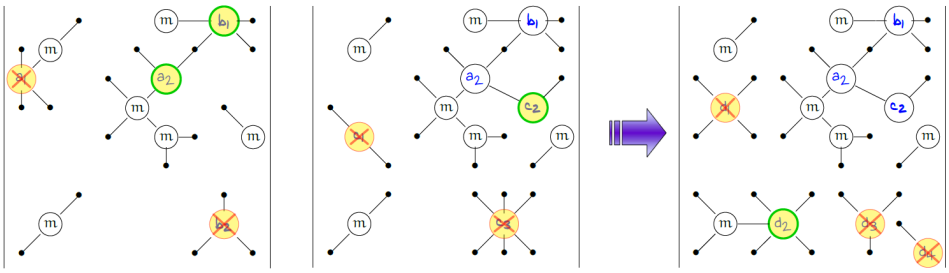
UNSUPERVISED KNOWLEDGE-BASED WORD SENSE DISAMBIGUATION: EXPLORATION & EVALUATION OF SEMANTIC SUBGRAPHS

STEVE LAWRENCE MANION



DOCTORAL THESIS IN COMPUTATIONAL & APPLIED MATHEMATICS

August 2014

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Department of Mathematics & Statistics
College of Engineering

Steve Lawrence Manion: *Unsupervised Knowledge-based Word Sense Disambiguation: Exploration & Evaluation of Semantic Subgraphs*, Doctoral Thesis in Computational & Applied Mathematics, August 2014

NOTE TO THE READER

This thesis is a collection of works in the area of Word Sense Disambiguation, specifically focused on the exploitation of semantic subgraphs. It is structured into three parts. Part I includes a literature review split into two chapters, as well as a chapter outlining the scope of the research. Part II includes four chapters, each describing milestones made throughout the research in chronological order. Each of these chapters include a methodology, results, and related work where appropriate. Part III concludes the contributions of the thesis, through discussion of results and outlook towards future work. Finally there is the appendix which contains papers published by the author, references, and other relevant material.

Often the chapters in Part II refer back to what has been formalised in the literature review of Part I, therefore how to navigate between various sections of the thesis will be explained here in advance. The example below illustrates the referencing system employed.

Section 1.2.3.4 := <Chapter 1>.<Section 2>.<Subsection 3>.<Subsubsection 4>

Note that the first digit is very important, as it refers to the chapter number. If reading this thesis as a PDF file, these references are hyperlinks that can simply be clicked on to navigate to each mentioned section. This is also the case for citations. Acronyms are re-introduced in every chapter, alleviating the burden of memorising them all. Where appropriate, *parts* and *chapters* of the thesis have *italicised* introductions. This weaves content of the thesis together to help the reader see the bigger picture.

ABSTRACT

Hypothetically, if you were told: "Apple uses the apple as its logo". You would immediately detect two different senses of the word "apple", these being the *company* and the *fruit* respectively. Making this distinction is the formidable challenge of Word Sense Disambiguation (WSD), which is the subtask of many Natural Language Processing (NLP) applications. This thesis is a multi-branched investigation into WSD, that explores and evaluates unsupervised knowledge-based methods that exploit semantic subgraphs. The nature of research covered by this thesis can be broken down to:

1. Mining data from the encyclopedic resource Wikipedia, to visually prove the existence of context embedded in semantic subgraphs
2. Achieving disambiguation in order to merge concepts that originate from heterogeneous semantic graphs
3. Participation in international evaluations of WSD across a range of languages
4. Treating WSD as a classification task, that can be optimised through the iterative construction of semantic subgraphs

The contributions of each chapter are ranged, but can be summarised by what has been produced, learnt, and raised throughout the thesis. Furthermore an API and several resources have been developed as a by-product of this research, all of which can be accessed by visiting the author's home page at `http://www.stevemanion.com`. This should enable researchers to replicate the results achieved in this thesis and build on them if they wish.

PUBLICATIONS, PRESENTATIONS, & POSTERS

Several ideas and figures in this thesis have appeared in the following peer reviewed publications.

Published Papers

Steve L. Manion and Raazesh Sainudiin. An Iterative 'Sudoku Style' Approach to Subgraph-based Word Sense Disambiguation (2014). *In Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (*SEM'14)*, pages 40–50, Dublin, Ireland. ACL.

Steve L. Manion and Raazesh Sainudiin (2013). DAEBAK!: Peripheral Diversity for Multilingual Word Sense Disambiguation. *In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13).*, pages 250-254, Atlanta, Georgia. ACL.

Alyona Medelyan, Steve L. Manion, Jeen Broekstra, Anna Divoli, Anna-lan Huang, and Ian H. Witten (2013). Constructing a Focused Taxonomy from a Document Collection. *In Proceedings of the 10th Extended Semantic Web Conference (ESWC'13)*, pages 367-381, Montpellier, France. Springer, Heidelberg.

Presentations & Posters

Presentation for paper (Manion and Sainudiin, 2014) at the 3rd Joint Conference on Lexical and Computational Semantics (*SEM), Dublin, Ireland (August, 2014)[1]

Presentation for Chapters 4 and 6 content at the New Zealand Mathematics and Statistics Postgraduate Conference (NZMASP), CASS Station, New Zealand (November, 2013)[2].

Presentation for Chapters 4 and 6 content as a *primer* to the Mathematics & Statistics Department at Canterbury University, New Zealand (May, 2013).

Presentation and poster session for paper (Manion and Sainudiin, 2013) at SemEval, Altanta, Georgia (June, 2013)[3]

Poster session for Chapter 4 content at the Machine Learning Summer School, Bordeaux, France (September, 2011)[4].

Presentation for Chapter 4 content at Nuffield College, Oxford University (June, 2011).

Attended

Attended New Zealand Mathematics Research Institute Inc. (NZMRI) Summer Workshop: Geometric Mechanics and Shape (January, 2013)[5] and NZMRI Summer Workshop: Random Media and Random Walks (January, 2012)[6].

---

1 http://www.cs.york.ac.uk/semeval-2013 - SemEval 2013 Homepage
2 http://www.math.canterbury.ac.nz/nzmasp2013/ - NZMASP 2013 Homepage
3 http://www.cs.york.ac.uk/semeval-2013 - SemEval 2013 Homepage
4 http://mlss11.bordeaux.inria.fr - MLSS 2011 Homepage
5 http://seat.massey.ac.nz/NZMRI13/index.html - NZMRI 2013 Workshop Homepage
6 https://www.stat.auckland.ac.nz/~mholmes/workshop/Nelson2012.html - NZMRI 2012 Workshop Homepage

Why not?

— Jung A (Amy) Kwon

ACKNOWLEDGEMENTS

Thank you to my parents, who have done their best to provide a stable environment for me to facilitate good research. I would also like to thank the rest of my greater family and close friends for supporting me in my endeavours.

I would like to thank Dr Raazesh Sainudiin, as a supervisor you have provided unconditional support for the direction of my research and a very unruly student.

Thank you to the Korean Foundation[7], who have supported me financially over the last 2 years of my PhD. I was able to purchase the hardware I needed to perform my experiments, I was able to attend the conferences I presented my publications at, and I was able to spend a duration of my PhD in Korea to improve my Korean language abilities at Sungkyunkwan University. I will always feel bonded to Korea, and naturally make a solemn effort apply my research to benefit the Korean language.

Lastly, the person quoted at the top of this page, you showed me that life is best lived by exploring all consequences, good or bad. Your uncompromising nature of doing this day in and day out, will always inspire me to dive head first into the unknown and try something new. You live life the best way possible, never change!

7 KF Graduate Studies Fellowship - http://koreanstudiesaa.wordpress.com/scholarships

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AGROVOC *Agr*icultural *Voc*abulary *Thesaurus*

AAAI Association for the Advancement of Artificial Intelligence

ACL Association for Computational Linguistics

ACM Association for Computing Machinery

AI Artificial Intelligence

API Application Programming Interface

ARPA Advanced Research Projects Agency

BC Betweenness Centrality

BFS Breadth First Search

CL Computational Linguistics

CS Cosine Similarity

DFS Depth First Search

EACL European Chapter of the Association for Computational Linguistics

ELDA Evaluations and Language resources Distribution Agency

ELRA European Language Resources Association

ESWC European Semantic Web Conference

FS First Sense

F-STEP Focused SKOS Taxonomy Extraction Process

HF-IPF Hyperlink Frequency - Inverse Page Frequency

HITS Hypertext Induced Topic Selection

HTML Hyper Text Markup Language

IR Information Retrieval

IAA Inter-Annotator Agreement

IEEE Institute of Electrical and Electronics Engineers

ISCA International Speech Communication Association

ISDN Integrated Services Digital Network

ITA Inter-Tagger Agreement

KF Korean Foundation

KSAA Korean Studies Association of Australasia

LCS Longest Common Subsequence

LD Levenshtein Distance

LDC Linguistic Data Consortium

LKB Lexical Knowledge Base

LREC Language Resources and Evaluation Conference

MFS Machine Frequency Sense

MRD Machine Readable Dictionary

MSI *New Zealand's* Ministry of Science & Innovation

ML Machine Learning

MLSS Machine Learning Summer School

MT Machine Translation

NAACL North American Chapter of the Association for Computational Linguistics

NLP Natural Language Processing

NZMASP New Zealand Mathematics and Statistics Postgraduate *Conference*

NZPSV New Zealand Public Service Vocabulary

PD Peripheral Diversity

PEE Pingar Entity Extract

POS Part-of-Speech

PSD Pairwise Semantic Dissimilarity

RDF Resource Description Framework

RDFS Resource Description Framework Schema

SKOS Simple Knowledge Organisation System

SPARQL SPARQL Protocol and RDF Query Language - *Recursive Acronym*

SSI Sorensen Similarity Index

TF-IDF Term Frequency - Inverse Document Frequency

URI Uniform Resource Indicator

URL Uniform Resource Locator

WSD Word Sense Disambiguation

WSJ Wall Street Journal

*SEM Joint Conference on Lexical and Computational Semantics

XML Extended Mark-up Language

Part I

BACKGROUND, FOREGROUND & FOCUS

Part I of this thesis presents the literature review split into Chapter 1 and Chapter 2. The former chapter is to provide the reader with an elementary background on Word Sense Disambiguation (WSD) and is aimed at making the thesis accessible to a wider audience. The latter chapter is more technical and details unsupervised knowledge-based WSD that employs the use of semantic subgraphs. Following the literature review chapters, Chapter 3 outlines the focus of the thesis. Here the literature review is connected to the WSD challenges the author will tackle.

1

THE BACKGROUND: LITERATURE REVIEW

The background of this literature review first formalises the problem of Word Sense Disambiguation (WSD) and details the core obstacles that make it such a formidable challenge. Following this the conventional, explicit, and in-vitro interpretation of a WSD system is formalised. Finally WSD approaches, applications, and resources are briefly elaborated on for later mention in the thesis. The purpose of this background chapter is to make the thesis more accessible, by introducing the elementary concepts and terminology of WSD.

1.1 AN INTRODUCTORY EXAMPLE OF WSD

A word's intended sense (*its meaning*), is characterised by the context it is used in. A definitive example is a *homograph*[1]: a class of words that map to several etymologically[2] unrelated senses. For instance "bank", disambiguated in the text below:

"...fishing on the river bank"

$$\text{BANK}_{\text{(n) } land} := \textit{The sloping edge of land by a river.}$$

It is used as a *noun* with adjacent *content*[3] words "fishing" and "river". Just as the human lexicon exploits these *lexical* features of context to interpret the sense of "bank" as $\text{BANK}_{\text{(n) } land}$ rather than $\text{BANK}_{\text{(n) } finance}$, so can a machine with access to a Lexical Knowledge Base (LKB).

The Lesk (1986) algorithm makes use of one the most established LKBs, the Machine Readable Dictionary (MRD), by exploiting the textual overlap of the word "river" appearing in both the *context* and *definition* of the above example. Furthermore, instinctive lexicon exploits such as *one-sense-per-collocation/discourse* (Yarowsky, 1995) and *one-sense-per-part-of-speech* (Stevenson and Wilks, 2001) are also effective, with respective experiments reporting precision over 96% on a *Lexical Sample* (LS) and over 94% on an *All-Words* (AW) evaluation[4]. There are many more published methods of exploiting the lexicon that achieve high precision at the *homograph* level. However, beyond homographs, WSD requires deeper semantic analysis, ensuring it has remained a formidable challenge since the 1950s[5]. To explain why, the core obstacles WSD must overcome will now be elaborated on.

