



WordNet 1135163 bioremediation

the act of treating waste or pollutants by the use of microorganisms (as bacteria) that can break down the undesirable substances

Wikipedia 434188 Bioremediation

Bioremediation can be defined as any process that uses microorganisms, fungi, green plants or their enzymes to return the natural environment altered by contaminants to its original condition. Bioremediation may be employed to attack specific soil contaminants, such as degradation of chlorinated hydrocarbons by bacteria. An example of a more general approach is the cleanup of oil spills by the addition of nitrate and/or sulfate fertilisers to facilitate the decomposition of crude oil by indigenous or exogenous bacteria. Naturally occurring bioremediation and phytoremediation have been used for centuries. For example, desalination of agricultural land by phytoextraction has a long tradition. Bioremediation technology using microorganisms was reportedly invented by George M. Robinson. He was the assistant county petroleum engineer for Santa Maria, California. During the 1960's, he spent his spare time experimenting with dirty jars and various mixes of microbes. Bioremediation technologies can be generally classified as in situ or ex situ. In situ bioremediation involves treating the contaminated material at the site while ex situ involves the removal of the contaminated material to be treated elsewhere. Some examples of bioremediation technologies are bioventing, landfarming, bioreactor, composting, bioaugmentation, rhizofiltration, and

Row	WN id	WN lemma	Articles	Correct match
14	1084637	deal	663184#DEAL 367358#Deal 4466133#Town and Country Planning Act 1990 18935471#Fair dealing 21803241#Banking Act 2009 9575159#Civic Government (Scotland) Act 1982 13096326#Water Industry Act 1991 569654#Rural Electrification Act 5667855#Courts Act 2003 12545242#Violent Crime Reduction Act 2006	none No match
15	1135163	bioremediation	434188#Bioremediation 2769113#Mycoremediation 5267764#Economic importance of bacteria 13475684#Microbial biodegradation 4483703#Intrinsic bioremediation 18008163#Groundwater remediation 7484901#Industrial microbiology 1392430#Geobacter 12883951#Xenocatabolism 1762628#Biostimulation	434188#Bioremediation No match
16	1155253	racial profiling	26131#Racial profiling 3059193#Covert racism 3644560#Multiracialism 681316#Racial hygiene 15727147#Racism in the Middle East 356785#Institutional racism 17171020#Laissez-Faire Racism 348111#Color blindness (race) 60562#Redlining 4114913#John R. Pinniger	none No match
			334321#Goldcrest 334321#Goldcrest 404133#Golden-crowned Kinglet 404157#Ruby-crowned Kinglet	

Figure 4.1: Interface shown to annotators.

the article should describe the synset concept (i.e. be exhaustive) and not describe other irrelevant concepts (i.e. be exclusive). However in some cases it was found that although there was no good match that satisfied both these conditions, there was nevertheless some article which was closely related to the synset. Therefore the annotation was extended to allow further categorisations of articles where no match could be found.

4.1.1 Synset-article relation categories

In some cases only part of an article matched the concept defined by the synset. In other cases there was an article which described a closely related concept. By identifying these relations it was possible to get a better understanding of the similarities and differences in the coverage of the two resources. The categories listed below were used to annotate each synset. Examples of each category are shown in Table 4.1; the 'Synset' column shows the synset lemma and extracts from the gloss, the 'Article' column shows an extract from the text of the article (where applicable), and the 'Category' column shows the manually assigned category from the 5 options.

1. Matching *article*. This indicates that the article is a match for the synset, exhaustively and exclusively describing the same concept as the synset. In the unlikely case that more than one article meets this requirement the best match is chosen. For example row 1 of Table 4.1 matches the synset about 'poaching' (as a cooking method) with the appropriate article.

2. Related *article*. No exact matching article can be found, but a closely related one can be found. These are divided into two types:

   (a) Part-of related - The synset corresponds to part of the article, but not the whole. If more than one article meets this requirement, the most strongly related is chosen. An example is found in row 5 there ‘tenon’ is described in part of the article about ‘Mortise and tenon’.

(b) Other related - This indicates that no matching article can be found but that there is an article directly related to the synset. If more than one article meets this requirement, the most strongly related is chosen. An example is in row 9 where ‘bath powder’ is a direct hyponym of ‘Powder’ as described in the article.

3. Not found. Where no article could be found, the annotators then classed the synset into one of two categories:

    (a) Dictionary term - The concept is one we might expect to find in a dictionary but not in an encyclopedia. An example is found in row 9, with the synset is ‘dumpiness’, related to the adjective for ‘dumpy’. This would not be an appropriate candidate for an encyclopedic article.

    (b) Not found - The concept is one we would expect to find in an encyclopedia, but cannot be found. For example in row 12, ‘vegetable sheep’ is a New Zealand herb but no exact reference could be found in Wikipedia.

4.1.2 Results

The initial inter-annotator agreement was 86% for both assigning the article and deciding on the category of relation or match. The annotators then discussed and resolved the disagreements to produce a final version of the 200NS data set. The distribution of categories for the 200 articles is shown in Table 4.2.

The majority of the synsets (63%) have a good matching article in Wikipedia. 27.5% of the synsets have a related article in Wikipedia (either part-of or other relation). These articles might provide possible sources for enriching the synset.

Row	**Category**	**WordNet Synset**	**Wikipedia Article**
1	Match	**poaching**: cooking in simmering liquid	**Poaching** is the process of gently simmering food in liquid...
2	Match	**catcher**, the position on a baseball team of the player who is stationed behind home plate...	**Catcher** is a position for a baseball or softball player...
3	Match	**chairlift**, a ski lift on which riders (skiers or sightseers) are seated...	An elevated passenger ropeway, or **chairlift**, is a type of aerial lift...
4	Match	**thumbstall**, protective covering for an injured thumb	A **finger cot** is a medical supply used to cover one or more fingers...
5	Part-of	**tenon**: a projection at the end of a piece of wood that is shaped to fit into a mortise and form a mortise joint	**Mortise and tenon** joint...The end of the first member is called the tenon, and it is usually narrowed with respect to the rest of the piece...
6	Part-of	**safe harbor**: the target company defends itself by making itself less attractive	**Safe harbor** has several usages... (Commerce) make acquisition by other parties unattractive.
7	Related	**ladies' tresses**, an orchid of the genus Spiranthes	**Spiranthes**, commonly called Ladies-tresses, is a genus of orchids
8	Related	**bath powder**: a fine powder for spreading on the body	A **powder** is a dry, bulk solid composed of a large number of very fine particles...
9	Dict	**dumpiness**, a short or stout physique	
10	Not found	**vegetable sheep**, cushion-forming New Zealand herb	

Table 4.1: Examples of manually annotated mappings between WordNet synsets and Wikipedia articles. The synset column includes one lemma word and an extract of the gloss. The article column includes the first paragraph of the article text. **Bold** text highlights the lemma word and article title respectively.

Category	Synsets
1 - Match	126 (63%)
2a - Part-of related	11 (5.5%)
2b - Other related	36 (18%)
3a - Dictionary term	23 (11.5%)
3b - Not found	4 (2%)
Total	200 (100%)

Table 4.2: Distribution of synsets into categories.

However care would have to be taken to exclude irrelevant information in the article or to determing the exact relation. If such a relation already existed in WordNet, then the article should be aligned with the related synset before information was added to WordNet. 13.5% of the synsets have no related articles in Wikipedia, because they are either dictionary terms (11.5%) which would not be expected in an encyclopedia or were simply not found (2%).

4.1.3 Discussion

The results suggest that the majority of synsets in WordNet have a Wikipedia article that is broadly similar in meaning. However it is clear that Wikipedia and WordNet have substantially different coverage. The annotators found several issues when identifying correct matching articles for synsets. These issues help clarify the differences between the two resources which naturally arise as a result of their different objectives.

- Polsemy in Wikipedia. There may be several articles which have titles containing one of the lemmas in the synset, or which have some other similar content. For example there are two articles about 'Poaching', one referring to the act of taking wild animals or plants, and the other about the method of cooking. This issue is dealt with in Wikipedia using parentheses after the titles (e.g. ‘Poaching (cooking)’) and also by the disambiguation pages that list and disambiguate different articles with similar titles. For some cases the meanings of the different articles were quite close, especially for the botanical terms where the annotators did not have specialised knowledge of the field. In these cases disagreements were resolved after discussion.

- Synonyms not covered by WordNet. For example ‘cattleship’ in WordNet is known as ‘Livestock carrier’ in Wikipedia. These are sometimes difficult to find, even for human users. In this case the article in Wikipedia was found by looking at the gloss of the synset (‘a cargo ship for the transport of livestock’).

- Missing meanings. This is where a WordNet concept is not present in Wikipedia. For example ‘tablespoon’ in WordNet refers to the quantity (one tablespoonful), but the closest related Wikipedia article is the one which describes the object itself. It would be considered incorrect to match these two.

- Difference in scope. As identified above sometimes only part of the article will match the concept defined by the synset, with the rest of the article being irrelevant. For example, there is an article in Wikipedia on ‘Mortise and tenon’ joints, while WordNet contains the concepts ‘mortise’ and ‘tenon’ in two separate synsets.

4.2 Evaluation metrics

In all experiments only the ‘Match’ category is considered as a positive match for a synset. Related and part-of articles are considered as a ‘no match’. This imposes strict standards for the evaluation since even closely related articles are not considered as correct mappings. Also at most one article is selected for each synset as the correct mapping, even though other close matches may exist.

Using this approach collapses the categorisation to a simple binary case of match or no-match. Once the categories are collapsed into the binary case this way, the annotated set can be considered to be an instantiation of the gold standard $match_{ideal} : synset \rightarrow article$ function (defined in Section 3.1) for the 200 synsets in the 200NS set.

The candidate articles generated from the methods in Section 3.3 are evaluated in terms of recall against the gold standard data. The mapping methods in Section 3.4 and Section 3.5 are evaluated using precision, recall, accuracy and F1 measure against the gold standard data. These are standard metrics which have been widely used in language processing evaluations (Olson and Delen, 2008). The specific method of application of the metrics for the mapping methods over the gold standard data is explained in this section.

4.2.1 Evaluating recall for candidate articles

The aim is for the candidate article sets retrieved using the methods of Section 3.3 to include the correct matching article (where there is one). The candidate sets are evaluated in terms of recall against the $200NS$ set. Let $200NS_{match}$ be the set of 126 synsets where a matching article was found.

To help give a better comparison of the different candidate retrieval approaches, each output candidate set is ranked in order, with those most likely to be matches occurring at the start of the list. This results in an ordered sequence $cand_{seq}$ rather than the unordered set of articles $cand$ for each synset. This then allows measurement of recall up to the ith value in each sequence, a standard metric for recall at n (Baeza-Yates et al., 1999).

$$
\begin{aligned}
recall_{seq}(i) &= \frac{\sum\limits_{s \in 200NS_{match}} score_{seq}(s,i)}{|200NS_{match}|} \\
&= \frac{\sum\limits_{s \in 200NS_{match}} score_{seq}(s,i)}{126}
\end{aligned}
\tag{4.1}
$$

where the $score_{seq}$ function is defined as:

$$
score_{seq}(s,i) = \begin{cases} 1 & \text{if } \exists x.\ x \leq i \wedge cand_{seq}[x] = match_{ideal}(s) \\ 0 & \text{otherwise} \end{cases}
\tag{4.2}
$$

where $cand_{seq}[x]$ is the xth article in the sequence.

This recall metric can be illustrated using a miniature gold standard example, $5NS$. Consider 5 synsets $[p,q,r,s,t]$ with correct mappings respectively $[a,b,c,d,null]$. Given the following candidate sequences:

$$\overbrace{\hphantom{1\quad 2\quad 3}}^{cand_{seq}}$$

Synset	1	2	3	Correct
p	x	y	$\boldsymbol{a}$	a
q	$\boldsymbol{b}$	x	y	b
r	x	$\boldsymbol{c}$	y	c
s	x	$\boldsymbol{d}$	y	d
t	x	y	z	$null$

$recall_{seq}(1) = \frac{1}{4} = 0.25$ since only 1 correct match (b) is correctly found in column 1.

$recall_{seq}(2) = \frac{3}{4} = 0.75$ because 3 matches (b, c, d) are found in columns 1 and 2.

$recall_{seq}(3) = \frac{4}{4} = 1$ since all 4 matches (a, b, c, d) are found in the 3 columns.

Here the *null* mapping for t is irrelevant since we are only looking at recall of positive matches.

4.2.2 Evaluating mappings

The aim of this evaluation is to compare automatically generated mappings against the gold standard. Where there is a good matching article, the mapping method should identify this correctly. The mapping method should also correctly identify when no mapping (or null) is found for a synset.

This section describes how standard evaluation metrics for a given *match* function are calculated over the $200NS$ sample set. Given a particular *match* function, the **precision** metric indicates what fraction of the labelled positive (i.e. non-null) mappings are correct. In contrast the **recall** metric indicates what fraction of the positive mappings from the gold standard data are correctly identified by the *match* function. The precision and recall metrics are averaged in the **F1-measure** to give an overall measure of performance. Finally the **accuracy** metric simply identifies what fraction of the mappings in the *match* function are correct.

Firstly we can identify those labelled positive by *match* within $200NS$ as $match_{pos}$:

$$match_{pos} = \{s | s \in 200NS \wedge match(s) \neq null\} \tag{4.3}$$

Then we can identify the true positives using the gold standard $match_{ideal}$ function:

$$match_{tp} = \{s | s \in match_{pos} \land match(s) = match_{ideal}(s)\} \tag{4.4}$$

This allows us to find the precision of the *match* mapping as the number of true positives divided by the number of those labelled positive by *match*:

$$precision = \frac{|match_{tp}|}{|match_{pos}|} \tag{4.5}$$

To find the recall we require the number of positive matches in the gold standard. We can reuse the $200NS_{match}$ from the previous section, giving *recall* as the number of true positives divided by the number of gold standard positives:

$$recall = \frac{|match_{tp}|}{|200NS_{match}|} = \frac{|match_{tp}|}{126} \tag{4.6}$$

This is similar to the recall metric for the candidate articles, except there is at most one article present in the mapping, which simplifies the calculation. To combine precision and recall we can then use the standard $F1$ measure:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.7}$$

Finally for accuracy we require the correct mappings found in $200NS$ (including *null* values). This is defined as:

$$match_{correct} = \{s | s \in 200NS \land match(s) = match_{ideal}(s)\} \tag{4.8}$$

This gives the accuracy as the number of correct mappings divided by the total number of synsets in $200NS$:

$$accuracy = \frac{|match_{correct}|}{|200NS|} = \frac{|match_{correct}|}{200} \tag{4.9}$$

We can again illustrate these metrics using the simple miniature gold standard set $5NS$, where synsets $[p, q, r, s, t]$ map to $[a, b, c, d, null]$ respectively. Consider the following mappings:

Synset	Match	Correct
p	$\boldsymbol{a}$	a
q	$\boldsymbol{b}$	b
r	x	c
s	$null$	d
t	$\boldsymbol{null}$	$null$

Then we have $precision = \dfrac{|match_{tp}|}{|match_{pos}|} = \dfrac{2}{3} = 0.67$, since there are 3 positive labels (a, b, x) of which two are correct.

Then $recall = \dfrac{|match_{tp}|}{|5NS_{match}|} = \dfrac{2}{4} = 0.5$, since there are 4 positive instances (a, b, c, d) in the gold standard set, of which 2 are correctly identified.

From this $F1 = 2 \cdot \dfrac{precision \cdot recall}{precision + recall} = 0.57$.

Finally $accuracy = \dfrac{|match_{correct}|}{|5NS|} = \dfrac{3}{5} = 0.6$, because there are 3 correct matches out of the 5 instances.

4.3 Statistical significance

The aim of statistical significance testing when comparing the evaluated task performance of two different methods is to determine if the better performance is achieved by chance or not. If the probability of the better performance being achieved by chance is below a certain threshold, than the conclusion is that the better performance was not just the result of chance, and therefore the system genuinely is better.

There are several different types of experiment and evaluation methodologies presented in the following chapters. In Chapter 5 the candidate retrieval process is evaluated by recall against the gold standard. The mapping methods are evaluated by recall, precision, and accuracy. In Chapter 6 performance of derived relations from the mappings are tested for accuracy on a word sense disambiguation system.

However all experiments can be considered as a series of independent experiments over single test instances which can yield only a success or failure (a synset is mapped correctly to an article or not; a word is disambiguated correctly or not). As such all experimental evaluations in this thesis can be considered binomial distributions - therefore the most applicable statistical test is the binomial test for significance (Upton and Cook, 1997).

The binomial test can best be explained using an example. Consider an experiment using a 1000 test instances. A baseline method X achieves accuracy of 80%, i.e. 800 correct. A new method Y achieves accuracy of 82.7%, i.e 827 correct. The question is then whether method Y is genuinely better than method X, or if it has achieved this result by chance alone.

The *null hypothesis* is that method Y is no better than method X. However if it can be shown that the probability of method Y achieving that result by chance is sufficiently low then the null hypothesis is rejected and the alternative hypothesis is accepted instead; that method Y is more accurate than method X. The threshold probability is called the *significance level.* The choice of significance level is somewhat arbitary but generally a level of 5% is used. Therefore if a probability of less than 0.05 is achieved than the result is determined to be statistically *significant.*

Since method X achieved 80% accuracy, the estimated probability of correctly classifying a single instance is 0.8. Then assuming the null hypothesis, we consider method Y to be equivalent to method X, i.e. the probability of getting a single instance correct to be 0.8. The question is then what the probability of method

Y getting 827 out of a 1000 instances correct, given this assumption. To calculate this the binomial distribution is consulted, with $n = 1000$ (number of instances), and $p = 0.8$. The probability of getting exactly k instances correct is given by the following formula:

$$\binom{n}{k} p^k (1-p)^{n-k} \tag{4.10}$$

However it is required to repeat this calculation to give a cumulative result to find the probability of achieving 827 or more correct. This probability is found to be 0.0168, therefore the result is significant ($p < 0.02$). This low probability indicates that method Y is genuinely better than method X and the result was not simply a result of chance.

4.4 Independent evaluation data

Ponzetto and Navigli (2010) develop their own gold standard data set for evaluation. Their work also involves mapping WordNet synsets[3] to Wikipedia articles, and so is similar to that presented in this thesis. However there are a number of differences in their aims, and this is reflected in their gold standard data. The most important difference is the direction of the mapping. Their aim is to find for each Wikipedia article the best possible synset match, in contrast to this thesis which has the mapping in the opposite direction. The gold standard has for each article 0 or more synsets which are judged to match the concept of the article.

The mappings in this thesis can be evaluated against this gold standard data as described in Ponzetto and Navigli (2010). However it is worth noting that because of the different direction of their mappings they have an advantage, since they allow

[3] In fact Ponzetto and Navigli (2010) map from articles to *sense labels* rather than synsets; however when a sense label is associated with a lemma it uniquely identifies a synset, and therefore the term can be used interchangeably.

multiple articles to match to a single synset, thus giving them more ‘chances’ to get the right match. As noted at the start of Section 3.1 it is unlikely for more than one article to match a single synset using the strict standard defined in this thesis; therefore it is likely that many of these article matches are incorrect when judged by the same standard. This is discussed in more detail in Section 5.4.

4.4.1 Gold standard data

The gold standard data[4] from Ponzetto and Navigli (2010) comprises 1000 Wikipedia articles which have been manually assigned 0 or more associated synsets which represent good matches for the article. This will be henceforth be referred to as the $1000A$ set.

The number of matches for each synset is shown in Table 4.3.

Synset matches	Number of articles
0	502
1	448
2	48
3	2

Table 4.3: Number of articles with number of synset matches

There are 50 articles with more than one synset mapping. This does not cause a problem since the mapping methods of this thesis can map more than one synset to each article. However the converse (one synset mapping to more than one article) is problematic, since the assumption here is that there is at most one good article for each synset. In the gold standard data there are only 3 synsets which are mapped to by more than one article. Since there are only a few of these cases it was decided

[4] Note that the gold standard data differs slightly from that described in the publication - the authors fixed a few consistency errors before publishing the data online

to manually select one best article for each case. These are listed below:

- Synset 3130340: crenel, crenelle – (a notch or open space between two merlons in a crenelated battlement). Maps to Wikipedia articles entitled CRENEL and CRENELLE. Both these titles (in the Wikipedia snapshot used by Ponzetto and Navigli (2010)) are in fact redirects to the same article EMBRASURE. Both terms 'Crenel' and 'Crenelle' seem to be used equally frequently, therefore it is arbitarily decided to discard 'Crenel' and keep 'Crenelle'.

- Synset 1187620: naturalization, naturalisation – (the proceeding whereby a foreigner is granted citizenship). Maps to articles NATURALIZATION and NATURALISATION. Here 'Naturalisation' redirects to the NATURALIZATION article. Therefore the 'Naturalisation' match is discarded.

- Synset 2581957: dolphinfish, dolphin, mahimahi – (large slender food and game fish widely distributed in warm seas (especially around Hawaii)). Maps to articles DOLPHINFISH and MAHIMAHI. 'Dolphinfish' redirects to CORYPHAENIDAE (a family of marine ray-finned fishes belonging to the order Perciformes). 'Mahimahi' redirects to MAHI-MAHI (The mahi-mahi or common dolphinfish[1] (Coryphaena hippurus) is a surface-dwelling ray-finned fish found in off-shore temperate, tropical and subtropical waters worldwide). The MAHI-MAHI article provides a better match in this case therefore is kept over the DOLPHINFISH article.

With these 3 articles eliminated, there are now 997 articles with associated synset matches. Additionally 9 articles could not be found in Wikipedia, and therefore the mapping methods could not possibly match them. Also there is a duplicate article-synset match (TOPONYMY). Listing synset matches separately this gives $448 + (48 * 2) + (2 * 3) - 9 - 1 - 3 = 537$ article-synset matches and 502 articles with no synset match, giving a total of 1039 pairings. This data can now be used as an additional intrinsic evaluation for the mapping methods.

4.4.2 Recall for candidate articles

The retrieval methods create a set of candidate articles for each synset. The gold standard data of Ponzetto and Navigli (2010) comprises article-synset pairings. For each synset in this data, the aim is to see if the associated article occurs in the candidate article set for that synset. As for the $200NS$ set the $recall_{seq}$ metric is used to determine recall at n for each synset.

4.4.3 Evaluating mappings

The mappings in this thesis are evaluated in exactly the same way as in Ponzetto and Navigli (2010). The precision is calculated as the ratio of correct synsets to the total number of non-empty labels output by the mapping algorithms. The recall is calculated as the ratio of correct synsets to the total number of non-empty labels in the gold standard. The F-measure is calculated in the usual way to combine precision and recall ($\frac{2PR}{P+R}$). Finally the accuracy is the number of correct sense labels divided by the total number of instances, which (as opposed to the other metrics) takes into account empty mappings.

4.5 Summary

A random sample of 200 noun synsets from WordNet were annotated with a matching article from Wikipedia or with other tags if no matching article could be found. The results from this process suggest that the majority of synsets (63%) have a good matching article in Wikipedia that describes the same concept.

The candidate articles are evaluated in terms of recall against the gold standard test data. The mappings are evaluated using standard metrics of precision, recall, accuracy and F1 against the test data.