1 See (Templeton, 2003) and (Nelson, 1976) for a clarification of homonymy.
2 A word's etymology is a chronical account of its origin and derivation.
3 Content, or open-class words, include nouns, verbs, adjectives and adverbs. Conversely, functional, or closed-class words, link content words together.
4 For a *lexical sample*, a set of words is selected in which a number of corpus instance are found that represent the word's usage. As for *all words*, this is the disambiguation of all open-class words in text (Kilgarriff and Rosenzweig, 2000).
5 Generally speaking, the origins of WSD took shape as a subtask of Machine Translation (MT), see (Hutchins, 1995)

1.2 THE CORE OBSTACLES OF WSD

1.2.1 *WSD Obstacle #1: Sense Granularity*

Shared etymologies between word senses makes establishing distinctions between them an intractable challenge for lexicographers[6]. In the words of Wilks and Stevenson (1996) they tackle it as either "*lumpers* or *splitters*: those who like to divide senses without apparent end, or those who prefer larger (more *homographic*) clusters of usages". Whether lexicographers prefer to lump or split, they must strive to do so in a consistent manner, classifying words as:

a) *Monosemous* - Having *one* intended sense, regardless of context.

b) *Homonymous* - Having *many* senses, all with *unique* etymologies.

c) *Polysemous* - Having *many* senses, sharing *common* etymologies.

This classification task presents the obstacle of *sense granularity*. At first glance a word can appear to fit neatly into one of these classifications; however on closer inspection it may have the attributes of another depending on how coarsely or finely distinctions are made by the lexicographer.

For instance, it is easy to understand the shared etymology between the senses LINE(n) *fishing* and LINE(n) *power*, since they are both a *length of fibre*. While they are unique senses, their shared etymology could warrant them being *coarsely lumped* together. Counterwise as Hanks (2000) points out, even the homograph "bank" ↦ BANK(n) *institution* can be *finely split* into the senses BANK(n) *institution* or BANK(n) *building*. This obstacle of establishing the ideal level of sense resolution, must not only be overcome by lexicographers but also by WSD systems.

The implications of sense granularity for a WSD task is comparable to image granularity for an image recognition task. In Figure 1, does a system

6 A lexicographer is a someone who edits and compiles dictionaries, effectively formalising the human lexicon in written form.

need to identify a *fish* (left image) or go as far as identifying a *Tarpon fish* (right image) at the risk of a misidentification? Likewise, to what extent does a WSD system need to articulate a word's intended sense to achieve its Natural Language Processing (NLP) task?

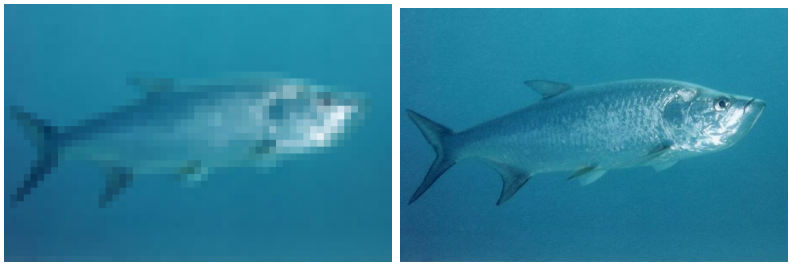


Figure 1: Visual Analogy of Sense Granularity

Ide and Wilks (2007) assert that coarse-grain (homographic) level WSD is sufficient for most NLP tasks. While many publications support this, it does not negate the importance of supplying a WSD system with *knowledge* that is capable of providing finer granularity if required. In this respect, knowledge is classified as either:

a) *Structured* - e.g. MRDs, ontologies, theasauri

b) *Unstructured* - e.g. Mono/Bi-lingual corpora, word frequencies

For structured knowledge, the level of sense granularity is decided by the lexicographers who design them. International WSD evaluations have demonstrated that the use of coarser lexicons results in higher reported precision (Kilgarriff, 2001), furthermore WSD evaluated at the coarse-grained level also achieves higher precision (Navigli et al., 2007). Conversely, for unstructured knowledge the level of sense granularity is decided by the knowledge resource itself. For example cross-lingual approaches to WSD that rely on bilingual corpora[7] can dynamically deal with granularity; since finer sense distinctions are only relevant as far as they are lexicalised in different translations of words (Lefever and Hoste, 2013).

7 Bilingual corpora are translated texts (such that corpora is the plural of corpus).

1.2.2 *WSD Obstacle #2: Under-specified Context*

The human need to communicate new ideas has ensured that language continuously evolves. New words are created while existing ones are infused with new senses, often established through metaphoric usage (Hirst, 1987, p5-6). By analysing word frequencies in running text, a power-law distribution can be observed known as Zipf's Law (Zipf, 1949) in Equation (1):

$$P_n \sim \frac{1}{n^a} \tag{1}$$

In running text the probability of a word in the human lexicon being the next word shares an inverse relationship with its frequency rank n, such that $a > 1$ but very close to 1 in value. In his work Zipf (1945) noticed that *more* frequent words tend to be *more* polysemous. The implication of this for WSD, is that even though most words in the lexicon are monosemous (Leacock et al., 1993, p260), a majority of words in running text are not! He explained his law in terms of *economies of effort*: the *speaker's* effort to produce *contextualised* speech finding an equilibrium with the auditor's effort to *disambiguate* what is heard. In order for communicative efficiency, humans tend to make *just* enough effort to contextualise their speech, only drawing on more infrequent monosemous words when necessary to alleviate the disambiguation effort required by the auditor. It is only when a speaker misjudges the effort of contextualisation required for the auditor, that presents the WSD obstacle of context *under-specification*.

Under-specification is a combination of a word's context not being *abundant* or *salient* enough. When Weaver (1949, p20-21) first wrote about WSD in his discussion of meaning and context, he likened under-specification to be the same as reading text through an opaque mask with a window slit. Hence the context *abundancy* is the number of words either side of the ambiguous word that are visible, better known as a word's *local context* or *context window size*. On the other hand, context *saliency* refers to how unique

the lexical, morphological, and other such features present in the context window are.

WSD systems often base context window size on defined segments of text, such as a sentence. If the sentence is very short, there is a risk there will be a lack of context abundancy. For example the sentence may rely on a *cohesion device* (Halliday and Hasan, 1976) that establishes a relationship to words in previous or following sentences. Take for example the cohesion device known as *ellipsis*[8], in which B's reply in the text below can only be understood by having access to A's preceding statement:

A:"I like the blue {hat}."
B:"I prefer the green {}."

The WSD system would struggle since B neglects to make Zipf's stated *effort* to mention the word "hat" in {}. If the context window included a sentence on either side, the crucial word "hat" would be available as input to the WSD system. Early work by Kaplan (1950) and later more extensive experiments by Yarowsky (1993), have demonstrated that the local context is remarkably small with the two content words on either side of the ambiguous word proving to be adequate in most cases. If the context is abundant enough, perhaps it lacks saliency. In Figure 2 the image can easily be confused as either a *skull* or a *lady*[9]. While there is an abundance of detail in this image that can be interpreted as context, these details are not very salient and conflict with each other making it difficult to arrive at a final disambiguation of what the eye can see. The contextual features that can lack in saliency are:

Figure 2: All Is Vanity

8 Nunan (1993) defines ellipsis as occurring when some essential structural element is omitted from a sentence or clause and can only be recovered by referring to an element in preceding text.

9 This is a double image (or visual pun) painted by Charles Allan Gilbert in 1892, an early contributor to camouflage art for the U.S. Shipping Board in WWI.

a) *Morphological/Categorical*: This type of ambiguity occurs for words like "key", which have many senses in more than one of the syntactic categories such as *nouns*, *verbs*, or *adjectives* (e.g. KEY$_{(n)\ music}$, KEY$_{(v)\ to\ set}$, or KEY$_{(a)\ to\ be\ important}$). In the experiments conducted by Wilks and Stevenson (1996) that exploited parts-of-speech in WSD at the homograph level, when there was more than one sense per syntactic category, this was precisely what induced an upper bound in performance. Homographic ambiguity is also very prominent in the Japanese language due to its seemingly *ad hoc* use of Chinese characters (Kanji) (Olinsky and Black, 2000). While words like "key" are prevalent in English, Turdakov (2010) notes that other languages like Russian have a morphology that innately avoids this ambiguity.

b) *Syntactic Ambiguity*: This is the compounded result of several instances of morphological ambiguity, which would confuse a part-of-speech tagger. For example sentences such as: "They're cooking apples."(Hirst, 1987), even with the morpheme /ing/, the word cook-*ing* can still be interpreted as both the *continuous present* form for the verb cook, or alternatively as the *noun phrase* cooking. The collective syntax of the sentence as a whole does not divulge which interpretation is correct. In English, this type of ambiguity is often caused when the copula[10] is dropped, which typically happens in casual speech or in newspaper headlines.

c) *Vagueness/Indeterminacy*: This occurs when the speaker tends to use coarser rather than finer sense granularity in his or her speech. In other words, there is a lack of lexical redundancy built into speech. For example, the word "child" can refer to either a boy or a girl. Which exactly cannot be determined; therefore it is described as indeterminant or vague (Ravin and Leacock, 2000).

d) *Pragmatic/Non-linguistic*: This type of ambiguity occurs when the text relies on non-linguistic cues to fill the information gap. A famous example

10 A copula is a *function* word that links the subject of a sentence with a predicate, i.e. Whales *are* mammals.

often used is "A brick dropped on the table, and it broke." (Norvig, 2007). In this instance it cannot be known from the text alone if the pronoun "it" refers to is the *brick* or the *table*, consequently it cannot be disambiguated which object, if not both, actually broke. This highlights the fact that the WSD system needs to draw on real world, or pragmatic knowledge. Most areas of pragmatics have not received much attention in statistical NLP, both because it is hard to model the complexity of world knowledge with statistical means and due to the lack of training data (Manning and Schutze, 1999).

At any time these features of context can lack enough saliency in the defined context window for disambiguation. It reveals that WSD systems should not rely on humans to make an adequate level of *effort* in building redundancy into their speech/text through specification of context.

1.2.3 *WSD Obstacle #3: Domain Coverage*

The amount of human knowledge harnessed into machine readable resources is improving, yet still remains scarce. This *wait* to harness *all* of human knowledge at the machine level is often cited as the "Knowledge Acquisition Bottleneck" (Gale et al., 1992b). The implications of this *bottleneck* can be understood by trying to guess what the spiral shape (left) is in Figure 3 below.

Figure 3: Visual Analogy of Domain Coverage

If you have a good understanding of ballistics or are an avid James Bond[11] fan, perhaps you immediately noticed that you were *staring down the barrel of a gun*, as the spiral in question depicts the *rifling* of a gun barrel[12]. If you struggled to disambiguate Figure 3, it is because the image stems from outside your *domain of knowledge*. This introduces the obstacle of *domain knowledge coverage* for WSD. Regardless of the LKB the WSD system draws from, if there is no trace of James Bond present in the knowledge, the WSD system will lack adequate *coverage* to produce any sensible disambiguation output.

Bar-Hillel (1960) understood this early on in his critique of Machine Translation (MT) – of which also applies to WSD as an implicit subtask of MT. He argued that fully automated high quality MT would be impossible, since the prerequisite task of mapping human knowledge to a machine readable resource for a MT system to access is also impossible. He reasons this with his example of "The box was in the pen". A machine would require either real world knowledge to gauge the size of boxes and pens, to determine that a typical box could fit inside a play pen, but not a writing pen. Or alternatively, a machine would require real-time non-linguistic knowledge, described by Kilgarriff (2007) as what is being observed or done or encouraged or forbidden at the time (perhaps implying MT would require machine vision amongst other sensory input).

Although Bar-Hillel considered covering the vastness of human knowledge to be a chimerical goal, he stated that a MT system would not only need to be supplied with a MRD, but also an encyclopedia. As it turns out this is exactly where researchers are headed today, in attempts to pool together as much knowledge from as many domains and resource types to solve this obstacle of coverage. Endeavours to improve the semantic web

11 http://knowyourmeme.com/memes/people/james-bond - James Bond image source. This visual context is reminiscent of the classic introduction scene of every James Bond movie. The barrel of a gun is trained on Bond as he walks across the screen until at some point Bond swings around and shoots the supposed holder of the gun.