Chapter 5

Creating and evaluating mappings

This section describes experiments evaluating the mapping methods from Chapter 3. The test data set and evaluation approaches from Chapter 4 are used in these experiments. For all experiments the Wikipedia snapshot of 3rd November 2009 was used. This snapshot of Wikipedia was imported into a MySQL database to allow quick access to the article content. In all sections the mappings are evaluated on the $200NS$ evaluation data except for Section 5.5 where the mappings are evaluated on the $1000A$ set.

Section 5.1 evaluates the performance of the candidate article retrieval process from Section 3.3. A set of candidate articles which may be good matches for each synset is retrieved for each noun synset using information retrieval and title matching methods. Section 5.2 evaluates the mapping selection methods from Section 3.4. These use text similarity metrics to find the best matching article from the candidate article set. Section 5.3 employs the refinement methods from Section 3.5 to select a more precise set of mappings. Section 5.4 compares the experimental results with that of similar previous work. Section 5.5 evaluates the mappings on the independent $1000A$ gold standard set. Finally the chapter is summarised in Section 5.6.

5.1 Stage 1: Candidate article retrieval

This section gives evaluation results of the first stage in the synset-article mapping process. The methods described in Section 3.3 are employed to retrieve a set of candidate articles for each noun synset in WordNet. The aim is for the correct matching article to be included amongst this set. For each method, the aim is to try to order the candidate articles best matching first. Performance is evaluated using the $recall_{seq}$ metric from Section 4.2.1. This calculates the recall of matching articles for various points in the candidate sequence. Let N be the number of candidate articles that are returned for each synset. Increasing N increases the chance of including the correct article, but will increase the search space for the subsequent mapping methods (and thus may reduce the chance of finding the correct match).

5.1.1 Title matching

The approach described in Section 3.3.1 is used to retrieve candidate articles for each synset. Articles are retrieved where titles match any of the lemmas representing each synset. This is done by searching the title field in the SQL database. The following combinations are tested.

- Articles (A) only - Use only articles whose title matches one of the lemmas.
- Articles (A), redirects (R) - As above but also follow any redirects.
- Articles (A), all disambiguation links (D) - Add all links from disambiguation pages
- Articles (A), redirects (R), all disambiguation links (D).

The retrieved articles are ordered in the sequence as given above, so for the last combination, this would be articles (A) then redirects (R) then disambiguation links (D).

Articles	1	2	5	10	20
A	60.3	67.5	68.3	68.3	68.3
A+R	69.8	76.2	76.2	76.2	76.2
A+D	61.9	68.3	70.6	70.6	70.6
A+R+D	**71.4**	**77.8**	**79.4**	**79.4**	**79.4**

Table 5.1: Title matching performance evaluated on $200NS$.

The recall performance for each method is shown in Table 5.1. These show that redirect pages provide a significant boost to recall (binomial test, $p < 0.05$). The disambiguation links also improve performance but not as effectively as the redirects and the improvement is not statistically significant. This reflects on the nature of the redirects and the disambiguation links in Wikipedia. Redirects can be considered as synonymous terms for the article title, but disambiguation links list many different concepts for an article, therefore we would expect redirects to be more likely to be matches overall.

In the first row, using title matched articles plateaus at 68.3% after 5 articles. Note that in most cases the number of articles returned by this method will be limited to the number of lemmas in the synset. However in some cases it will retrieve articles with different spelling variations. For example one synset has two lemmas: {enzyme-linked-immunosorbent serologic assay, ELISA}. The title matching method retrieves 2 articles - ELISA (the correct match), and also Élisa, which redirects to a disambiguation page. In any case there are typically only very few articles retrieved, which explains why performance plateaus so quickly.

The redirect and disambiguation pages shown in the second and third rows both plateau after 5 articles. The redirects typically do not retrieve more than one or two further articles. However highly polysemous terms may have many possible disambiguation links - the table suggests that the correct match is found within the first few if at all.

Further analysis of the results shows how many articles are retrieved using the different title matching methods (Table 5.2). The T and T+R methods receive at most a handful (≤ 5) candidate articles for each synset, with most cases receiving 2 or fewer candidate articles. In contrast when the disambiguation links are added a substantial number of synsets have more than 5 candidate articles. This reflects on the polsemy of the synset terms in Wikipedia. The extreme case was for the synset 'cone' where 154 candidate articles were found, mostly disambiguation links reflecting the highly polysemous nature of the word.

Total number of candidate articles	T	T+R	T+D	T+R+D
0	65	31	65	31
1	98	111	80	97
2	31	43	17	26
3	4	12	5	8
4	2	1	2	2
5	0	2	0	0
> 5	0	0	31	36

Table 5.2: For each title-matching method the table shows the distribution of the number of retrieved candidate articles over the 200 synsets.

5.1.2 Information retrieval results

The information retrieval approach (as described in Section 3.3.2) is used with different features from each WordNet synset to form queries. The Terrier software (Ounis et al., 2007) was used for IR over the Wikipedia snapshot. This was indexed using Terrier, treating each page as a document in the collection. Terrier offers a variety of weighting models for retrieval. The one used here was the widely used TF-IDF model (Spärck Jones, 1972). The candidate articles are ordered as they are retrieved by Terrier which returns the most relevant articles first. The following query combinations are used:

- Lemmas (L) of the synset.
- Lemmas (L) + the gloss of the synset (G).
- Lemmas (L) + lemmas of related synsets (RL).
- Lemmas (L) + gloss (G) + lemmas of related synsets (RL).

Note that 'related synsets' here are considered to comprise all synsets immediately related to the synset (as described in Section 3.3.2). The results are shown in Table 5.3.

Articles	1	2	5	10	20
L	48.4	59.5	69.8	78.6	82.5
L+G	**58.7**	**73.8**	**84.9**	88.9	91.3
L+RL	42.9	57.1	75.4	84.1	86.5
L+G+RL	54.0	69.0	**84.9**	**91.3**	**93.7**

Table 5.3: Recall against number of articles for IR methods over $200NS$.

Glosses appear to be slightly more effective than related lemmas as queries. However the best results are achieved using all in combination: (L+G+RL). Using the (L) method as the baseline, the (L+G+RL) method is significantly better (binomial test, $p < 0.01$).

5.1.3 Comparison of approaches

Comparing the results from the IR approaches with the title matching approaches there are several observations. Firstly, the results do not plateau by 20. Therefore, retrieving more articles may increase the recall. However for the reasons discussed at the start of the section, more candidate articles may have a detrimental effect on the subsequent mapping methods. Secondly, the title matching results are better for the first and second articles. This implies that the title alone is the best single indicator of a good match. Thirdly, when retrieving 5 articles or more, the recall of the IR approaches exceeds the title matching results. This implies that although the title is the best indicator, it is not always sufficient, and searching the full text is sometimes required to find a good match. Finally, the best performing IR method L+G+RL is significantly better than the best performing title matching method A+R+D (binomial test, $p < 0.01$).

5.1.4 Combining title matching & IR

The next experiments use different combinations of these methods to determine which give the best performance. The two best performing title matching approaches (A+R) and (A+R+D) are combined with the two best IR approaches (L+G) and (L+G+RL). Since title matches gave the best recall results for 1 or 2 articles, the articles retrieved from the title matching approach are used first in the sequence, followed by the articles from the IR methods.

The results are shown in Table 5.4.

Articles	1	2	5	10	20
(A+R)+(L+G)	**79.4**	**90.5**	**94.4**	**97.6**	**97.6**
(A+R+D)+(L+G)	74.6	84.9	92.1	92.1	92.9
(A+R)+(L+G+RL)	75.6	87.3	92.9	96.0	96.0
(A+R+D)+(L+G+RL)	76.2	86.5	91.3	94.4	95.2

Table 5.4: Recall against number of articles combining title matching & IR methods on 200NS.

All results are very close, and there is no statistically significant difference between the performance of the methods. Comparing results with and without the disambiguation links (D) it seems that the IR approaches are more likely to return relevant results than the disambiguation links. The best overall result is from the (A+R)+(L+G) approach, which achieves 97.6% recall after 10 articles. Comparing this result against the best result for the IR approaches (93.7%), this is not a statistically significant improvement.

Following from the analysis in Table 5.2, an observation is that the IR methods effectively plug gaps where the title matching approach cannot find candidate articles. Since the title matching results are retrieved first it is possible to directly see the results of this. For instance when just 1 article is retrieved the (A+R)+(L+G) method achieves 79.4% compared to 69.8% for the (A+R) method. From Table 5.2 we can see that for the A+R approach there were 31 synsets no article was found. From the figures we can see that the correct article is found by the IR approach in 19 of these 31 cases. This shows a substantial improvement over previous work which has relied on the title of the articles alone when matching with synsets.

5.2 Stage 2: Selecting the best mapping

The previous stage returned a set of candidate articles for each noun synset. The best performing method (A+R+L+G) was used to retrieve 20 candidate articles for all 82115 noun synsets, with estimated recall of 96% of matching articles within the candidate set. The work described in this section employs the methods from Section 3.4 to select the best article from this candidate set for each synset.

All methods in this section assign a score to each article in the candidate set based on similarity with the synset. The article with the highest score is chosen as the best match. However this score must exceed a threshold for the article to be assigned as a positive match, otherwise it is decided that there is no match for the synset. This is an important difference from the previous section, where the aim was simply to retrieve the best match for each synset. Here it is necessary to decide that there may not be a good enough matching article for a given synset.

Thresholds are determined using the gold standard mappings as training data. To ensure that there is no bias 10-fold cross validation is used, ensuring the test data is not used in the threshold estimation. To estimate the threshold the J48 decision tree classifier is employed in Weka (Hall et al., 2009). This generates a simple two-branch decision tree which splits the data at the best threshold point, determined as the point which maximizes accuracy on the training data.

5.2.1 Text similarity

This uses the text similarity approach from Section 3.4.1 to assign scores to each article in the candidate set. The whole Wikipedia text is used for each article. The articles were pre-processed to remove Wiki markup from the text. Different features from the synset are used for the comparison (the same combinations as from Section 5.1.2):

- Lemmas (L) of the synset.
- Lemmas (L) + the gloss of the synset (G).
- Lemmas (L) + lemmas of related synsets (RL).
- Lemmas (L) + gloss (G) + lemmas of related synsets (RL).

Different variations of the approach were tested using combinations of the following:

- (Stem) Stemming with the Porter stemmer.
- (Stop) Common stopwords were removed.
- (IDF) TF-IDF weighting for each term. IDF weights are computed from the BNC corpus, and the term frequency (TF) is the frequency of the term in synset. For this variation, instead of computing text similarity using equation 3.4, this equation is used instead to incorporate the TF-IDF weights:

$$text\_sim(A,B) = \frac{\sum\limits_{x \in |A \cap B|} tf(x) \times idf(x)}{min(|A|,|B|)} \tag{5.1}$$

Accuracy, precision, recall and F1-measure were calculated using the methods in Section 4.2.2. The full set of results is given in Table 5.5. The column headings show the combination of features used in the similarity metric. The row headings show which pre-processing steps are used (stemming, stopword removal, IDF weighting).