12 http://www.flickr.com/photos/ranfog/7172461895/in/pool-74682133@N00 - Photo source for the rifling of a gun barrel.

take things even further, by standardising linked data formats and methods of including more semantic content in web structures.

1.2.4 *WSD Obstacle #4: Meaningful Evaluation*

Early evaluations of WSD systems were isolated and difficult to compare up until Kilgarriff (1998) established SensEval, the collaborative forum and framework to evaluate WSD. Part of the success of SensEval is the publishing of sense tagged texts. This allows researchers to evaluate various WSD solutions in identical settings and understand which methods are most effective.

SensEval also adopts the use of *upper* and *lower* bounds of performance, advocated by Gale et al. (1992a). Upper bounds represent a ceiling for Inter Tagger Agreement or Inter Annotator Agreement (ITA/IAA) for the humans producing an answer key for the WSD task, since naturally humans will disagree with each other for a certain percentage of taggings. However as Kilgarriff (2001) notes, replicability of ITA scores defines the true upper bound of a task. That is, the level of agreement between two completely different groups of taggers using the same methodology to tag a particular data set need to be able to produce reasonably similar taggings.

As for the lower bound this is a baseline for WSD systems to try to beat, of which there are four in conventional use. Listed in order of increasing difficulty, these baselines are generated as the:

a) *Random Sense* - A randomly selected sense.

b) *Lesk Sense* - The sense selected using the Lesk (1986) algorithm.

c) *First Sense (FS)* - The first listed sense in a dictionary (or MRD).

d) *Most Frequent Sense (MFS)* - The sense most frequently used for tagging in a sense tagged corpus.

Unsurprisingly, the Random baseline is the easiest to beat. The Lesk (1986) baseline is not much harder, yet it is an important legacy baseline

because it was first to exploit MRDs strictly for the purpose of WSD. As for the FS and MFS baselines, they are comparable in terms of difficulty and are now the baselines used in most evaluations found in modern literature. In line with Zipf's *law of meaning* that infers more frequent senses are represented by more frequent words, the FS or MFS baseline is correct the majority of the time and therefore making it notoriously hard to beat. In fact the FS/MFS baseline, provides a reasonable fail safe option when the context is *under-specified*, described by McCarthy et al. (2004) as *backing-off*. Before and since, it has been made use of in many endeavours such as Wilks and Stevenson (1998); Navigli et al. (2007) and Ponzetto and Navigli (2010). In fact the WSD survey written by Navigli (2009) asserts that virtually all WSD systems refer to some form of back-off strategy due to the phenomenon of data sparseness, which is synonymous to a lack of domain knowledge coverage.

In addition to upper and lower bounds, there are two modes of evaluation for WSD. Categorised by Ide and Veronis (1998) in terms borrowed from biology, these modes are:

a) *In-vitro* - Evaluates WSD *independent* of an application.

b) *In-vivo* - Evaluates WSD *dependent* on its overhead NLP task.

SensEval began with a focus on *in-vitro* WSD, in which sense tagged corpora were prepared for both Lexical Sample (LS) and All Words (AW) tasks in a range of languages[13]. However over time the tasks of SensEval evolved to investigate the WSD *obstacles* discussed in this section, along with WSD evaluated in more *in-vivo* circumstances (this is further discussed in the following section). To reflect the expanding range of semantic tasks, SensEval was retitled to SemEval in 2007.

13 Refer to Table 2 and 3 in Section 1.4.4 at the end of this chapter for more details on these tasks.

1.2.5 *The Core & All Other Obstacles Considered*

All of the core obstacles discussed are inter-related. For example both an under-specified context or a lack of knowledge coverage can render a WSD system incapable of producing output (providing there is no back-off strategy). Again the sense granularity of a WSD system's output may not be appropriate for the NLP task it is being evaluated for. These four obstacles are among a number of other prevailing obstacles that have ensured WSD to be the formidable challenge that it is. In fact, WSD is considered to be an "AI-Complete" problem (Ide and Veronis, 1998; Mallery, 1988), which means WSD is part of a subset of problems that need to be solved in order to achieve Artificial Intelligence (AI) comparable to that of a human. The effects of these four obstacles, among others, will be noticeable throughout the thesis; therefore hopefully this discussion has been helpful to those who have been unfamiliar with Word Sense Disambiguation.

1.3 WSD SYSTEMS

1.3.1 *WSD Applications*

WSD is present in a range of NLP applications, and is often referred to as an *intermediate task*, meaning it is only desired as a means to an end, not as an application in and of itself (Wilks and Stevenson, 1996). For this reason in-vitro WSD can be criticised as having no real world purpose, yet as Navigli (2009, p57) asserts and this author agrees, investigations into in-vitro WSD must go on. Since there are many questions it can answer, that in turn could improve in-vivo WSD. This is reflected in the diversification of SemEval tasks over the years which address new WSD obstacles as others are overcome. More recent tasks are also geared towards more in-vivo evaluation, perhaps because some researchers feel in-vitro WSD has

plateaued (Agirre and Edmonds, 2007) or perhaps there is a growing desire to demonstrate WSD has real world use. As a subtask, WSD can be either:

a) *Implicit* - WSD is an *inseparable* process from its NLP task.

b) *Explicit* - WSD is a *separable* process from its NLP task.

In-vitro WSD is inherently explicit, because it is performed independent of an application. A challenge for researchers is to use in-vitro WSD in in-vivo settings, because unfortunately by design, NLP applications tend not to accommodate WSD being an explicit module that can be added or removed. Instead, WSD is an implicit process of a greater module in the NLP application, and therefore is inseparable. Such an example of implicit in-vivo WSD, is a statistical Machine Translation (MT) system that draws from knowledge that is a set of translated documents. Words and chunks of text are probabilistically mapped from one language to another, such as was performed by Brown et al. (1990).

This makes it difficult for SemEval advances made in explicit in-vitro WSD to be projected onto the in-vivo WSD realm. As relatively recent surveys detail, it has yet to be seen for in-vivo WSD to significantly improve the performance of one of its applications (Agirre and Edmonds, 2007; Navigli, 2009) and for a framework for in-vivo WSD evaluation to be well established (Turdakov, 2010) to the extent of the in-vitro framework established by Kilgarriff (1998) for SensEval. If explicit in-vivo WSD can successively be harnessed, then some NLP applications that it holds promise for are briefly described as follows:

a) *Machine Translation* (MT) was the first notable application that highlighted a need for WSD. Naturally, a MT system needs to understand the correct sense of a word in the source text, and produce target translations with words that map to the original senses as close as possible. As for some examples taken from the literature, to translate the French word "grille" to English, depending on the context, can be translated as *railings*, *gate*, *bar*, *grid*, *scale*, or *schedule* (Ide and Veronis, 1998), likewise

the Italian word "penna" can be translated in English as *feather, pen,* or *author* (Navigli, 2009).

b) *Information Retrieval* (IR) or Hypertext Navigation is another notable application of WSD, that all internet users are familiar with when browsing the web. For example, when searching for a person who shares the same name as someone very famous like "Michael Jackson", the chances of finding them can be potentially troublesome if they have a lack of presence on the internet. However as Turdakov (2010) suggests, this turns out not to be too much of a problem since modern IR systems do not use special WSD algorithms and rely on the assumption that the user introduces additional information to the context that is sufficient to get relevant results. Even more so these days, search engines collect data from their users to build up a pre-defined context for their searches.

c) *Lexicography* is another interesting application of WSD. For example the automated tagging of text can help lexicographers build up a large sense tagged corpus from a collection of documents. This would be a repository of word senses used in authentic ways. While not every tagging would be correct, it would alleviate the burden of trying to locate a sense that is rarely in use. For example locating an example of the word "date" used with the intended sense of DATE$_{(n)}$ *fruit*, rather than DATE$_{(n)}$ *time*.

There are many other NLP applications that could make use of WSD advancements, such as speech and text processing, content and theme analysis, semantic web endeavours, information extraction, and so forth. This highlights that WSD is an intermediate task in a broad range of greater NLP applications, even if only implicitly.

1.3.2 *WSD Approaches*

The previous section covered motivations of *why* WSD should be achieved by reviewing the applications it is used in, whereas this section details

how WSD is achieved by reviewing the core approaches employed. Once again this section is also brief, primarily because it only serves to point out the broader differences and similarities in each approach. Recall from the title of this thesis, the approach this author will investigate is unsupervised knowledge-based WSD, which employs the use of semantic subgraphs. This will be formalised in detail later in Chapter 2.

a) *Knowledge-based WSD* – As the name suggests, this approach exploits LKBs to achieve WSD. The knowledge is structured, drawing from resources such as MRDs or semantic graphs, rather than unstructured corpora. Advancements in the semantic web and organisation of knowledge have seen LKBs take strides in development. This directly benefits knowledge-based WSD and helps alleviate the *knowledge acquisition bottleneck* (Gale et al., 1992b). The key strength of knowledge-based methods for WSD is they are usually applicable to all words in unrestricted text (Mihalcea, 2007) ensuring it is capable of a broader range of tasks than corpora based WSD methods.

b) *Supervised WSD* – This approach is typically achieved with the use of a sense tagged corpus, to which machine learning techniques are applied. This form of WSD tends to achieve the best performance out of all approaches (Màrquez et al., 2007). Yet as Pedersen (2007) rightly states, supervised WSD systems are bound by their training data, and therefore are limited in portability and flexibility in the face of new domains, changing applications, or different languages. Furthermore training data is very expensive to produce and is by no means an easy task even for experienced human taggers. As Palmer et al. (2001) describe, the sense inventories that taggers refer to may have redundancies and gaps, sense descriptors (glosses) can be ambiguous, among many other unforeseen issues.

c) *Unsupervised WSD* – A key advantage of this approach is it does not require the use of sense-tagged corpora that are so scarce and expensive

to produce. Furthermore if untagged corpora are used then sense granularity poses no obstacle, since this is only caused when knowledge is structured and senses are discretised (Pedersen, 2007). Although in early SensEval tasks unsupervised systems under-performed in comparison to their supervised counter-parts (Palmer et al., 2001; Snyder and Palmer, 2004), as of SemEval they have proven to be a robust and competitive alternative, particularly with the use of LKBs (Navigli et al., 2007; Agirre et al., 2010; Navigli et al., 2013).

It is worth pointing out that these approaches are not mutually exclusive. For example an unsupervised approach as seen in (Yarowsky, 1995) is *knowledge-lean* because it relies on *unstructured* untagged corpora. Whereas an unsupervised approach like (Ponzetto and Navigli, 2010) is *knowledge-rich* because it relies on a *structured* LKB. Each WSD system exhibits some degree of both supervision and knowledge-richness (or structuredness). To establish a better understanding of these two dimensions, the explanation and figure produced by Navigli (2009, p15-16) is very helpful.

1.4 WSD RESOURCES

To end this chapter some well known resources used for WSD are briefly introduced, in order of sense inventories, then corpora, and finally sense annotated corpora which are the product of a corpus and sense inventory. However before these introductions, it is important to clarify the relationship between LKBs and sense inventories, and the implications this has for WSD.

1.4.1 *LKB to Sense Inventory*

Representing the sense of a word and the context it is used in is a difficult challenge for WSD. In Table 1, a snapshot of a well known LKB can

be observed, the *dictionary*[14]. It has entries for word senses with their *part-of-speech*, *definition*, and a *context example*. A dictionary's purpose is to be an educational device for humans, facilitating exchange of knowledge and helping humans establish contextual boundaries between word senses. To achieve this, lexicographers must identify a discrete set of senses for each word that warrant being listed for easy comprehension. This makes a dictionary an easy LKB to convert into a *sense inventory*.

Table 1: Example of WSD for Pen and Bank

Sense	Definition	Context Examples
	"PEN"	
Pen[1] (*noun*)	An instrument for writing or drawing with ink, originally consisting of a shaft with a sharpened quill or metal nib, now more widely applied.	This *pen* won't write.
Pen[2] (*noun*)	A small enclosure for cows, sheep, poultry, etc.	The dogs herded the sheep into the *pen*.
	"BANK"	
Bank[1] (*noun*)	A financial establishment that uses money deposited by customers for investment, pays it out when required, makes loans at interest, exchanges currency, etc.	The *bank* lent her money to buy a car.
Bank[2] (*noun*)	The sloping edge of land by a river.	The River Frome had burst its *banks* after the torrential rain.

However, what is a sense inventory and how does it differ from a LKB? Amsler (1984) defines a LKB as being a general repository of any kind of lexical knowledge about concepts and their relationships, that is not intended for any particular application. A sense inventory on the other hand is a collection of senses specifically organised to complement WSD and other NLP applications. Therefore if dictionary's definitions are viewed as discrete senses, it makes for an easy conversion into a sense inventory.

As Agirre and Edmonds (2007) note, a consequence of dictionaries is they lead many to assume that words have a finite and discrete set of senses.

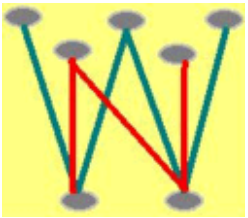
14 The *meanings* are taken from the *New Zealand Oxford Dictionary* (Deverson and Kennedy, 2005), with the *context examples* taken from the *Oxford Collocations Dictionary for Students of English* (Crowther et al., 2002).

While the discretisation of senses helps humans more efficiently pass on knowledge, an information loss occurs through all the minor interpretations that are omitted. As Ide and Veronis (1998) pointed out, dictionaries are created for human use, and not for machine exploitation. While LKBs can be converted to sense inventories to be harnessed by machines, the information loss involved in forming discrete sets of senses will inherently affect the sense granularity, domain coverage, and other such *obstacles* for the WSD system. In fact, some researchers argue against treating senses as discrete altogether (Kilgarriff, 1997; Hanks, 2000) and advocate WSD as an *implicit* subtask.

As previously alluded to, LKBs vary in how easily they are converted into sense inventories. Suitable candidates include thesauri, encyclopedias, ontologies, bilingual corpora, and other LKBs with a little data mining and processing. Certain LKBs are even better because they are designed specifically for NLP tasks, including WSD, and have endpoints and Application Programming Interfaces (APIs) developed for them for easy integration. These include semantic graphs, concordance systems, web dictionaries/encyclopedia made accessible through standardisation of the semantic web, and so forth.

The experiments conducted in this thesis focus on explicit and in-vitro WSD, for input that is linguistic-only. Therefore only sense inventories which by their design have senses discretised will be used throughout this thesis. They will now be described.

1.4.2 *Sense Inventories*



Firstly there are the obvious candidates for sense inventories, those that map the human lexicon such as the Longman Dictionary of Contemporary English (Procter, 1978) or the Oxford Dictionary of English (Deverson and Kennedy, 2005). Also online,

there are collaboratively developed dictionaries such as Urban Dictionary[15] which is constantly updated with the latest colloquial expressions, and Wiktionary for which semantic resources such as DBnary (Sérasset, 2012) have had their development based on. Again there are thesauri such as Roget's Thesaurus (Chapman, 1977) or WordNet (Fellbaum, 1998) which doubles as both a dictionary and thesaurus with its synonym sets (known as synsets). Since 1985 WordNet has been continuously updated at the Princeton University's Cognitive Science Laboratory[16]. It has since then been perhaps the most widely used sense inventory for WSD evaluations.

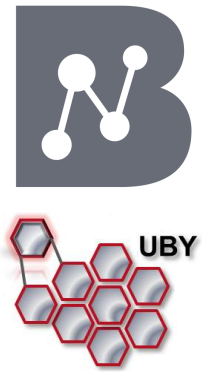
Next there are encyclopedia based sense inventories, perhaps the most well known being Wikipedia, it has been able to demonstrate similar accuracy to Britainica as an encyclopedic source (Giles, 2005). Naturally many researchers try to exploit this multilingual and freely available knowledge, therefore several tools[17] have been developed in order to harness this knowledge. There are no shortage of works that try to explore this ever growing collaborative resource, see (Medelyan et al., 2009) for a comprehensive account of mining Wikipedia.

Freebase[18] is an online collaborative resource much like Wikipedia, except it is more geared towards machine exploitation and link structure, rather than human use and textual content. This is reflected by Freebase's SPARQL endpoint allowing it to be remotely queried, or locally if Freebase is downloaded. Again the semantic links are defined with the semantic web in mind, using RDF (Resource Description Framework) triples, that denote relationships by *subject-predicate-object*. For ex-

15 http://www.urbandictionary.com/ - Urban Dictionary Homepage
16 http://wordnet.princeton.edu - The Princeton University Homepage of WordNet
17 http://www.mediawiki.org/wiki/Alternative_parsers - Alternative Parsers used to Mine Wikipedia.
18 http://www.freebase.com - The Homepage of Freebase

ample, <Napoléon Bonaparte> </people/person/height_meters> <1.68> is a triple that denotes the height of Napoléon Bonaparte at 1.68cm tall.

DBpedia[19] is an effort to mine Wikipedia as a semantic graph, to align it with other linked data resources (e.g. Freebase). It also has a SPARQL endpoint to query, leading it to be a popular means of easily accessing Wikipedia. Works such as DBpedia Spotlight (Mendes et al., 2011) which is a system for automatically annotating text documents with DBpedia URIs is evidence of this.



Over 50 years ago Bar-Hillel (1960) in his critique of MT, and later in his report to the Automatic Language Processing Advisory Committee (ALPAC), brought on starvation of funding to MT in the USA for over a decade (Hutchins, 1995). He described his colleague's suggestion of giving a machine access to both a dictionary and an encyclopedia as utterly chimerical and hardly deserving any further discussion. Yet this is now a reality with sense inventories such as BabelNet (Navigli and Ponzetto, 2012a) and Uby (Gurevych et al., 2012) that map the lexicon WordNet (Fellbaum, 1998), the encyclopedia Wikipedia, along with a collection of other semantic resources all together as one sense inventory.

1.4.3 *Corpora*

The Hector Lexical Database, now known as the British National Corpus[20] is an ongoing project at Oxford University (Atkins, 1992). It was the corpus used for the first SensEval (Kilgarriff and Palmer, 2000) in 1998 at Herstmonceux Castle, Sussex, England.

19 http://dbpedia.org - The DBpedia Homepage (Wiki)
20 http://www.natcorp.ox.ac.uk - The British National Corpus

Then there is the Brown Corpus (Kucera and Francis, 1967) which was originally put together in 1963 to analyse the word frequencies in American English. It contains 500 samples of writing from a range of styles and domains, each is approximately 2000 words in length, making up a total of 1,014,000 running words of text in the corpus.

For a more universal representation of the English language, there is the Internet Corpus (Sharoff, 2006) which is an ongoing project at Leeds University. In fact this is an open-source corpus in a number of languages. Notably, the Cross-Lingual Lexical Substitution task in SemEval 2010 made use of this corpus (Mihalcea et al., 2010).

There is a range of corpora that are available, many of which can be acquired through the Linguistic Data Consortium (LDC)[21] or the Evaluation and Language resources Distribution Agency (ELDA)[22].

1.4.4 *Sense Tagged Corpora*

Sense tagged corpora are produced either manually for precision or automatically for speed and quantity. It is the process of tagging each word in running text with a sense taken from a common sense inventory. Over the next three pages are lists of notable sense tagged corpora in English, most of which are the by-product of a SensEval or SemEval evaluation tasks[23].

The sense tagged corpora listed in Table 2 are Lexical Samples based, whereas those listed Table 3 are All Words based. Table 4 also lists All Words based sense tagged corpora, however the corpora contain extra complexity because they are each created for the purpose of investigating a particular WSD obstacle, such as those discussed earlier in Section 1.2.

21 https://www.ldc.upenn.edu - Linguistic Data Consortium Homepage
22 http://www.elda.org - Evaluation & Language resources Distribution Agency Homepage
23 Note that for sense tagged corpora which are the product of an ongoing project, the details listed in the following tables are likely to change over time.

Table 2: Notable Lexical-Sample Sense Tagged Corpora (in English)

Sense Annotated Corpora	Texts Annotated	Sense Inventories Used	Lemmas	Instances
Line-hard-serve Leacock et al. (1993)	WSJ, American Printing House for the Blind, & the San Jose Mercury	WordNet 1.5	3	4000 per lemma
Interest Bruce and Wiebe (1994)	WSJ (ACL/DCI Treebank)	LDOCE (6 senses)	1	2,369 total
SensEval 1 LS English Task Kilgarriff and Rosenzweig (2000)	Hector Corpus	Hector Dictionary	35	160-400 per lemma
SensEval 2 LS English Task Kilgarriff (2001); Palmer et al. (2001)	Penn Treebank II WSJ (BNC as supplementary)	WordNet 1.7	73	7567 total (noun + adj)
SensEval 3 LS English Task Mihalcea et al. (2004)	BNC	WordNet 1.7.1, & Wordsmyth	57	3,944 total
SemEval 1 LS WSD of Prepositions Litkowski et al. (2007)	FrameNet (Mostly BNC)	Oxford Dictionary	34	25,000+ total

Table 3: Notable All-Words Sense Tagged Corpora (in English)

Sense Annotated Corpora	Texts Annotated	Sense Inventories Used	Instances
Hector Atkins (1992)	BNC (pilot version)	Hector Dictionary	≈ 200,000
SemCor (Semantic Concordance) Miller et al. (1993)	Portion of Brown Corpus	WordNet 1.6	234,136
DSO (Defence Science Organisation) Ng and Lee (1996)	Brown Corpus & WSJ (28.6% / 71.4%)	WordNet 1.6	≈ 192,800
SensEval 2 English AW Task Palmer et al. (2001)	WSJ (x3 Articles)	WordNet 1.7 (pre-release version)	2473
Open Mind Word Expert Chklovski and Mihalcea (2002)	Penn Treebank, LA Times, +others	WordNet 1.7	≈ 70,000
SensEval 3 English AW Task Snyder and Palmer (2004)	WSJ & Brown Corpus (x2 Articles / x1 Excerpt)	WordNet 1.7.1	2,081
BabelCor Navigli and Ponzetto (2012a)	SemCor & Wikipedia Texts (2.3% / 97.7%)	BabelNet 1.0.1	1,000,000+ (>3×330,993)

Table 4: Notable All-Words Sense Tagged Corpora (for the Evaluation of a Specific WSD Obstacle)

Sense Annotated Corpora	Texts Annotated	Sense Inventories Used	Instances
SemEval 1/Task 7 - Coarse-grained WSD Navigli et al. (2007)	WSJ, Wikipedia, & Book Excerpt (-"Knights of the Art")	WordNet 2.1	2,269
SemEval 2/Task 3 - Cross-Lingual WSD Lefever and Hoste (2010)	EuroParl Parallel Corpus	Deduced from corpus translations, see task details	Lexical Sample task (20 nouns, 50 instances)
SemEval 2/Task 17 - WSD on a Specific Domain Agirre et al. (2010)	European Center for Nature Conservation & World Wildlife Forum Texts	Publicly available 'wordnets' for each language, see task details	5342 in English (nouns & verbs only)
SemEval 4/Task 12 - Multilingual WSD Navigli et al. (2013)	Statistical Machine Translation Workshop Texts (2010-2012)	BabelNet 1.1.1, WordNet 3.0, & Wikipedia	1,931 in English (nouns only)

2

THE FOREGROUND: LITERATURE REVIEW

Previously in Chapter 1, Word Sense Disambiguation (WSD) was introduced along with its core obstacles to overcome, as well as its applications, approaches, and resources. Now the basics of WSD have been covered, this chapter details the mode of WSD to be investigated and improved. That is – unsupervised knowledge-based WSD, which makes use of semantic subgraphs and is subjected to in-vitro evaluation.

The formalisations established in this chapter represent the author's own interpretation of this WSD mode, as well as a consolidation of key advancements made in the literature unified under a common notation. All of which are drawn from in successive chapters, using this chapter as a reference.

2.1 SUBGRAPH-BASED WSD

Essential to this mode of WSD is the semantic subgraph, which encapsulates the contextual usage of a set of words. It contains candidate sense nodes for each ambiguous word, and edges that represent the semantic relationships between them. WSD can be achieved based on the assumption that – *the most central sense nodes in the semantic subgraph best reflect the intended sense for each ambiguous word*. The resources and methodology for this mode of WSD will now be formalised.