Adding the gloss and related lemmas degrades performance. At first this is somewhat surprising, as it might be expected that this additional information would improve performance. However it seems that it simply adds noise, and that the lemmas by themselves are the most salient features of the synset. Stemming and removing stopwords seem to slightly improve performance, but the TF-IDF weighting

Overlap	Metric	L	L, G	L, RL	L, G, RL
Tokens	Accuracy	54.9	36.2	40.8	38.6
	Precision	52.9	36.7	39.6	34.9
	Recall	70.0	53.7	55.8	44.2
	F-measure	60.3	43.6	46.3	39.0
Stem	Accuracy	55.0	33.6	40.8	42.1
	Precision	52.8	34.0	38.8	36.0
	Recall	**73.8**	53.1	50.3	42.5
	F-measure	61.6	41.5	43.8	39.0
Stop	Accuracy	**55.7**	38.7	38.8	38.3
	Precision	**53.7**	38.9	38.2	38.1
	Recall	70.8	59.7	53.4	53.7
	F-measure	61.1	47.1	44.5	44.5
StemStop	Accuracy	55.5	39.8	36.5	44.6
	Precision	53.4	38.4	35.4	40.0
	Recall	**73.8**	49.3	52.3	53.4
	F-measure	**62.0**	43.2	42.2	45.7
IDF	Accuracy	49.0	43.0	32.4	41.6
	Precision	49.2	41.8	32.5	38.8
	Recall	61.0	49.5	51.4	45.8
	F-measure	54.5	45.3	39.8	42.0
StemIDF	Accuracy	46.2	41.2	29.7	30.5
	Precision	47.1	37.5	30.0	30.0
	Recall	71.5	51.6	46.7	41.2
	F-measure	56.7	43.4	36.5	34.7
StopIDF	Accuracy	46.6	35.0	32.8	38.0
	Precision	47.8	35.9	33.1	35.9
	Recall	64.8	53.0	51.7	42.6
	F-measure	54.9	42.8	40.4	38.9
StemStopIDF	Accuracy	46.6	33.8	31.2	36.6
	Precision	47.3	34.2	31.0	33.1
	Recall	72.9	48.7	45.6	35.6
	F-measure	57.4	40.2	36.9	34.2

Table 5.5: Accuracy, precision, recall and F-measure on 200NS (%) with different text similarity methods and pre-processing steps. Using 20 candidate articles.

appears to be detrimental. This may be because the synsets are relatively small and thus all information is important when calculating the similarity, even fairly common terms; applying the TF-IDF weighting would reduce the weighting of these terms in the calculation.

Using L+G accuracy as a baseline (36.2%), the L method achieves a statistically significant improvement (54.9%, binomial test, $p < 0.001$). However comparing the result for stopword removal (55.7%) to the result without (54.9%) the improvement is not statistically significant.

Reducing candidate articles

The previous experiment used 20 candidate articles from which the best mapping was chosen. The results from the previous section showed that the recall at 10 candidate articles was the same as for 20 (97.6% recall). Therefore we might expect if we reduced the number of candidate articles available to the mapping to 10 (or even further) the precision might increase, since there would be a smaller and higher quality search space from which the mapping methods can select the best article.

To measure this effect, experiments were performed which varied the number of candidate articles available to the mapping method. The text similarity method using lemmas (L) with stemming and stop word removal was chosen since this had the highest F-measure performance.

The results show that reducing the number of candidate articles increases precision. Recall also improves up to a point - the recall with 5 candidate articles is the highest. The highest overall F-measure is achieved with 2 candidate articles.

5.2.2 Title similarity

The title similarity approach described in Section 3.4.2 is used to score each article in the candidate set. The results are shown in Table 5.7. Again the number of

Candidates	Accuracy	Precision	Recall	F-measure
1	**61.5**	**62.1**	69.0	65.4
2	61.0	60.0	76.2	**67.1**
5	60.5	58.4	**77.0**	66.4
10	56.5	54.4	73.8	62.6
20	55.5	53.4	73.8	62.0

Table 5.6: Measuring the effect of varying number of candidate articles on performance over $200NS$

candidate articles is varied to measure the effect this has on performance.

Candidates	Accuracy	Precision	Recall	F-measure
1	**67.0**	**69.4**	61.1	**65.0**
2	65.5	67.5	61.1	64.2
5	65.5	67.5	61.1	64.2
10	65.5	67.5	61.1	64.2
20	65.5	67.5	61.1	64.2

Table 5.7: Title similarity on $200NS$

This shows that just using the title alone when comparing with the synset gives better results than using any of the text similarity approaches. The improvement in accuracy achieved here (67.0%) when compared with the best result using the text similarity approach (lemmas, 61.5%) is statistically significant (binomial test, $p < 0.01$).

Using just 1 candidate article gives the best performance. This shows that the first article retrieved is most likely to be the best match. After the 2nd article there is no difference in performance. This is because title matches are to be found in either 1st or 2nd place if at all.

5.2.3 Combining title and text

Finally the two approaches are combined. For the text similarity approach, the best performing method was used: lemmas as features, stemming and stopword removal. This is then combined with the title similarity metric by taking a simple average of the two scores:

$$combined = \frac{title\_sim + text\_sim}{2} \tag{5.2}$$

The results are shown in Table 5.8.

Candidates	Accuracy	Precision	Recall	F-measure
1	67.0	70.0	61.1	65.3
2	69.5	77.5	62.7	69.3
5	69.5	77.5	62.7	69.3
10	69.5	77.5	62.7	69.3
20	69.5	77.5	61.2	67.5

Table 5.8: Title similarity + text sim on 200NS

These show better results than either method used alone. This shows that although the title is the most important feature of the article, examining the article content improves performance. This reflects the results from the candidate article retrieval in Section 5.1. However comparing the combined method accuracy at 2 candidate articles (69.5%) with the result using title similarity alone (65.5%) shows the improvement is not statistically significant. It is worth noting that when a title match is found amongst the candidate articles the title similarity score is much higher than the text similarity scores. This means that the combined method effectively only comes into play when no articles are found using the title matching method (an issue discussed in Section 5.1.1). For these cases the articles will be those returned by the IR methods and these are ranked by the text similarity score.

Using 2 candidate articles gives better results than 1. However the results plateau after 2 articles and then drops very slightly after 10 articles. This again indicates that the title is the single best indicator of a match. When a title match cannot be found, then the text similarity gives a good indication of a possible match within the first 2 articles. The results also show that using more than 2 articles does not degrade performance, although there is a small drop after 10. Therefore 10 articles are used when applying the method for the full set of synsets.

5.3 Stage 3: Refining the mappings

In this section, the methods from Section 3.5 are used to refine the mappings at a global level, eliminating those which are likely to be incorrect.

The best scoring method from the previous section (combined title + text) was applied to all 82115 noun synsets. This is henceforth referred to as the *match* mapping which will be used as a basis for all of the refinement approaches presented here.

The approach described in Section 3.5.1 is used to remove all many-to-1 mappings from the *match* mapping function, creating the $1to1$ function. The results are shown in Table 5.9. For each method, the final column shows if there is a statistically significant improvement over the baseline precision of the *match* method. Note that this takes into account the fact that the refinement methods classify fewer instances as positive (making it harder to show a statistically significant improvement as the sample size is reduced).

The results show the precision rises (from 77.5% to 84.5%) with the expected

Metric	Accuracy	Precision	Recall	F-measure	Significant
match	69.5	77.5	**62.7**	69.3	N/A
$1to1$	68.5	84.5	56.4	67.6	No
link	**70.0**	79.0	**62.7**	**69.9**	No
bilink	66.5	82.9	54.0	65.4	No
$link + 1to1$	69.0	86.6	56.4	68.3	Yes ($p < 0.05$)
$bilink + 1to1$	64.5	**88.2**	47.6	61.9	Yes ($p < 0.02$)

Table 5.9: Refined mappings on $200NS$

lowering of recall (from 62.7% to 56.4%). This confirms the hypothesis, showing that the precision of the one-to-one mappings is greater than for the many-to-1 mappings. Although there is a clear improvement in the precision scores for all refinement methods, only for the $link + 1to1$ and $bilink + 1to1$ methods is this improvement statistically significant. This may be partly due to the fact that since many mappings are removed the refined approaches classify fewer instances as positive, which means it is harder to show a statistically significant improvement.

The results for *link* show a clear rise in precision (from 77.5% to 79.0%). Interestingly however there is no drop in recall - indicating that the method successfully removed many incorrect mappings, while preserving all correct ones. For *bilink* the precision rises further, to 82.9%, however now there is a drop in recall, from 62.7% to 54.0%. Again the hypothesis that the links are indicative of good quality mappings is confirmed by the experimental results.

Finally, experiments are performed to combine the approaches. Mappings are only used which have both the 1-to-1 and link properties. This result shows higher precision than either method alone, giving overall the best quality mappings, achieving 88.2% precision for the $bilink + 1to1$ mapping using bilinks and 1-to-1 constraints.

5.4 Comparison with previous work

As mentioned in Section 2.3.3, recently work has been published which is similar to the work described in this thesis (Ponzetto and Navigli, 2010). Here a comparison is made between the methods and results presented in this thesis and their work.

5.4.1 Methodological differences

One key difference is the direction of the mapping. In Ponzetto and Navigli (2010) the aim is to find for each article the best matching WordNet word sense. This gives a search space of approximately 80,000 noun synsets to search, which is far fewer than the 3 million or so articles in Wikipedia. However the approach used in Ponzetto and Navigli (2010) is even more restrictive - only synsets which contain the same word as the Wikipedia page title are considered.

Their method therefore makes two assumptions:

1. Only WordNet synsets which contain the title word of the Wikipedia page will be a good match - it does not consider any synsets which do not contain that word, but may still overlap in meaning. Therefore mappings between synsets and articles which do not share the title word are missed, resulting in false negatives. Table 5.10 shows examples where the information retrieval approach has found the correct candidate articles for the synset, which the approach of Ponzetto and Navigli (2010) cannot find.

2. If there is one or more synsets with the title amongst its words, then one of those synsets will be a good match - it never decides to make a null match, resulting in false positives. Examples are shown in Figure 5.1, where sports arenas with the name coliseum are linked to the synset.

Wikipedia article	WordNet synset
Livestock carrier	cattleship, cattle boat
Finger cot	thumbstall
pulmonary alveolus	alveolus, air sac, air cell
Benefit performance	benefit

Table 5.10: Mappings which cannot be detected by Ponzetto and Navigli (2010) due to the article title not being found in the synset.

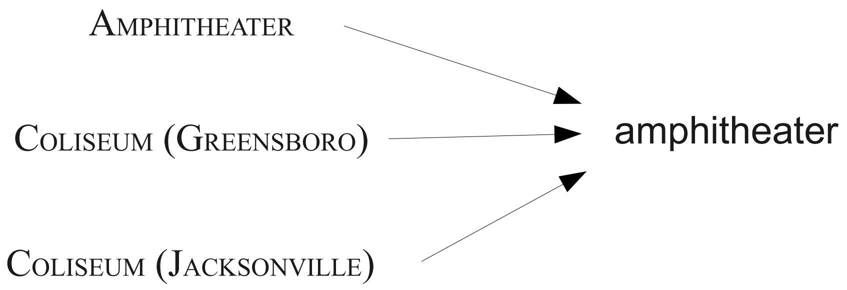


Figure 5.1: Many articles linking to a single synset

5.4.2 Comparison of results

The rest of this section gives an evaluation of the mappings of Ponzetto and Navigli (2010) against the 200NS set for comparison. This is the result of applying their approach to find the best matching synset for articles in Wikipedia. Their mappings have been made publicly available online[1]. As discussed earlier, one issue with the mappings of Ponzetto and Navigli (2010) is that many articles may map to a single synset - the converse problem of the mappings obtained in this thesis, where many synsets may map to a single article. The method described in Section 3.5.1 addressed this issue by eliminating many-to-one mappings leaving a 1-to-1 mapping between synsets and articles. In this section a different method is used to address this problem; combining the mappings of Ponzetto and Navigli (2010) and the mappings generated here. By only preserving those matches which exist in both, both problems are solved simultaneously - finding the best article for a given synset, and the best synset for a given article. The result is guaranteed to be a 1-to-1 mapping, while hopefully discarding fewer correct mappings than the elimination approach. The results are shown in Table 5.11, together with results from the previous sections for comparison.