2.1.1 *Requirements of Sense Inventory $\mathcal{G}$*

Firstly, to construct a semantic *subgraph* a WSD system needs to draw from a *supergraph*. Let this be $\mathcal{G}$, which is simply a Lexical Knowledge Base (LKB) that can be represented as a graph-based sense inventory. Regardless of the LKB that $\mathcal{G}$ is based on, an intrinsic requirement for WSD is the mapping of words and senses. Therefore $\mathcal{G}$ would at least need to contain a *bipartite subgraph*, $\mathcal{G}_\beta$, of word and sense nodes such as illustrated by Figure 4.

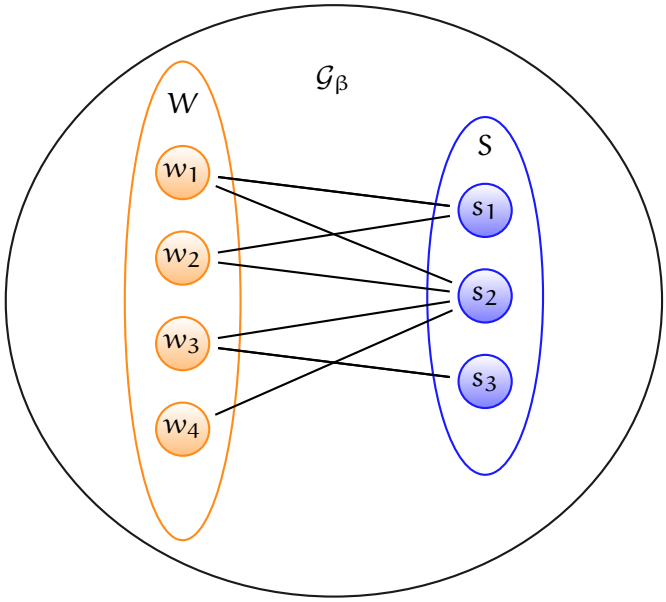
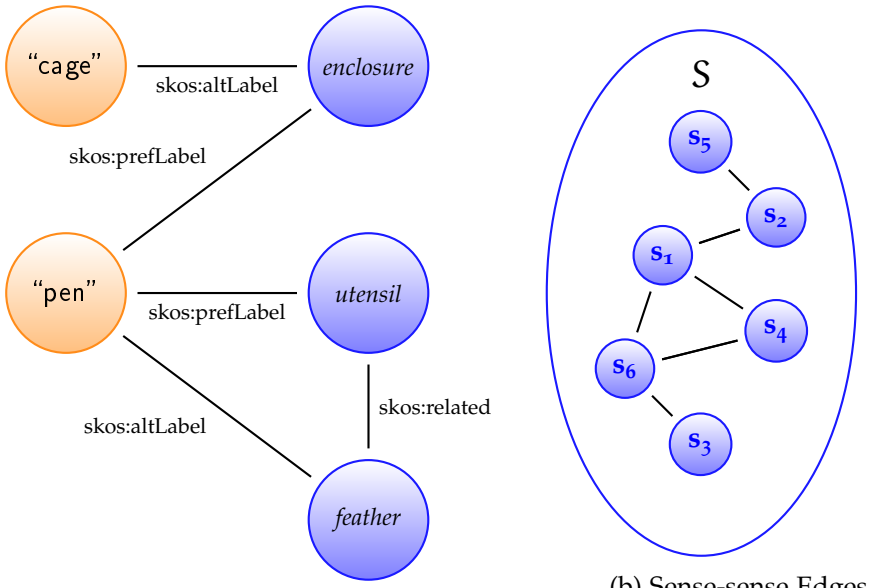


Figure 4: Arbitrary Sense Inventory

Paraphrasing Wilson and Watkins (1990, p37), $\mathcal{G}_\beta$ would have a vertex-set that could be split into sets $\mathcal{W}$ and $\mathcal{S}$ in such a way that each edge of the graph joins a vertex $w_i \in \mathcal{W}$ to a vertex $s_j \in \mathcal{S}$. Let β be this set of edges that denote *one-to-many* mappings between $\mathcal{W}$ and $\mathcal{S}$, therefore $\mathcal{G}_\beta = (\mathcal{W}, \mathcal{S}, \beta)$. Naturally not all LKBs are suitable for subgraph WSD, for example how would the necessary bipartite word/sense edges be established from a large collection of word collocation frequencies? On the other hand a LKB such as Wikipedia has these bipartite edges readily available, since each hyperlink has a text label (a word) that links to another page (a sense).



(a) Labelled Edges

(b) Sense-sense Edges

Figure 5: Extended complexities of sense inventories

Putting bipartite edges aside, Wikipedia has a wealth of semantic information that can be exploited when converted to sense inventory $\mathcal{G}$. For example in Figure 5 (a) the edge information in Wikipedia could be transliterated into standardised vocabulary for the semantic web such as SKOS[1] (Simple Knowledge Organisation System). Again in Figure 5 (b) the links between pages could be included as sense to sense edges. Sense inventories can vary widely, even if they are converted from the same LKB, therefore careful consideration is required when selecting (or building) one for WSD.

1 http://www.w3.org/TR/swbp-skos-core-spec

2.1.2 *Disambiguation Methodology*

At a glance across the text of any language, meaning and new information is absorbed through its *lexical composition*. Depending on the length of text being read, it could be interpreted as one of many structural subsequences of writing such as a *paragraph*, *excerpt*, *quote*, *verse*, *sentence*, among many others. Let $\mathcal{T} = (t_a, ..., t_b)$ be this subsequence of words, which can be interpreted as a sliding window. Again let $\mathbb{T} = (t_1, ..., t_m)$ be the larger body of text of length m, such as a *book*, *newspaper*, or *corpus of text*, that the sliding context window of length $b - a$ moves through to disambiguate. For each window, collect these text tokens in $\mathcal{T}$ into a set of words $\mathcal{W}$ as seen in Equation (2). Note word *identity* and *order* are preserved by subscript i.

$$\mathcal{W} = \bigcup_{i=a}^{b} t_i : \mathcal{T} = (t_a, ..., t_b) \tag{2}$$

$\therefore$ *if* $\mathcal{T}$ = (the, children, were, fishing, from, the, river, bank)

then $\mathcal{W} = \{bank_8, children_2, fishing_4, from_5, river_7, the_1, the_6, were_3\}$

Explicit WSD benefits from some preprocessing, in which the words in sequence $\mathcal{W}$ are mapped to a set of lemmas $\mathcal{L}$. To do this each word is tagged with its part-of-speech, then mapped to its lemmatisation, such that $\{w_a, ..., w_b\} \mapsto \{\ell_a, ..., \ell_b\}$. Lemmatisation is the *many-to-one* mapping of the different inflected forms of a word to a consolidated *lemma* (or *headword*)[2], as seen in Equation (3).

$$\ell_w : \mathcal{W} \to \mathcal{L} \tag{3}$$

$\therefore$ *if* $\mathcal{W} = \{bank_8, children_2, fishing_4, ..., were_3\}$

then $\mathcal{L} = \{bank_{(n),8}, children_{(n),2}, fish_{(v),4}, ..., be_{(v),3}\}$

2 For a detailed explanation of the processes leading up to lemmatisation (and beyond), see (Navigli, 2009, p12)

Given $\ell_i \in \mathcal{L}$, it needs to be disambiguated, as seen in Equation (4). Let $R(\ell_i)$ be a function that R*etrieves* from $\mathcal{G}$ all the senses, $\{s_{i,1}, s_{i,2}, ..., s_{i,k}\}$, that lemma ℓ_i could refer to:

$$
\begin{aligned}
& R(\ell_i) = \{s_{i,1}, s_{i,2}, ..., s_{i,k}\} \\
\therefore \; & \textit{if} \quad \ell_i = \textit{bank}_{(n),8} \\
\textit{then} \quad & R(\ell_i) = \{\textsc{Bank}_{(n),\text{finance},8}, ..., \textsc{Bank}_{(n),\text{land},8}\}
\end{aligned}
\tag{4}
$$

Given this, let $R(\mathcal{L})$ return all senses for all lemmas in the context window $\mathcal{L}$. Assume that $\mathcal{G}_\mathcal{L}$ is a semantic subgraph constructed from the senses retrieved by $R(\mathcal{L})$, which is detailed later in Section 2.2.1. Furthermore, explained later in Section 2.2.2, assume ϕ is a graph centrality measure employed to estimate the most appropriate sense, $s_{i,*} \in R(\ell_i) = \{s_{i,1}, s_{i,2}, ..., s_{i,k}\}$, which is added to a set of disambiguated senses $\mathcal{D}$. This subgraph-based WSD process is illustrated in Figure 6.

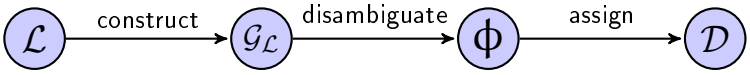


Figure 6: Subgraph-based WSD Process

Effectively, this is a classification problem to estimate $s_{i,*}$, the most appropriate sense for ℓ_i, by finding $\hat{s}_{i,*} = \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$ with $\mathcal{G}_\mathcal{L}$ taken as input. In the running example, a robust subgraph-based WSD system should be able to correctly estimate $\hat{s}_{i,*} = \textsc{Bank}_{(n),\text{land},8}$ (such that $i = 8$).

2.2 FURTHER EXPLANATION OF ϕ AND $\mathcal{G}_\mathcal{L}$

2.2.1 *Construction of Semantic Subgraph $\mathcal{G}_\mathcal{L}$*

For unsupervised subgraph-based WSD, the key publications that have advanced the field broadly construct subgraph, $\mathcal{G}_\mathcal{L}$, as either a union of

subtree paths, *shortest paths*, or *local edges*[3]. First $\mathcal{G}_\mathcal{L}$ is initialised, by setting $\mathcal{S}_\mathcal{L} := \bigcup_{i=1}^{n} R(\ell_i)$ and $\mathcal{E}_\mathcal{L} := \varnothing$. Next edges are added to $\mathcal{E}_\mathcal{L}$, depending on the desired subgraph type, by adding one of the following:

(a) *Subtree paths* of up to length L, via a Depth-First Search (DFS) of $\mathcal{G}$. In brief, **for each** sense $s_a \in \mathcal{S}_\mathcal{L}$, **if** a new sense $s_b \in \mathcal{S}_\mathcal{L}$, i.e. $s_b \neq s_a$, is encountered along a path $P_{a\to b} = \{\{s_a, s\}, ..., \{s', s_b\}\}$ with path-length $|P_{a\to b}| \leq L$, **then** add $P_{a\to b}$ to $\mathcal{G}_\mathcal{L}$. [cf. Navigli and Velardi (2005), Navigli and Lapata (2007), or Navigli and Lapata (2010)]

(b) *Shortest paths*, via a Breadth-First Search (BFS) of $\mathcal{G}$. In brief, **for each** sense pair $s_a, s_b \in \mathcal{S}_\mathcal{L}$, find the shortest path $P_{a\to b} = \{\{s_a, s\}, ..., \{s', s_b\}\}$; **if** such a path $P_{a\to b}$ exists and (optionally) $|P_{a\to b}| \leq L$, **then** add $P_{a\to b}$ to $\mathcal{G}_\mathcal{L}$ [cf. Agirre and Soroa (2008), Agirre and Soroa (2009), or Gutiérrez et al. (2013)]

(c) *Local edges* up to a local distance D. In brief, **for each** sense pair $s_a, s_b \in \mathcal{S}_\mathcal{L}$, **if** the distance in the text $|b - a|$ between the corresponding words w_a and w_b satisfies $|b - a| \leq D$, **then** add edge $\{s_a, s_b\}$ to $\mathcal{G}_\mathcal{L}$ (preferably with edge-weights). [cf. Mihalcea (2005) or Sinha and Mihalcea (2007)] (Note this subgraph is a hybrid, because only its vertices belong to $\mathcal{G}$)

In practice, subgraph edges may be *directed*, *weighted*, *collapsed*, or *filtered*. However to keep the distinctions between subgraph types simple, this is not included in our formalisation (albeit implemented in the algorithms of later experiments).

2.2.2 *Graph Centrality Measures* ϕ

In this section each graph centrality measure ϕ, that is used throughout this thesis to find $\hat{s}_{i,*} = \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$ is formalised.

3 Note that *local* describes the *local context*, which is typically the 2 or 3 words either side of a word, see (Yarowsky, 1993)

DEGREE CENTRALITY Firstly ϕ does not need to be a complicated measure, this is demonstrated by the success of ranking senses by their number of incoming and outgoing edges. Even though it is very simple, it performs surprisingly well against others for both in-degree (Navigli and Lapata, 2007) and out-degree (Navigli and Ponzetto, 2012a) simply written as functions in Equations (5) and (6).

$$\text{In-Degree} := I(s) \tag{5}$$

$$\text{Out-Degree} := O(s) \tag{6}$$

THE SEMANTIC WEB Next there are the graph centrality measures primarily used to disambiguate the *semantic web*, such as PageRank (Brin and Page, 1998), Hypertext Induced Topic Selection (HITS) (Kleinberg, 1999), and a *personalised* PageRank (Haveliwala, 2003); which have since been applied to WSD by Mihalcea (2005), Navigli and Lapata (2007), and Agirre and Soroa (2009) respectively.

HITS scores each page on the web in terms of being an *authority* ($A(s')$, a page with lots of content), and in terms of being a *hub* ($H(s')$, a page that points to a lot of authoritative pages, but lacks its own content). By substituting senses for pages in $\mathcal{G}_\mathcal{L}$, HITS is denoted in Equation (7). Notice HITS is mutually recursive, meaning the hubs score defines the authority score and vice-versa for each iteration.

$$H(s') = \sum_{s:(s,s')\in\mathcal{E}_\mathcal{L}} A(s) \quad ; \quad A(s') = \sum_{s:(s',s)\in\mathcal{E}_\mathcal{L}} H(s) \tag{7}$$

PageRank effectively models a random surfer on the internet, in which after a sufficiently large amount of time there is a probability of the surfer will end up at a particular page in the graph (or internet). However there is a probability $1 - d$ the surfer will jump to another page by for instance, clicking a bookmark. This is observed in the left term in Equation (8), in

which d is a damping factor that determines this. By substituting senses for pages in $\mathcal{G}_\mathcal{L}$, then for traditional PageRank, if the surfer does jump there is uniform probability $\frac{1}{|\mathcal{S}_\mathcal{L}|}$ to land on any of the other sense in the graph.

$$PR(s') = \frac{1-d}{|\mathcal{S}_\mathcal{L}|} + d \sum_{s,s' \in \mathcal{E}_\mathcal{L}} \frac{PR(s)}{O(s)} \tag{8}$$

Personalised PageRank takes this notion one step further, by biasing the chance to land on particular senses in the graph. Bias can be given towards seed senses (Agirre and Soroa, 2009) or senses that rank highly in sense annotated corpora (Gutiérrez et al., 2013) such as SemCor (Miller et al., 1993) .

SOCIAL NETWORK ANALYSIS Also included from the study of social networks is Betweenness Centrality (Freeman, 1979). Betweenness is defined as the a ratio of how often a sense s belongs in a shortest path $P_{a \to z}(s)$ from a to z out of all the shortest paths $P_{a \to z}$

$$BC(s) = \sum_{a,z \in \mathcal{S}: a \neq s \neq z} \frac{P_{a \to z}(s)}{P_{a \to z}} \tag{9}$$

WSD FOCUSED The graph centrality measures described so far were designed with other disciplines in mind. However for the last measure, Sum Inverse Path Length (Navigli and Ponzetto, 2012a,b), it has been designed with WSD in mind, therefore is a little less well known.

$$SIPL(s) = \sum_{p \in P_{s \to c}} \frac{1}{e^{|p|-1}} \tag{10}$$

This measure scores a sense by summing up the scores of all paths that connect to other senses in $\mathcal{G}_\mathcal{L}$ (i.e. senses that are not intermediate nodes, but have a mapping back to a lemma in the context window $\mathcal{L}$). In the words of Navigli and Ponzetto (2012a), $P_{s \to c}$ is the set of paths connecting

s to other senses of context words, with $|p|$ as the number of edges in the path p and each path is scored with the exponential inverse decay of the path length.

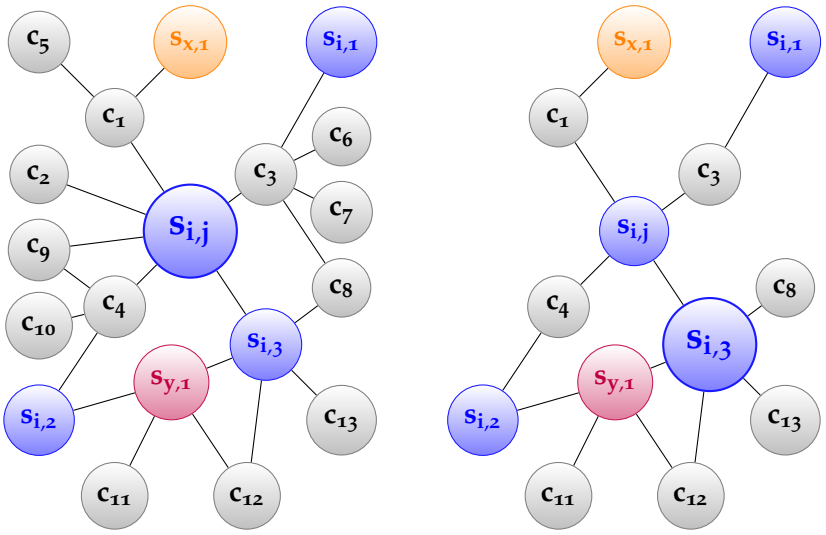
2.2.3 *Filtering/Refinement of Subgraph $\mathcal{G}_\mathcal{L}$*

After producing any of the subgraphs previously formalised in Section 2.2.1, it is also worth considering if they can be refined by means of adding or filtering vertices, edges, or whole paths. Subgraphs can occasionally be too sparse, providing little semantic information to work with. Or alternatively subgraphs can contain a lot of noise, thereby increasing the time to run graph centrality measures and reducing disambiguation accuracy.

One such issue that has been reported to be a problem by both Agirre and Soroa (2009, p36) and Navigli and Ponzetto (2012a, p238) is that of which, if very polysemous senses derived from the same lemma in $\mathcal{G}_\mathcal{L}$ are in close proximity to each other, they reinforce each other's score in whatever graph centrality measure is applied. Fortunately BabelNet ships with a filter called `SENSE_SHIFTS`, that removes paths added to $\mathcal{G}_\mathcal{L}$ that share senses derived from the same lemma. The effect of the filter is observable in Figures 7 (a) and (b). Let the graph centrality measure ϕ be the out-degree of a sense. Then for Figure 7 (a), the sense $s_{i,j}$ is the most appropriate since it has the highest out-degree. However after the `SENSE_SHIFTS` filter is applied in Figure 7 (b), $s_{i,3}$ is now has a higher out-degree and therefore is the most appropriate.

2.3 PRECISION, RECALL, & F-MEASURE

Precision, recall, and F-Measure are the conventional evaluation measurements used across the in-vitro WSD literature to compare results. Originally these measurements were used to evaluate Information Retrieval (IR) systems in terms of their *effectiveness*; this is assumed to be an indicator of user



(a) Unfiltered $\mathcal{G}_{\mathcal{L}}$ (b) $\mathcal{G}_{\mathcal{L}}$ + `SENSE_SHIFT`

Figure 7: Effects of Subgraph Filtering

satisfaction with search results. With this in mind, Van Rijsbergen (1979) describes effectiveness to be a measure of a system's ability to retrieve relevant documents while at the same time holding back non-relevant ones.

Thus in WSD terms, for a system to be effective, it should strive to provide disambiguations where possible (achieve high recall), however not so many as to significantly dilute the accuracy of all disambiguations made (maintain high precision). To report both precision ad recall as a single measurement, effectiveness is often reported as the F-measure. This is simply the harmonic mean between precision and recall.

Let $\mathcal{D}$ be the set of senses that the WSD system maps to the set of ambiguous words $\mathcal{W}$. Furthermore let $\mathcal{K}$ be the key set of correct senses for each lemma $\ell_i \in \mathcal{L}$.

Precision, P, is therefore:

$$P = \frac{|\mathcal{D} \cap \mathcal{K}|}{|\mathcal{D}|} \tag{11}$$

Recall, R, is:

$$R = \frac{|\mathcal{D} \cap \mathcal{K}|}{|\mathcal{L}|} \tag{12}$$

Finally, F-measure, F, is equal to:

$$F = 2 \cdot \frac{P \cdot R}{P + R} \sim 2 \cdot \frac{|\mathcal{D} \cap \mathcal{K}|}{|\mathcal{D}| + |\mathcal{L}|} \tag{13}$$

Refering to the methodology of subgraph-based WSD outlined in Section 2.1.2, assume in Equation (14) the WSD system needs to disambiguate content words only.

$$\begin{aligned}
\mathcal{T} &= (\text{fishing, for, eels, from, the, river, bank}) \\
\mathcal{L} &= \{bank_{(n),7}, eel_{(n),3}, fish_{(v),1}, river_{(n),6}\} \\
\mathcal{D} &= \{\textsc{Bank}_{(n),\text{financial},7}, \textsc{Fish}_{(v),\text{sport},1}, \textsc{River}_{(n),\text{stream},6}\} \\
\mathcal{K} &= \{\textsc{Bank}_{(n),\text{land},7}, \textsc{Eel}_{(n),\text{fish},3}, \textsc{Fish}_{(v),\text{sport},1}, \textsc{River}_{(n),\text{stream},6}\}
\end{aligned} \tag{14}$$

First, the text $\mathcal{T}$ undergoes part-of-speech tagging and lemmatisation to produce the set of lemmas $\mathcal{L}$. Next the WSD system attempts to disambiguate each lemma $\ell_i \in \mathcal{L}$ to produce an appropriate set of senses $\mathcal{D}$. Finally the key set $\mathcal{K}$ containing the correct disambiguations is used to score the disambiguation results in set $\mathcal{D}$. Notice in this example, the WSD system fails to produce output for the lemma $\ell_3 = eel_{(n),3}$. Also notice the senses for the lemma $\ell_7 = bank_{(n),7}$ are not the same in sets $\mathcal{D}$ and $\mathcal{K}$. Given $\mathcal{L}$, $\mathcal{D}$, and $\mathcal{K}$, the precision, recall, and F-measure can be calculated for this example as seen in Equation (15).

$$\begin{aligned}
P &= \frac{|\mathcal{D} \cap \mathcal{K}|}{|\mathcal{D}|} & \frac{2}{3} &= 0.667 \\
R &= \frac{|\mathcal{D} \cap \mathcal{K}|}{|\mathcal{L}|} & \frac{2}{4} &= 0.500 \\
F &= 2 \cdot \frac{|\mathcal{D} \cap \mathcal{K}|}{|\mathcal{D}| + |\mathcal{L}|} & 2 \cdot \frac{2}{3+4} &= 0.571
\end{aligned} \tag{15}$$

3

RESEARCH FOCUS

3.1 MOTIVATIONS

The author is motivated to take on the challenge of Word Sense Disambiguation (WSD) due to past research endeavours – these being an undergraduate final year project and a masters thesis. Research was undertaken into making Machine Translation (MT) accessible through mobile devices (Punchihewa et al., 2006), constructing a MT system (Manion and Punchihewa, 2008b), and improving MT output to sound more native-like (Manion and Punchihewa, 2008a) with the use of the Google Web 1T 5-gram corpus (Brants and Franz, 2006)[1]. As already covered in the literature review, the implications of WSD are perhaps the most evident in MT, after all it was this application that urged Weaver (1949) to write his influential memorandum on the matter.

Beyond the motivation stemmed from the author's passion for MT, is the concrete appeal of the unsupervised knowledge-based approach to WSD, that could have its success extended to MT. As already noted, corpus-based supervised approaches to WSD have dominated for some time now (Màrquez et al., 2007) but they are restricted by the availability of training data due to the *knowledge acquisition bottleneck* (Gale et al., 1992b). Therefore supervised approaches fail to be portable across alternative languages and domains if the annotated corpora do not exist. Conversely, knowledge-based approaches for WSD are usually applicable to all words in unrestricted text (Mihalcea, 2007). It is this innate scalability that is a motivation for the author to pursue knowledge-based approaches. Regardless of

1 http://catalog.ldc.upenn.edu/LDC2006T13 - Catalogue listing of Google Web 1T 5-gram corpus.

whether sense inventories can maintain a high level knowledge-richness (or structuredness) as they grow, their continued refinement by contributors should ensure any type of WSD/MT system that employs a knowledge-based approach can only hope to improve.