Metric	Accuracy	Precision	Recall	F-measure
$match$	69.5	77.5	**62.7**	69.3
$link$	70.0	79.0	**62.7**	69.9
$link + 1to1$	69.0	86.6	56.4	68.3
$bilink$	66.5	82.9	54.0	65.4
$bilink + 1to1$	64.5	88.2	47.6	61.9
$ponzetto$	*(66.5)*	*(65.0)*	*(70.6)*	*(67.7)*
$ponzetto \cap match$	71.5	90.4	59.5	71.8
$ponzetto \cap link$	**72.0**	91.5	59.5	**72.1**
$ponzetto \cap bilink$	67.5	**92.9**	51.6	66.3

Table 5.11: Ponzetto mappings evaluated on $200NS$

The *ponzetto* row gives the results of the Ponzetto mappings against the 200NS

[1] Currently at `http://lcl.uniroma1.it/wordnetplusplus/`

data. These results are not directly comparable to the other methods, since multiple articles are given for a single synset, while the other methods are limited to at most one article for each synset. If any of the articles linked to the synset is correct then this is considered a correct mapping. This means recall is artificially high, since the approach has more 'chances' to find the right article.

The last three rows show the result of combining the *ponzetto* mappings with the *match*, *link* and *bilink* approaches respectively using the intersection of the mappings. These show that the precision achieved is higher than using the $1to1$ mappings, while recall remains similar. The highest F-measure and accuracy are achieved using the $ponzetto \cap link$ approach. However comparing $ponzetto \cap link$ to $link+1to1$ and $ponzetto \cap bilink$ to $bilink+1to1$ respectively shows the improvements are not statistically significant.

5.5 Independent evaluation

In addition to the evaluations on the $200NS$ set the candidate articles and mappings were also evaluated on the $1000A$ gold-standard data. This is a slightly modified version of the gold standard data independently produced in Ponzetto and Navigli (2010). The modification process and evaluation methodology was described in Section 4.4.

5.5.1 Candidate articles

The $1000A$ data contains 537 article-synset pairings. For each synset that appears the aim is to find if the associated article occurs within the candidate article sets. As before the $recall_{seq}$ metric is used to determine the recall at n for each synset.

The (A+R)+(L+G) approach was used since this performed best on $200NS$. The results for this are shown in Table 5.12.

Articles	1	2	5	10	20
(A+R)+(L+G)	68.0	81.4	89.6	95.0	97.6

Table 5.12: Recall on $1000A$ combining title matching & IR methods.

Recall at 1-5 articles is lower than when evaluated on the $200NS$ set. However at 20 articles the recall is at the same level. This provides further supporting evidence about the recall quality of the candidate articles.

However as discussed in Section 4.4 the $1000A$ data contains mappings in the opposite direction. So each article is mapped with the best synset - but the converse does not necessarily apply. Therefore the first few candidate articles may in fact be better matches for the synset than the match given in $1000A$ (and this seems to be demonstrated by the $200NS$ evaluations).

5.5.2 Mappings

The method described in Section 4.4.3 is used to evaluate the mappings against the $1000A$ data. The results are shown in Table 5.13. For comparison the evaluation of the mappings made available by Ponzetto and Navigli (2010) are also shown for comparison in the *ponzetto* row. A further experiment combines the mappings of *ponzetto* with the *match* mappings. This is done by taking the union of mapped articles for each synset, resulting the $ponzetto \cup match$ mapping.

The *ponzetto* mappings perform better when evaluated on the $1000A$ data; both precision and recall are higher than any of the mappings of this thesis. As discussed in Section 4.4 the standard of $1000A$ is different, allowing a single synset to map to multiple articles. This puts the mapping methods in this thesis at a disadvantage. The best synset for a given article (as given in $1000A$) may not necessarily mean that this article is the best for that synset, which is what my approaches aim to find.

Metric	Precision	Recall	F-measure	Accuracy
match	52.0	47.0	49.4	70.8
link	51.9	46.6	49.1	70.6
*link*1*to*1	68.1	17.6	28.0	55.8
bilink	52.0	40.7	45.7	68.4
*bilink*1*to*1	68.0	15.2	24.8	55.0
ponzetto	**72.3**	54.1	61.9	72.6
$ponzetto \cup match$	57.1	**73.8**	**64.4**	**81.8**

Table 5.13: Evaluation on 1000A data.

The results show that $ponzetto \cup match$ achieves the highest recall, higher than *ponzetto* alone. This approach also achieves higher accuracy and F-measure, although the precision is significantly lower (since there are more article-synset pairings, some of which are incorrect). The improvements are statistically significant. This shows that *match* contains correct mappings which are not found in *ponzetto*.

5.6 Summary and Discussion

This chapter evaluates the methods presented in Chapter 3. Stage 1 of the mapping process retrieved a set of candidate articles which may be good matches for each noun synset in WordNet. A set of 20 articles was retrieved for each synset using title matching and information retrieval approaches. The best results are obtained using a combination of the two approaches. When evaluated against the 200NS data, a recall of 96% is achieved for matching articles. This represents a great reduction in the search space allowing further methods to select the best articles from the candidate article sets.

Stage 2 determines the best selection from the candidate articles for each synset.

Similarity between features of the synset and the text and title of the article are computed using overlap metrics. Thresholds are estimated from training data to determine whether there should be a mapping or not. Although the title is the most useful single feature, the best results are obtained using both title and text similarity methods. The result is the *match* mapping of approximately 46238 noun synsets in WordNet to a matching article (out of the total of 82115 synsets). On the 200NS data a precision of 75.3% and a recall of 61.2% is achieved.

In Stage 3 a series of refinement methods are applied to the mappings to select more precise sets. This uses a global approach. Firstly many-to-1 mappings are eliminated, creating the $1to1$ mapping. Then, Wikipedia links are used as evidence to select high quality mappings, creating the *link* and *bilink* mappings. The highest precision is obtained by combining refinements to give the $bilink + 1to1$ mapping. This has a precision of 87.8% and a recall of 46.9% on the 200NS evaluation. Although all refinement methods show improvement over the baseline *match* method, only the combined methods $link+1to1$ and $bilink+1to1$ show statistically significant improvements.

Comparisons were also made with previously published work. The publicly available mappings produced by Ponzetto and Navigli (2010) were evaluated against the 200NS data set. The precision achieved by the methods presented here are significantly better than the *ponzetto* mappings. Combining the *ponzetto* mappings with *link* and *bilink* mappings produced improved precision performance over the $link - 1to1$ and $bilink - 1to1$ mappings respectively, although this not statistically significant. However both approaches were significantly better in terms of precision than the baseline *match* approach.

The mappings were also evaluated against the gold standard data of Ponzetto and Navigli (2010). Performance is lower than the *ponzetto* mappings since the methods described here a limited to at most one article match per sysnet. However results confirm that the methods generate correct mappings that are not found in *ponzetto* confirming that the different approach provides new information which complements this previous work.

Finally the methods were applied to the full set of 82115 noun synsets in WordNet to give a complete set of mappings to Wikipedia articles.

Chapter 6

Enriching WordNet

The previous chapters described a process for mapping WordNet synsets to Wikipedia articles, and the process was shown to generate a high quality set of mappings, as evaluated against gold standard data. This can be considered as an *intrinsic* evaluation, measuring accuracy of the mappings against manually annotated data.

This chapter now presents an *extrinsic* evaluation, testing if the mappings provide benefit for an external task. This is done by using Wikipedia links between the mapped articles to enrich WordNet with new relations. This enriched WordNet is then tested to see if it proves more useful for an independent external task. The evaluation task used here is Word Sense Disambiguation.

Section 6.1 describes the results of applying the methods described in the previous chapters to generate a full set of mappings for all the noun synsets in WordNet. **Section 6.2** describes how relations are derived using Wikipedia links between these mapped articles. **Section 6.3** evaluates the new relations on the Word Sense Disambiguation task, using the relations to enrich the existing WordNet knowledge base. Finally **Section 6.4** gives a summary of the chapter.

6.1 Generating complete mappings

A complete mapping was generated for all 82115 noun synsets in WordNet to Wikipedia articles. This was created using the best performing approaches for candidate retrieval and mapping scoring (see Sections 5.1 and 5.2). The mappings were then refined using the methods in Section 5.3. The number of mappings generated using each method is shown in Table 6.1.

Test set	Positive matches
match	38249
$1to1$	29730
link	36677
bilink	30430
$link + 1to1$	28449
$bilink + 1to1$	23393

Table 6.1: Number of mappings

All refinement approaches reduce the size of the mapping. The *link* refinement has the least reduction, with just over a 4% reduction in the number of mappings. The evaluation results suggest that this is an effective way to increase precision without lowering recall, by eliminating incorrect matches while preserving correct ones. In contrast the most refined mapping ($bilink + 1to1$) is just over 60% of the size of the original *match* mapping.

6.2 Deriving New Relations from Wikipedia Links

The generated mappings were used to enrich WordNet with new relations. New relations between WordNet synsets were added using the hyperlink structure in

Wikipedia. If two synsets, a and b, are mapped onto Wikipedia articles, a' and b', and there is a hyperlink connecting a' and b' in Wikipedia then a relation between a and b is added to WordNet. For example, consider the synset-article matches shown in Figure 6.1. If all links are used, then new relations would be added from the Internal control synset to the accountancy synset, and from the accountancy synset to the count synset. However if only bidirectional links are used, then only the relation from accountancy synset to the count synset would be added, since the link from Internal control to Accountancy is not reciprocated.

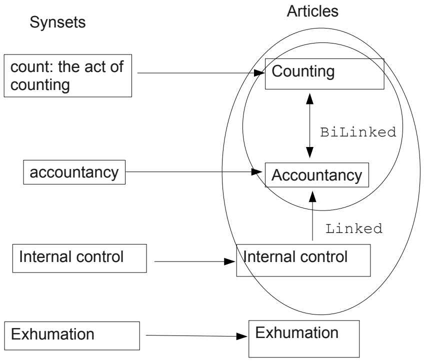


Figure 6.1: Links between articles

This method of deriving new relations was applied to the mappings generated from the previous chapter. Table 6.2 shows the number of relations that were derived using this method for each mapping. For comparison the number of mappings in the mapping from Ponzetto and Navigli (2010) are also shown. The table also shows the overlap with existing relations that were already present in WordNet.

Table 6.2 shows that the majority of the relations derived using these method are novel. As would be expected, the number of generated relations falls using the more refined mappings. With the bi-directional refinement there is a slightly larger

Mapping	Total	Existing	Ratio (%)
link	1,351,106	16,249	1.22
link + 1*to*1	566,166	11,008	1.94
bilink	285,852	10,135	3.55
bilink + 1*to*1	140,801	7115	5.05
ponzetto	1,159,538	30,405	2.62

Table 6.2: Number of relations generated from each mapping, with proportion that were already in WordNet.

overlap with WordNet compared to using directional links.

Table 6.3 shows some examples of novel relations found in the *bilink* + 1*to*1 set.

6.3 Word Sense Disambiguation

Once WordNet is enriched with new relations from the mappings, the next step is to test if the new relations have an impact on WSD performance.

6.3.1 Approach

As described in Section 2.1.4, recently systems have used graph-based methods over knowledge sources for the task of word sense disambiguation. This has the advantage of being unsupervised - i.e. no hand-labelled training data is required. This type of approach provides an ideal test task for knowledge bases since there is no dependence on any other type of data, and thus the effectiveness of the knowledge base can be tested in an unbiased way.

The UKB system (Agirre and Soroa, 2009) is used as the WSD system. This represents a lexical knowledge base, such as WordNet, as a graph. This graph is created by representing each synset as a vertex and adding edges between them if they

Synset 1	Synset 2	Note
tennis	racket	The classic motivating example is found amongst the relations
family therapy	countertransference	Family therapy is a form of psychotherapy, while countertransference is a psychoanalytic process
opium	Afghanistan	Afghanistan is one of the largest producers of opium
tricyclic	narcolepsy	Tricyclics used as treatment for some kinds of narcolepsy
palatine raphe	sublingual gland	Two parts of the mouth in close proximity to each other
New Orleans	African-American	New Orleans is home to one of the largest African-American communities in the USA
ibuprofen	headache	Ibuprofen is one of the most common forms of treatment for headaches
hospital	health insurance	Possibly reflecting the American bias in Wikipedia, where health insurance is required by many for hospital treatment
al-Qaeda	Taliban	Another highly topical relation useful for news analysis

Table 6.3: Novel synset relations found in $bilink + 1to1$

are related in WordNet. Both the relations already existing in WordNet (hypernyms, meronyms etc.) and those that can be derived from the disambiguated glosses can be used to add edges to the graph. To enrich WordNet with the relations derived from Wikipedia new edges are simply added to the graph. The UKB system applies the Personalized PageRank to rank the vertices and thus perform disambiguation. A description of this approach was given in Section 2.1.4. The use of PageRank in this approach was inspired by its successful application in the Google search engine (Page et al., 1999). PageRank identifies the most 'important' nodes in a graph by counting the links into that node from other nodes. Additionally, links from other important nodes are counted more highly than less important nodes. PageRank is therefore a global measure applied over the whole network, not just one local node area. On the web the reasoning is that highly reputable and well known websites will receive links from other reputable sites, while obscure sites will not.

Applying the same ideas to lexical knowledge bases such as WordNet usually required modifying the traditional PageRank algorithm. Highly linked nodes in WordNet may just be common words, which may in fact be just the opposite of what sense disambiguation requires, a specific concept that fits into the given context. In the case of Agirre and Soroa (2009) the PageRank algorithm is adapted into the *Personalized PageRank* (*ppr*) algorithm, which is given the content words of the context as input, over which the initial weightings are applied. The disambiguation then chooses the sense which has the highest weighting in the presence of the input context. A further refinement ensures that different senses of the target word do not reinforce each other. This involves creating a separate graph for each target word, which is built around the other words in the context, but not over the senses of the target word itself. This refined algorithm is termed *ppr_w2w*.

The UKB system takes as input a knowledge base comprising concepts and relations between the concepts. Practically the knowledge base is defined by two parts. The first is a dictionary, mapping each lemma to a set of concepts which contain that word. The second part encodes the relations. Each relation is defined as a connection between two concept nodes in the KB.

So for instance the dictionary will contain lines like the following:

```
cartwheel 02531639-n 00308383-n 11072773-n 01525369-v
```

This identifies that the lemma `cartwheel` is found in the listed concepts.

The relations are then listed as connections between pairs of nodes; a source vertex `u` and a target vertex `v`. So for example as follows:

```
u:00001740-n v:00018241-n
u:00001740-n v:03714099-n
u:00004753-n v:00018241-n
```

Relations can optionally be denoted as directional, and given weights.

6.3.2 Knowledge Bases

The following knowledge bases are used as input for the algorithm. First is the baseline Wordnet 3.0. This encodes all synsets in WordNet 3.0 as concept nodes in the graph, and all relations between synsets as found within WordNet (hypernyms, meronyms etc.)

Then new relations derived from the mappings are added into this baseline knowledge base. Each of the mappings described in Table 6.2 is used. This gives the following knowledge bases which are added onto the baseline WordNet 3.0 knowledge base : Links, BiLinks, Links+1to1, and finally BiLinks+1to1. These use the mappings as described in Chapter 5. For comparison experiments are also performed using relations derived from the mappings from Ponzetto and Navigli (2010). These have been made publicly available by the authors. Using relations derived from these mappings gives the Ponzetto knowledge base.

Another knowledge base for performance is obtained by using extra information which has been present in recent versions of WordNet. This is contained in the disambiguated glosses of each synset. If a word is referred to in the gloss of a synset then it is considered that there is a relation between those two terms. So for example the noun synset Offertory - the part of the Eucharist when bread and wine are offered to God is related to the synsets containing the content terms in the gloss - Eucharist, bread, wine, and God. Adding in these gloss relations gives the WordNet 3.0 + Gloss knowledge base. This can be considered to be an upper bound for performance, since the glosses have been manually disambiguated and tailored to precisely describe the synset. Each of the new relation sets is also added to the WordNet 3.0 + Gloss knowledge base for evaluation, to see if any improvement can be achieved.

6.3.3 Semcor 3.0 Evaluation

The Semcor corpus (Miller et al., 1993) is a sense-tagged corpus created at Princeton University. It is a subset of the Brown corpus and comprises 352 texts containing 360,000 words. It is the largest publicly available sense-tagged corpus.

The original Brown corpus comprised 500 texts of 2000+ words each. A wide range of styles and varieties of prose are present. The types of prose included 'informative prose' such as news, books and scientific journals, as well as 'imaginative prose' such as fiction and humor.

Results

The UKB system was run over Semcor 3.0. The *ppr_w2w* algorithm was used since this was found to be more accurate in previous work (Agirre and Soroa, 2009). The results with the WordNet baseline are shown in Table 6.4.

	Precision	Recall	F-measure
WordNet 3.0	54.9	52.2	53.5
+Links	54.0	51.3	52.6
+BiLinks	**55.8**	**53.0**	**54.3**
+Links+1to1	53.4	50.7	52.0
+BiLinks+1to1	54.4	51.7	53.1
+Ponzetto	54.1	51.4	52.7
WordNet 3.0 + Gloss	**61.3**	**61.1**	**61.2**
+Links	59.3	58.7	59.0
+BiLinks	60.7	60.5	60.6
+Links+1to1	60.3	60.0	60.2
+BiLinks+1to1	61.1	60.4	60.7
+Ponzetto	59.3	59.1	59.2

Table 6.4: WSD accuracy on Semcor 3.0 using ppr_w2w. The WordNet 3.0 + gloss result can be considered an upper bound for performance.

The BiLinks relations improve performance both for recall and precision (this is a statistically significant improvement, using binomial test $p < 0.01$). The other relations appears to be detrimental to performance. These results indicate that the BiLinks mapping approach adds useful relations between WordNet synsets which have a real impact for an extrinsic task. There may be several reasons for the performance gain over the ponzetto results. One possibility is that the direction of the mapping matters; finding the best article for each synset produces better relations than mapping in the opposite direction. As discused in Section 5.4, in Ponzetto and Navigli (2010) many false articles may be associated with synsets due to their mapping method. Experiments using the WordNet + gloss baseline are also shown in Table 6.4. As expected the manually disambiguated glosses are superior to the automatically created relations and provide the upper bound for performance. In all cases adding the automatic mappings give poorer results than using the gloss alone.

Analysis

The BiLinks relations improve upon the baseline WordNet score. Further analysis (Table 6.5) showed that the improvement was focussed on nouns although verbs and adjectives also had a small improvement. This is expected, since only noun synsets are mapped to Wikipedia articles and thus only noun-to-noun relations are added to the knowledge base.

To analyse which nouns were being more accurately disambiguated all instances were collated where the BiLinks knowledge base disambiguated correctly, and the baseline WordNet knowledge base disambiguated wrongly. The most frequent instances are shown in Table 6.6.

Interestingly the word ‘person’ is by far the most common word which is correctly identified by the new knowledge base. In total 7066 instances were correctly identified

	Nouns (86899)			Verbs (47532)		
	Precision	Recall	F-measure	Precision	Recall	F-measure
WordNet 3.0	58.2	58.0	58.1	40.7	40.3	40.5
+Links	56.8	56.6	56.7	40.4	40.0	40.2
+BiLinks	**59.8**	**59.6**	**59.7**	**41.1**	**40.7**	**40.9**
+Links+1to1	55.5	55.4	55.5	40.3	39.9	40.1
+BiLinks+1to1	57.4	57.2	57.3	40.6	40.2	40.4
+Ponzetto	57.4	57.2	57.3	39.6	39.2	39.4
	Adjectives (31551)			Adverbs (10480)		
	Precision	Recall	F-measure	Precision	Recall	F-measure
WordNet 3.0	65.4	64.9	**65.2**	59.2	33.4	42.7
+Links	64.6	64.1	64.4	**59.4**	33.4	**42.8**
+BiLinks	65.4	**65.0**	**65.2**	59.2	33.4	42.7
+Links+1to1	65.0	64.5	64.7	59.2	33.4	42.7
+BiLinks+1to1	65.4	64.9	65.1	59.2	33.3	42.7
+Ponzetto	65.0	64.5	64.8	59.3	33.4	42.7

Table 6.5: WSD accuracy for different parts of speech on Semcor 3.0

Word	Frequency
person	2749
boy	53
house	50
location	47
way	47
man	37
information	33
school	32
society	31

Table 6.6: The most frequent nouns where BiLinks found the correct sense and the baseline WordNet did not.

by BiLinks but not by WordNet. Therefore the word ‘person’, with 6375 occurrences, accounts for over a third of the entire performance gain. The word person has 3 senses in WordNet:

1. person, individual, someone, somebody, mortal, soul (a human being; “there was too much for one person to do”)
2. person – (a human body (usually including the clothing); ”a weapon was hidden on his person”)
3. person – (a grammatical category used in the classification of pronouns, possessive determiners, and verb forms according to whether they indicate the speaker, the addressee, or a third party; ”stop talking about yourself in the third person”)

In all of the error cases the WordNet system had identified the word ‘person’ as the 3rd sense instead of the first.

This analysis suggests that the Wikipedia links in some way enrich the knowledge base with information that helps to correctly identify when persons are described in the first, more concrete sense, rather than the more abstract third sense. A deeper analysis is difficult due to the nature of the PPR algorithm which uses the global structure of the knowledge base which means specific results like this are difficult to track.

6.3.4 Semeval 2007 Evaluation

The *SemEval 2007* coarse grained all words task (Navigli et al., 2007) was also used as an evaluation. This uses a much smaller test set than Semcor and thus it is more difficult to generate statistically significant improvements. However it does provide an interesting further evaluation because of the difference in sense granularity.

The creation of this task was motivated by the recognition that WordNet is a very fine-grained sense inventory over which even humans find difficulty distinguishing similar senses (Kilgarriff, 2001). To address this issue, a coarse-grained version of WordNet was created based on the procedure described in (Navigli, 2006). This involved mapping WordNet senses to top-level entries in the Oxford Dictionary of English (ODE, Soanes et al. (2005)). This was done in two steps: firstly disambiguating the two resources with the SSI algorithm (Navigli and Velardi, 2005) which uses structural pattern matching over term definitions, and secondly using hypernyms and domain labels to find the best mapping. Additionally to the automatic methods, the sense mappings for all words in the test corpus were manually matched by an expert lexicographer. This proved useful in some cases where senses could not be mapped automatically due to entries missing in the ODE, or different spellings and derived forms. For words where even the manual approach could not find an appropriate mapping, the WordNet sense itself was adopted for that word.

The test corpus used for this task consisted of 5377 words of running text from 5 different articles. The first 3 were taken from the *Wall Street Journal Corpus*, the 4th from the Wikipedia entry for computer programming, and the fifth was an excerpt from Amy Steedman's Knights of the Art, consisting of biographies of Italian Painters. Table 6.7 gives summary statistics for each of these documents.

Article	Domain	Words	Annotated
d001	Journalism	951	368
d002	Book Review	987	379
d003	Travel	1311	500
d004	Computer Science	1326	677
d005	Biography	802	345
Total		5377	2269

Table 6.7: The five articles used in Semeval coarse-grained all words task.

In total 2316 content words were found in the documents. However 47 (2%) were excluded because no WordNet sense was deemed appropriate. Only 8 were assigned more than one sense, two coarse senses were assigned to a single word instance, and two distinct fine-grained senses were assigned to 7 word instances. This gave a clear indication that the sense clusters were not ambiguous for the vast majority of words.