3.2 OBJECTIVES & SCOPE

The scope of this research focuses on investigating and exploring unsupervised knowledge-based WSD that makes used of semantic subgraphs. The objectives to be achieved are:

OBJECTIVE 1: To construct a large semantic graph $\mathcal{G}$ of concepts and named entities, to be indexed and accessible to a WSD system.

OBJECTIVE 2: To build a system that achieves WSD by constructing a context-seeded subgraph from $\mathcal{G}$, which then utilises a graph centrality measure to select the appropriate sense based on the context embedded in the subgraph.

OBJECTIVE 3: To evaluate and experiment with a range of subgraph construction methods and graph centrality measures, including the range of variables that influence them, in order to understand optimal conditions for unsupervised subgraph based WSD.

Part II

BRANCHES OF RESEARCH

Part II of this thesis contains a chapter devoted to each branch of research undertaken, in which each branch details peer-reviewed published research, that has been presented either orally or as a poster (or both) at various venues. These branches consist of:

- Chapter 4: The development of a data mining tool, that mines Wikipedia to construct a large semantic graph, in which smaller subgraphs can be shown to demonstrate the context embedded in its structure.
- Chapter 5: The development of a module to disambiguate concepts from a range of heterogeneous semantic graphs. This module was part of an automated system that constructs a taxonomy tailored to a given document collection.
- Chapter 6: The development of a new graph centrality measure (Peripheral Diversity), that was entered into SemEval 2013 Task 12 - "Multilingual Word Sense Disambiguation".
- Chapter 7: An iterative 'Sudoku Style' approach to subgraph-based WSD, that improves the performance of a range of graph centrality measures.

Each chapter contains a methodology, results achieved, and related literature where appropriate. Overall discussion of results is left to Part III of the thesis.

4

MINING SEMANTIC GRAPHS

This chapter details work completed, in which a data mining tool was developed to extract the underlying semantic graph inherently present in Wikipedia. Given that there are publicly available and continuously evolving tools capable of this it was an ambitious undertaking. However the key motivation was the flexibility that an in-house data mining tool could offer given it could be customised to the needs of a WSD system.

Despite this endeavour, work on the data mining tool eventually ceased in favour of using BabelNet. Not only does BabelNet provide the author with an indexed version of Wikipedia that is accessible through its free to use Application Programming Interface (API), it also maps Wikipedia pages to WordNet[1] and vice-versa. Furthermore, the development of BabelNet is ongoing and its publications clearly intend on improving the quality of WSD as one of its foremost objectives. Therefore it was decided that the data mining tool code base should be reused, in order to develop it into its own API which extends that of BabelNet's. Dubbed as the Daebak API, it has continued to be developed throughout the rest of this thesis to produce the results in successive chapters.

Finally, prior to ceasing work on the data mining tool experiments presented in this chapter were completed on the data that was successfully mined and indexed. Through visual illustrations, the experiments were aimed at demonstrating subgraphs do actually hold a unique context for the words that seed their construction. These particular results found in this chapter, were presented at the Machine Learning Summer School (MLSS) as a poster in Bordeaux, France.

1 See (Navigli and Ponzetto, 2012a) for concise details on BabelNet's construction, also see (Navigli and Ponzetto, 2012c) for examples of how to use its API.

4.1 MINING WIKIPEDIA

4.1.1 *The Consequences of Collaborative Editing*

Wikipedia is freely available to download[2] in the form of XML dumps. These XML dumps contain various splices of Wikipedia content, from edit histories to descriptive statistics. To begin with, only the English Wikipedia in its *then* current state was downloaded. At the time of download it was 5GB, then 26GB upon been unzipped with approximately 9.7 million Wikipedia pages.

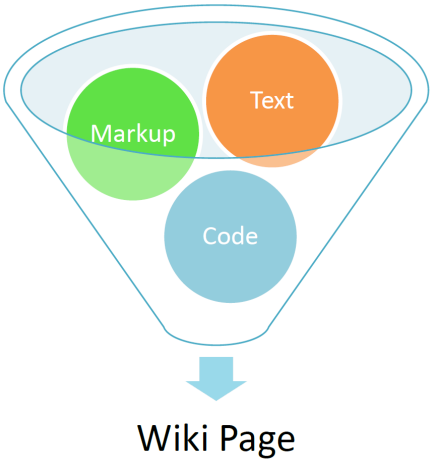


Figure 8: Ambiguity of Markup Language Clashes

As illustrated by Figure 8, the source of a Wikipedia page is a combination of various markup languages, code/template environments, running text, and more. The markup languages used in Wikipedia, such as HTML, XML, as well as Wikimedia's own markup language, need to be well-formed. That is, tags need to be appropriately open and closed, nested, and conform to the standards of the markup language in question. The XML dumps of Wikipedia reveal that contributing Wikipedians are not required to check the article they edited is well-formed before saving. Therefore it appears the collaborative editing process of Wikipedia leads to in-

2 http://en.wikipedia.org/wiki/Wikipedia:Database_download - Here the Wikipedia XML dumps can be downloaded.

consistent use of markup language. This is possibly because later edits by Wikipedians break up the previous correct use of markup from earlier edits. Two such examples that were found to occur often are given in Table 5.

Table 5: Examples of Markup Language Misuse

Misusage Type	Example	Issue
Unclosed Tags	<ref> ...	Where is the closing tag </ref> supposed to be?
Floating Tags	[[... [[...]]	Add extra]] tag? Or remove middle [[tag?

The first is when Wikipedians fail to close </ref> tags which hold website URLs that contain any number of characters that serve a purpose in another markup language. The second is when extra [[and]] tags used for internal hyperlinking are left floating about, either requiring removal or complementing. Figure 9 shows the source for the Wikipedia article page "CLS Holdings" found in the XML dump. It will be referred to throughout the rest of this chapter to provide sample output from the data mining tool.

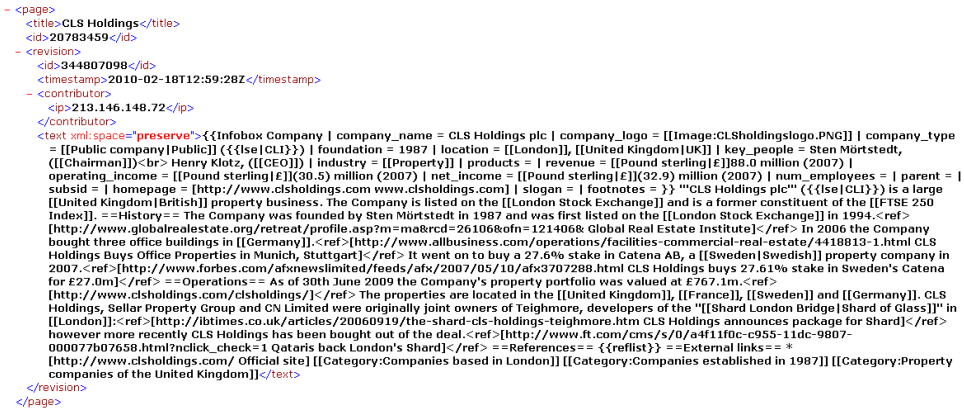


Figure 9: Sample Page (CLS Holdings) from Wikipedia XML Dump

While *this* page is well-formed in its simultaneous use of different markup languages, it does help one appreciate the complexities that could arise if it was not. The Wikimedia engine most likely compensates for the most common bad uses of markup languages, therefore so must the data mining tool. Most of these issues could be addressed with the SAX API[3] that the

3 http://www.saxproject.org/apidoc/org/xml/sax/package-summary.html - SAX Parser Documentation

data mining tool made use of, however not all pages could be parsed 100% correctly. When markup languages are used in conjunction with each other and are not well-formed, the boundaries for parsing environments become unclear. In such cases as the examples in Table 6 below, the data mining tool attempts to recover as much information as possible.

Table 6: Examples of Markup Language / Text Similarity Clashes

Clash	Example / Explanation
:	It represents an *indent* in Wiki Markup, however in Text it is used in *times* (2:45), *URLs* (http://...) and many other expressions
==	It represents a *level 2 heading* (<h2>) in Wiki Markup, however in Code written within Text it it often used for *if-statements* (if x == 1)...

If a page was too difficult for the parser to make sense of then it is added to a list of *voided* pages. This list was a useful reference in understanding how to parse in a more robust manner, to eventually achieve a rate of voiding less than 1 in 1000 pages.

4.1.2 *The Structure of Wikipedia*

Through parsing the XML dump with the data mining tool, the greater structure of Wikipedia began to emerge. Illustrated by Figure 10 in terms of page and hyperlink types, Wikipedia contains both a mix of cycles and trees. This ties in with the findings of Lizorkin et al. (2009), in which they showed Wikipedia's structure to be a hierarchy of communities (or alternatively, strongly connected components). As for cycles, a good example they offered taken from category links was "The Beatles" < "Apple Records artists" < "Apple Records" < "Apple Corps" < "The Beatles".

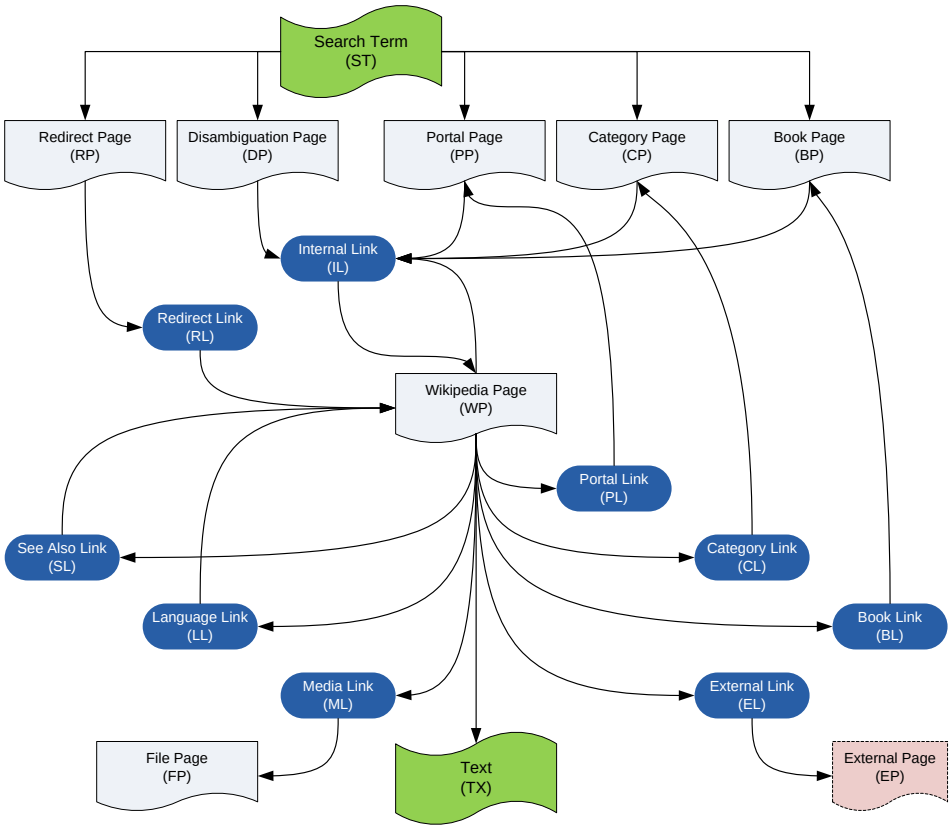


Figure 10: Wikipedia Structure Revision

4.1.3 *Indexing Methodology of Wikipedia*

After careful consideration of Wikipedia's structure, the following information in Figure 11 was mined from the Wikipedia XML dump.