For the coarse-grained mapping, a second annotator was employed to independently annotate part of the manual sense mapping (590 word senses). The pairwise agreement with the other annotator was found to be 86.4%.

Likewise for the sense annotation of the test corpus, a second annotator was employed to annotate part of the corpus (710 word instances). The pairwise agreement was found to be 93.8%.

Experiments are carried out using the 1108 noun instances in this data set.

The accuracy of the WSD system is computed as the percentage of tokens that are correctly disambiguated. A baseline for this evaluation data is computed using the most frequent sense in WordNet. Since this classifies all instances, precision = recall = accuracy = F1 measure.

Results

The UKB system by Agirre was run over the noun instances in the Semeval 2007 coarse grained all-words task. Again, the more accurate *ppr_w2w* algorithm was used.

Results of the WSD evaluation using these enriched knowledge bases are shown in Table 6.8. The core knowledge bases are respectively WordNet 3.0 and WordNet 3.0 + Gloss relations.

Knowledge Base	WordNet 3.0	WordNet 3.0 + Gloss
-	**77.9**	84.0
+Links	73.5	80.7
+BiLinks	77.3	83.7
+Links+1to1	74.8	82.4
+BiLinks+1to1	77.4	**84.3**
+Ponzetto	74.4	79.7

Table 6.8: WSD accuracy on SemEval 2007 coarse grained all words task.

Adding the Wikipedia-derived relations proves detrimental to performance over the WordNet baseline. The upper bound using WordNet 3 with the gloss relations gives a score close to state of the art performance on this task. Again, adding the Wikipedia-derived relations is in most cases detrimental to performance. Only in the most refined case, using bi-directional links and 1-to-1 mappings does the score improve on this baseline, although this improvement is not statistically significant (using a binomial test).

Although no improvement is shown over the baseline, a pattern emerges again suggesting that adding fewer but more precise relations to the knowledge base is more effective than adding many relations which are less likely to be correct. Using the Links scores as a baseline, the BiLinks scores are significantly better (binomial distribution, $p < 0.02$), while adding far fewer relations (approx. 140,000 compared to approx. 1.35 million).

For comparison, the results from state of the art approaches from the Semeval 2007 task are shown in Table 6.9. These are for noun instances only, and for all systems the most frequent sense (MFS) backoff is used when no sense is assigned by the system. The systems are: the best performing unsupervised system in SemEval 2007 (Koeling and McCarthy, 2007), the best supervised system (Chan et al., 2007), and a knowledge-rich system (Navigli and Velardi, 2005) which participated outside the competition. Additionally the result obtained by Ponzetto and Navigli (2010) is shown, which uses a degree-centrality graph-based algorithm using WordNet and Wikipedia relations in the knowledge base. For information the performance using most frequent sense alone is also shown. These results show that performance achieved here is close to the state of the art.

6.4 Summary

The methods from the previous chapters were used to generate mappings from WordNet to Wikipedia. These range from the basic *match* mapping which maps approximately 46000 of the 82000 synsets to an article, to the most refined mapping $match_{bilink+1to1}$ mapping of 24000 synsets with associated articles. Wikipedia links between the mapped articles were then used to add relations between the

Method	Accuracy
WordNet 3.0 + Gloss + BiLinks+1to1	84.3
(Koeling and McCarthy, 2007)	81.1
(Chan et al., 2007)	82.3
(Navigli and Velardi, 2005)	84.1
(Ponzetto and Navigli, 2010)	**85.5**
MFS	77.4

Table 6.9: Comparison with state of the art.

corresponding WordNet synsets. As would be expected the more refined sets generated far fewer relations.

These new relations were evaluated on a Word Sense Disambiguation task. The relations were used as the knowledge base for the UKB system from (Agirre and Soroa, 2009). The system was then evaluated over the Semcor 3.0 corpus. Results show that the Links and BiLinks relations improve performance over using WordNet relations alone. These relations also outperform those published by Ponzetto and Navigli (2010), showing that despite the similarity of the work, the approach presented here creates more accurate and useful mappings. There is a pattern suggesting that adding fewer, but more precise relations to the knowledge base gives better results than adding greater numbers of less precise relations. This pattern is confirmed by the additional evaluation over Semeval 2007.

Chapter 7

Summary and Conclusions

This final chapter presents a summary of the thesis (Section 7.1), the conclusions that can be drawn (Section 7.2) and ideas for future work (Section 7.3).

7.1 Thesis summary

The aim of this thesis was to conduct an investigation of different approaches for enriching WordNet using information derived from Wikipedia. This was tested by mapping WordNet synsets to Wikipedia articles, and deriving new relations based on Wikipedia links. The results generated at intermediate stages provide useful data which it is hoped will provide the basis for future research in language processing and knowledge-based tasks.

A set of WordNet noun synsets has been manually annotated with associated Wikipedia articles. This gives an analysis of the overlap between noun synsets and Wikipedia articles, with over 60% of the synsets having a good matching article. This annotated data set is provided online[1], and it is hoped that this data could be reused in further research on work on mapping between the two resources. Some

[1] http://staffwww.dcs.shef.ac.uk/people/S.Fernando/

processing was performed on third-party gold standard data (Ponzetto and Navigli, 2010) to create independent evaluation data for the methods. In this gold standard data there were many articles mapped to multiple synsets (50), whereas far fewer synsets mapped to multiple articles (3). This fact lends support for the decision to map from synsets to articles rather than vice versa.

Automatic methods were presented for mapping WordNet synsets to articles. This comprised a three stage approach. In the first stage, candidate article retrieval, an estimated recall of 96% was achieved for the top 20 articles. This reduces the search space dramatically for future methods, which could explore different methods of searching and scoring the candidate articles to improve mapping performance.

In the second stage, synsets are mapped to single articles using text similarity metrics to compare features between the synset and the article. The best performing method achieves precision of 67.5% and recall of 61.1%.

In the third stage, the mappings from the previous stage are refined by selecting a set of more precise matches using a global approach. This eliminates many-to-1 links and uses Wikipedia links as evidence for good matches. The most refined mappings have a precision of 87.8% and recall of 46.9%.

The mappings generated using these methods are shown to be of higher precision than those generated in previous work such as Ponzetto and Navigli (2010). However combining mappings with those of Ponzetto and Navigli (2010) produce the most accurate mappings, with precision of 93.6% and recall of 46.0%.

Given each of the mappings, Wikipedia links were used to identify new relations with which to enrich WordNet. The relations were derived between synsets where Wikipedia links exist between the corresponding mapped articles. The enriched WordNet was then used as input knowledge for a WSD system. When evaluated over the Semcor 3.0 corpus the new relations significantly improved performance, especially when using the bi-directional link refinement. Using fewer links derived from the more precise sets of mappings were shown to give better performance than using large numbers of less accurate links.

7.2 Conclusions

The annotation work shows that most synsets have a good matching article. This shows there is potential to enrich WordNet by finding the best article match for these synsets. Further observations from the annotation process gives an insight into the overall coverage and nature of WordNet and Wikipedia. WordNet is shown to have a certain proportion of synsets which are very specific and/or obscure, and thus not suitable candidates for encyclopedic articles (e.g. DUMPINESS, see Section 4.1). WordNet also contains certain synsets which are closely related to Wikipedia article concepts, but not deserving of a separate article themselves (e.g. BATH POWDER).

The evaluations show that the candidate articles are very likely to contain the correct matching article. In contrast to previous methods (Ruiz-Casado et al., 2005; Ponzetto and Navigli, 2010) which rely on the article title matching one of the synset words, the approach here uses an IR engine to make full use of the article text. This approach identifies several candidate matches which cannot be obtained by previously proposed methods.

The matching methods proposed here use a simple text overlap metric to measure the similarities between synsets and articles. Further experiments show that high precision is possible for a small set of synsets using a global refinement approach, making use of Wikipedia links.

The improvement in the WSD score over the Semcor 3.0 corpus establishes that the new relations offer valuable information over and above the baseline WordNet relations. The relations produced in this thesis give a far bigger boost to performance than those produced in Ponzetto and Navigli (2010). This confirms that despite the similarity of the approaches the methods proposed in this thesis offer superior performance.

7.3 Future work

7.3.1 Addressing limitations of WordNet

In Chapter 1, it was noted that one shortcoming of WordNet is the lack of topical relations between synsets, the so-called ‘tennis problem’. There have been other issues noted about WordNet, which are describe below. It is useful to identify these, as they can inform directions for future work on utilising Wikipedia to improve or enrich WordNet.

One observation is that WordNet senses are too fine-grained, sometimes difficult even for humans to distinguish (Navigli, 2006). One possible approach to addressing this problem may be to group together synsets which are found to map to the same article in Wikipedia using the approaches described in this thesis. For example two synsets containing the word *constable* are quite similar:

1. A lawman with less authority and jurisdiction than a sheriff.
2. A police officer of the lowest rank.

If these map to the same article in Wikipedia, then this could be used to group these senses into one cluster. This is similar to the approach described in Navigli (2006) where WordNet senses are mapped to coarse senses in the Oxford English Dictionary. However Wikipedia may prove to be a more up-to-date and comprehensive resource for this purpose.

Another issue is that it is difficult for WordNet to keep up with new words which enter into common usage. There has been previous work on adding new words into WordNet (Ciaramita and Johnson, 2003; Curran, 2005; Pantel, 2005).

These new words can be broadly divided into two categories: new words for existing concepts (e.g. the word ‘feds’ referring to police officers, which was widely used by rioters in the recent UK disturbances), and entirely new concepts (e.g. tweet for Twitter posts). There are several ways in which information can be used from Wikipedia to find new words to add into WordNet. For the first category (new words for existing concepts), the redirect system in Wikipedia may be used to add new synonyms for existing synsets. This would consider all words and phrases which redirect to the mapped article as synonyms. This kind of approach has been used to find synonyms in specialised domains such as place names (Overell and Rüger, 2007) and agricultural terms (Milne et al., 2007). An example for place names would be synonyms for London: {‘London, UK’, ‘Londinium’}. For the second category (entirely new concepts), new synsets would have to be created, a process which may be have to be done manually. However Wikipedia links could then inform how the new synsets then link appropriately to existing ones. Many new instances and specialised concepts would be added to WordNet and linked appropriately to existing WordNet synsets, for example many specific instances of films, books etc. This would also help address other shortcomings of WordNet - the lack of coverage of specialised domains, and specific instances, such as people and places. This approach has been explored by aligning synsets with Wikipedia categories (Suchanek et al., 2008), but not so far with articles.

7.3.2 Other work

There are strong arguments to be made justifying the effort to map WordNet to Wikipedia. This thesis has covered only one application, adding topical relations to WordNet and using this on a WSD task. However there are many other possible directions of future work making use of this mapping. For example future work could integration WordNet and Wikipedia in different languages. This could be potentially very useful for machine translation and cross-language information retrieval. Such ideas may provide enough incentive to encourage a manual effort to map WordNet to Wikipedia. The most realistic way of achieving this may be a collaborative open source approach, or using crowd-sourcing such as Mechanical Turk.

If the manual mapping were not possible, then there is scope for improvement in automatic methods. It may be possible to achieve both high recall and precision with the mappings, perhaps by making more use of information in Wikipedia, such as the categories and Wikipedia links.

Bibliography

E. Agirre and A. Soroa. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics, 2009.

E. Agirre, O. Ansa, D. Martinez, and E. Hovy. Enriching WordNet Concepts with Topic Signatures. In *Proceedings of the North American Chapter of the Association for Computational Linguistics workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburg, USA, 2001.

R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern Information Retrieval*, volume 463. ACM press New York, 1999.

A. Bagga, J.Y. Chai, and A.W. Biermann. The role of WordNet in the creation of a trainable message understanding system. In *Proceedings of the National Conference on Artificial Intelligence*, pages 941–948. John Wiley & Sons Ltd, 1997.

S. Banerjee and T. Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, 2003.

R. Benassi, S. Bergamaschi, and M. Vincini. Web Semantic Search with TUCUXI.

In *Proceedings of the Twelfth Italian Symposium on Advanced Database Systems SEBD*, pages 426–433, 2004.

B. Boguraev and T. Briscoe. Large Lexicons for Natural Language Processing: Utilising the Grammar Coding System of LDOCE. *Computational Linguistics*, 13(3-4):203–218, 1987.

A. Budanitsky and G. Hirst. Evaluating Wordnet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

R. Bunescu and M. Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of European Chapter of the Association for Computational Linguistics*, volume 6, 2006.

A. Burgun and O. Bodenreider. Comparing terms, concepts and semantic classes in wordnet and the unified medical language system. In *Proceeding of thes North American Chapter of the Association for Computational Linguistics Workshop, 'WordNet and Other Lexical Resources: Applications, Extensions and Customizations'*, pages 77–82, 2001.

G. Carenini, Raymond T. Ng, and X. Zhou. Summarizing Emails with Conversational Cohesion and Subjectivity. In *Proceedings of the Association for Computational Linguistics 2008: Human Language Technology*, pages 353–361, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Y.S. Chan, H.T. Ng, and Z. Zhong. NUS-PT: Exploiting parallel texts for Word Sense Disambiguation in the English all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 253–256. Association for Computational Linguistics, 2007.

R. Chapman. *Roget's International Thesaurus (Fifth Edition).* Harper-Collins, New York, 1992.

K.W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

Massimiliano Ciaramita and Mark Johnson. Supersense tagging of unknown nouns in WordNet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 168–175, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proceedings of the 13th international conference on World Wide Web*, pages 462–471. ACM, 2004.

P. Clough and M. Stevenson. Cross-language information retrieval using EuroWordNet and word sense disambiguation. *Advances in information retrieval*, pages 327–337, 2004.

James R. Curran. Supersense tagging of unknown nouns using semantic similarity. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 26–33, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

H. Fang. A Re-Examination of Query Expansion using Lexical Resources. *Proceedings of the Association for Computational Linguistics 2008*, pages 139–147, 2008.

C. Fellbaum, editor. *WordNet: An electronic lexical database.* MIT Press, 1998.

C. Fellbaum, M. Palmer, H.T. Dang, L. Delfs, and S. Wolf. Manual and automatic semantic annotation with WordNet. In *Proceedings of the North American Chapter* *of the Association of Computation Linguistics Workshop on WordNet and Other Lexical Resources: Applications, Customizations*, 2001.

Samuel Fernando and Mark Stevenson. Aligning Wordnet Synsets and Wikipedia Articles. In *Proceedings of the AAAI Workshop on Collaboratively-built Knowledge Sources and Artificial Intelligence*, 2010.

Samuel Fernando and Mark Stevenson. Mapping wordnet synsets to wikipedia articles. In *Proceedings of the Eight International Conference on Language Resources and Evaluation, LREC*, volume 12, pages 590–596, 2012.

J. Firth. *A synopsis of linguistic theory 1930-1955.* Studies in Linguistic Analysis, Philological. Longman, 1957.

S. Flank. A layered approach to NLP-based information retrieval. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 397–403. Association for Computational Linguistics, 1998.

J. Giles. Special Report: Internet Encyclopedias Go Head to Head. *Nature*, 438(15): 900–901, 2005.

T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

RV Guha, D.B. Lenat, et al. Building large knowledge based systems. *Representation and Inference in the Cyc Project. Massachusetts*, 1990.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: An update. *Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining Explorations Newsletter*, 11(1):10–18, 2009.

S. Harabagiu, G.A. Miller, and D. Moldovan. WordNet 2-A Morphologically and Semantically Enhanced Resource. *Proceedings of the Association for Computational Linguistics Special Interest Group on the Lexicon*, 99:1–8, 1999.

S.M. Harabagiu and D. Moldovan. An intelligent system for question answering. In *Proceedings of the 5th International Conference on Intelligent Systems*, pages 71–75. Citeseer, 1996.

M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

A. Hliaoutakis, G. Varelas, E. Voutsakis, E.G.M. Petrakis, and E. Milios. Information retrieval by semantic similarity. *International Journal on Semantic Web and Information Systems*, 2(3):55–73, 2006. ISSN 1552-6283.

Hongyan, J. Applying WordNet to natural language generation. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, 1998.

M. Jarmasz and S. Szpakowicz. Roget's Thesaurus as an Electronic Lexical Knowledge Base. Technical report, School of Information Technology and Engineering, University of Ottawa, 2000.

J. J. Jiang and D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *Proceedings of the International Conference on Research in Computational Linguistics*, 1997.

A. Kilgarriff. English lexical sample task description. In *Proceedings of the Association for Computational Linguistics Special Interest Group on the Lexicon SENSEVAL workshop*, 2001.

A. Kilgarriff. Senseval: An exercise in evaluating word sense disambiguation programs. In *Proceedings from First International Conference on Language Resources and Evaluation pp. 581*, volume 588. Citeseer, 1998.

D. Klein and C.D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, page 3. MIT Press, 2003.

R. Koeling and D. McCarthy. Sussx: WSD using automatically acquired predominant senses. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 314–317. Association for Computational Linguistics, 2007.

O.Y. Kwong. Aligning wordnet with additional lexical resources. In *Proceedings of the Workshop on Usage of WordNet in Natural Language Processing Systems*, 1998.

C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. In Christine Fellbaum, editor, *WordNet – An Electronic Lexical Database*. MIT Press, 1998.

D. B. Lenat. CYC: a large-scale investment in knowledge infrastructure. *Communications of the Association for Computing Machinery*, 38(11):33–38, 1995.

M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. Association for Computing Machinery, 1986.

Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. *Institute of Electrical and Electronics*

Engineers Transactions on Knowledge and Data Engineering, pages 1138–1150, 2006.

D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

K. Litkowski. Towards a Meaning-Full Comparison of Lexical Resources. In *Proceedings of the Association for Computational Linguistics Special Interest Group on the Lexicon Workshop on Standardizing Lexical Resources*, pages 30–37, College Park, MD, USA, 1999.

C.D. Manning and H. Schütze. *Foundations of statistical natural language processing*, volume 59. MIT Press, 1999.

A.T. McCray, A.M. Razi, A.K. Bangalore, A.C. Browne, and P.Z. Stavri. The UMLS Knowledge Source Server: a versatile Internet-based research tool. In *Proceedings of the American Medical Informatics Association Annual Fall Symposium*, page 164. American Medical Informatics Association, 1996.

O. Medelyan and C. Legg. Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense. In *Proceedings of the WIKI-AI: Wikipedia and AI Workshop at the AAAI*, volume 8, 2008.

O. Medelyan and D. Milne. Augmenting domain-specific thesauri with knowledge from Wikipedia. In *Proceedings of the New Zealand Computer Science Research Student Conference, Christchurch, New Zealand*, 2008.

O. Medelyan, D. Milne, C. Legg, and I.H. Witten. Mining meaning from Wikipedia. *International Journal of Human-Computer Studies*, 2009.

R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In *Proceedings of North American Chapter of the Association for Computational Lingustics: Human Language Technology*, pages 196–203, 2007.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993.

D.N. Milne, I.H. Witten, and D.M. Nichols. A Knowledge-Based Search Engine powered by Wikipedia. In *Proceedings of the sixteenth Association for Computing Machinery conference on information and knowledge management*, pages 445–454. ACM, 2007.

D. I. Moldovan and R. Mihalcea. Using WordNet and lexical operators to improve Internet searches. *Internet Computing, Institute of Electrical and Electronics Engineers*, 4(1):34–43, 2000.

R. Navigli. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 105–112. Association for Computational Linguistics, 2006.

R. Navigli. Word sense disambiguation: A survey. *Association for Computing Machinery Computing Surveys*, 41(2):10, 2009.

R. Navigli and M. Lapata. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20^{th} International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1683–1688, Hyderabad, India, 2007.

R. Navigli and P. Velardi. Structural semantic interconnections: a knowledge-based approach to Word Sense Disambiguation. *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, pages 1075–1086, 2005.

R. Navigli, K.C. Litkowski, and O. Hargraves. Semeval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 30–35. Association for Computational Linguistics, 2007.

D.L. Olson and D. Delen. *Advanced data mining techniques.* Springer Verlag, 2008.

I. Ounis, C. Lioma, C. Macdonald, and V. Plachouras. Research directions in terrier. *Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper*, 2007.

S.E. Overell and S. Rüger. Geographic co-occurrence as a tool for GIR. In *Proceedings of the 4th Association for Computing Machinery workshop on Geographical information retrieval*, pages 71–76. Association for Computing Machinery, 2007.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab, 1999.

P. Pantel. Inducing Ontological Co-occurrence Vectors. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 125–132, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

Siddharth Patwardhan and Ted Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, volume 1501, pages 1–8, 2006.

W. Peters, M.T. Sagri, D. Tiscornia, and S. Castagnoli. The LOIS Project. In *Proceedings of Linguistic Resources Evaluation Conference*, Genoa, Italy, 2006.

S. P. Ponzetto and R. Navigli. Knowledge-rich Word Sense Disambiguation rivaling supervised system. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pages 1522–1531, 2010.

S.P. Ponzetto and R. Navigli. Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *Proceedings of the 21st international joint conference on Artifical intelligence*, pages 2083–2088. Morgan Kaufmann Publishers Inc., 2009.

S.P. Ponzetto and M. Strube. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1440. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

S.S. Pradhan, E. Loper, D. Dligach, and M. Palmer. SemEval-2007 task 17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92. Association for Computational Linguistics, 2007.

P. Procter, editor. *Longman Dictionary of Contemporary English.* Longman Group Ltd., Essex, UK, 1978.

N. Reiter, M. Hartung, and A. Frank. A resource-poor approach for linking ontology classes to Wikipedia articles. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 381–387. Association for Computational Linguistics, 2008.

P. Resnik. Using Information Content to Evaluate Semantic Similarity in a

Taxonomy. In *International Joint Conference on Artifical Intelligence*, pages 448–453, 1995.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets. *Advances in Web Intelligence: Third International Atlantic Web Intelligence Conference, Atlantic Web Intelligence Conference 2005, Lodz, Poland, June 6-9, 2005: Proceedings*, 2005.

M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. *Data & Knowledge Engineering*, 61(3):484–499, 2007.

R. Sinha and R. Mihalcea. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the International Conference on Semantic Computing. 2007*, pages 363–369. IEEE, 2007.

R. Snow, D. Jurafsky, and A. Y. Ng. Learning Syntactic Patterns for Automatic Hypernym Discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA, 2005.

C. Soanes, A. Stevenson, J. Pearsall, and P. Hanks. *Oxford dictionary of English*. Oxford University Press Toronto, Ontario, Canada, 2005.

K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. *Elsevier Journal of Web Semantics*, 2008.

F.M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

G. Upton and I. Cook. *Understanding Statistics*. OUP Oxford, 1997.

J. Veronis and N.M. Ide. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pages 389–394. Association for Computational Linguistics, 1990.

E.M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69. Springer-Verlag New York, Inc., 1994.

P. Vossen. EuroWordNet: a multilingual database with lexical semantic networks. *Computational Linguistics*, 25(4), 1998.

D. Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 197–204, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

F. M. Zanzotto and A. Moschitti. Automatic Learning of Textual Entailments with Cross-Pair Similarities. In *Proceedings of the 21st International Conference* *on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 401–408, Sydney, Australia, July 2006. Association for Computational Linguistics.