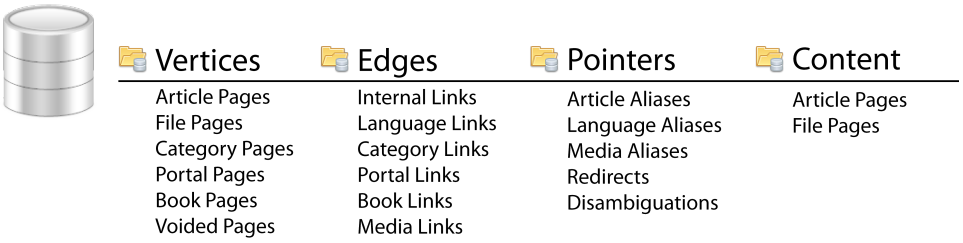


Figure 11: Indexed Information from Wikipedia

Information was first stored as raw text files, with tab delimitations where appropriate. This storage medium provides convenience of running the data mining tool only once, however having the flexibility to store the information in a range of database designs for later experiments. Effectively there were four classes of information mined:

1. **Vertices** - the title of each page was parsed and recorded as a vertex, along with the page type

2. **Edges** - the outgoing hyperlinks for each page were parsed and recorded as edges, again according to the page type

3. **Pointers** - the redirects, disambiguations, and aliases (alternative text for links) for each page were parsed and recorded

4. **Content** - the content of Wikipedia article pages and file pages was parsed and recorded as nice clean text without any markup language included

Over the next few pages examples of this captured information are given. Note the selected examples are not representative of everything the data mining tool can capture, just the information relevant to experiments completed in later sections of this chapter. To begin with, Table 7 illustrates

the useful links and textual content captured from the source of the "CLS Holdings" page from Figure 9.

Also given in Table 8 are two pointer type examples: *redirects* and *aliases*. Within the scope of this thesis, pointers are defined as a snippet of text that has the semantic realisation of another snippet of text. To describe each pointer, a redirect typically compensates for variances in spelling or alternative names people may use when searching for a Wikipedia article. An alias on the other hand is the alternative text used for hyperlinks in Wikipedia. Aliases were collected for hyperlinks to English articles, Non-English articles, and media files. This is one aspect in which our data mining tool sought to distinguish itself from competing alternatives at the time. Collectively, pointers are comparable to a *very short* gloss found in WordNet (Fellbaum, 1998) or alternative labels in SKOS[4]. SemEval tasks such as Cross-Lingual Lexical Substition (Mihalcea et al., 2010) or Semantic Textual Similarity (Agirre et al., 2013) are definitely candidates that can make use of pointers.

The "CLS Holdings" page also links to the page in Table 9 where the image is stored and its respective information is noted. With this information stored by the data mining tool, images can easily be pulled down from Wikipedia through knowing the file name and opening the following URL.

`http://en.wikipedia.org/wiki/File:` + *the file name* (`CLSholdingslogo.PNG`)

Finally in Table 10 some examples of media aliases are shown. The same image can be used in several Wikipedia pages, therefore can take on several text descriptions.

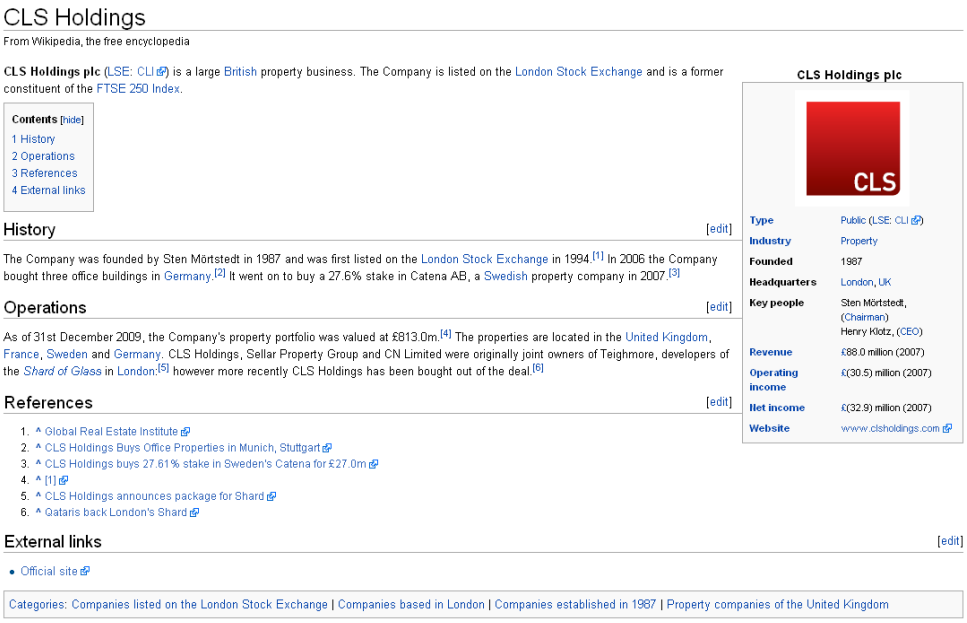
4.2 CONTEXT GRAPHS

After mining the previously described data, Wikipedia could now be modelled as a large semantic graph, that if queried with a word could return a

4 `http://www.w3.org/2004/02/skos/core` - Definition of `skos:altLabel` found here.

Table 7: Page Content

Wiki Page



CLS Holdings

From Wikipedia, the free encyclopedia

CLS Holdings plc (LSE: CLI) is a large British property business. The Company is listed on the London Stock Exchange and is a former constituent of the FTSE 250 Index.

Contents [hide]
1 History
2 Operations
3 References
4 External links

History [edit]

The Company was founded by Sten Mörtstedt in 1987 and was first listed on the London Stock Exchange in 1994.[1] In 2006 the Company bought three office buildings in Germany.[2] It went on to buy a 27.6% stake in Catena AB, a Swedish property company in 2007.[3]

Operations [edit]

As of 31st December 2009, the Company's property portfolio was valued at £813.0m.[4] The properties are located in the United Kingdom, France, Sweden and Germany. CLS Holdings, Sellar Property Group and CN Limited were originally joint owners of Teighmore, developers of the *Shard of Glass* in London;[5] however more recently CLS Holdings has been bought out of the deal.[6]

References [edit]

1. ^ Global Real Estate Institute
2. ^ CLS Holdings Buys Office Properties in Munich, Stuttgart
3. ^ CLS Holdings buys 27.61% stake in Sweden's Catena for £27.0m
4. ^ [1]
5. ^ CLS Holdings announces package for Shard
6. ^ Qataris back London's Shard

External links [edit]

- Official site

Categories: Companies listed on the London Stock Exchange | Companies based in London | Companies established in 1987 | Property companies of the United Kingdom

CLS Holdings plc

Type	Public (LSE: CLI)
Industry	Property
Founded	1987
Headquarters	London, UK
Key people	Sten Mörtstedt, (Chairman) Henry Klotz, (CEO)
Revenue	£88.0 million (2007)
Operating income	£(30.5) million (2007)
Net income	£(32.9) million (2007)
Website	www.clsholdings.com

Page Content

CLS Holdings

CLS Holdings plc is a large British property business. The Company is listed on the London Stock Exchange and is a former constituent of the FTSE 250 Index.

2History

The Company was founded by Sten Mortstedt in 1987 and was first listed on the London Stock Exchange in 1994. In 2006 the Company bought three office buildings in Germany. It went on to buy a 27.6% stake in Catena AB, a Swedish property company in 2007.

2Operations

As of 30th June 2009 the Company's property portfolio was valued at £767.1m. The properties are located in the United Kingdom, France, Sweden and Germany. CLS Holdings, Sellar Property Group and CN Limited were originally joint owners of Teighmore, developers of the Shard of Glass in London however more recently CLS Holdings has been bought out of the deal.

	Internal Links		Category Links
x2	United Kingdom	x1	Companies based in London
x2	London Stock Exchange	x1	Companies established in 1987
x1	FTSE 250 Index	x1	Property companies of the United Kingdom
x2	Germany		
x2	Sweden		
x1	France		
x1	Shard London Bridge		
x1	London		

Table 8: Examples of Pointers

Redirects		
Ein Kuniyye	⟶	Ein Qiniyye
ISN 22	⟶	Shakhrukh Hamiduva
HMS Junon (1810)	⟶	French frigate Bellone (1807)
Sunday Inquirer Magazine	⟶	Philippine Daily Inquirer
Patents are bad	⟶	Criticism of patents
Intellectual property is bad	⟶	Criticism of intellectual property
Welsh Communist Party	⟶	Communist Party of Wales
Survivors remake	⟶	Survivors (2008 TV series)
Article Aliases		
political issues of water	⟶	water politics
Jordan River Valley	⟶	Jordan Valley (Middle East)
alluvial	⟶	fluvial terrace
morphology	⟶	Geomorphology
delta	⟶	river delta
Hasbani	⟶	Hasbani River
Dan	⟶	Dan River (Israel)
Palestine	⟶	Palestinian territories

wealth of information about the typical contexts it is found in. With context being a subset of Wikipedia pages semantically associated with the target word. This context could then be used to derive semantic subgraphs in order to achieve WSD.

4.2.1 *From Wikipedia to Context Graph*

Wikipedia can be interpreted as a *multigraph* $\mathcal{G}_w$, with a set of pages $\mathcal{P} = \{p_1, ..., p_n\}$ connected by a multiset of *directed* and *unweighted* hyperlinks $\mathcal{H}$. Therefore $\mathcal{P}$ and $\mathcal{H}$ are the respective *vertices* and *edges* for the Wikipedia graph, such that $\mathcal{G}_w = (\mathcal{P}, \mathcal{H})$. Note that $\mathcal{H}$ is a *multiset* that allows duplicate (or parallel) edges to account for when Wikipedians *hyperlink* to the same page several times in the text of the page they are editing.

When two pages are hyperlinked, this indicates there is some semantic relationship between them. However since the hyperlinks are unweighted

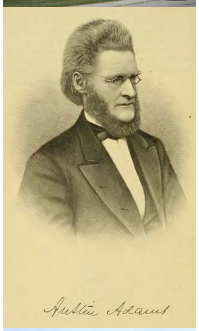
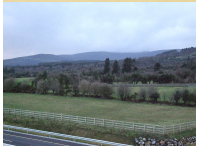
Table 9: File Page

File Page

File Details

File Name	=	CLSholdingslogo.PNG
Description	=	This is the logo of CLS Holdings.
Source	=	The CLS Holdings website
Portion	=	All.
Article	=	CLS Holdings
Purpose	=	To illustrate the organisation in question in the CLS Holdings article.
Resolution	=	No, the image is already at a low resolution.
Replaceability	=	No. This is irreplaceable as any image of the logo is copyrighted.

Table 10: Images Corresponding to the Media Aliases

Media Alias	Image
Carland Cross windfarm ↪ `Carland Cross Wind Farm.jpg`	
Singles defending champion Philipp Kohlschreiber ↪ `PhilippKohlschreiber GerryWeberOpen2008.jpg`	
Royal West Campus ↪ `RoyalWestHowardCoad.jpg`	
Austin Adams ↪ `Austin Adams - History of Iowa.jpg`	
Part of Glengarra Wood, December 2008 ↪ `Glengarra.JPG`	

edges in $\mathcal{G}_w$, the strength of the semantic relationship is unknown. Typically for the article pages in Wikipedia, the most semantically significant hyperlinks are found in the first few paragraphs, with hyperlinks in the body of the article having a more happenstance semantic association to the article's topic. However this is never a certainty, therefore to understand which hyperlinks are the most significant and best represent the topic of an article page, the hyperlinks in $\mathcal{G}_w$ need to be weighted to reflect this. Let this edge weighted semantic graph that more acutely represents context be $\mathcal{G}_c = (\mathcal{P}, \mathcal{H}, m)$, such that $m : \mathcal{H} \to [0, 1]$ is a mapping function of edges to semantic weights. The methodology of calculating edge weights is described in the next section.

4.2.2 *Step 1: Representing Pages as HF-IPF Vectors*

The measure chosen to weight the significance of a page's hyperlinks is a modification of TF-IDF (Term Frequency - Inverse Document Frequency). Traditionally it is used in Information Retrieval (IR) to indicate how significant a term is in a document relative to other documents within a whole collection (Salton and McGill, 1983). The modification of TF-IDF comes by way of substituting *hyperlinks* for *terms* and *pages* for *documents*, therefore re-acronymised to HF-IPF.

To implement HF-IPF, the first step is to represent each Wikipedia page $p_i \in \mathcal{P}$ as a hyperlink frequency vector. Given the adjacency matrix A for $\mathcal{G}_w$, this is simply the row vector $\vec{A}_i = \langle \alpha_{i,1}, ..., \alpha_{i,n} \rangle$ anchored by i to page p_i, such that $n = |\mathcal{P}|$. However each hyperlink frequency $\alpha_{i,j}$ in $\vec{A}_i$ only gives its *local* significance to page p_i, without factoring in its *global* significance to all pages in $\mathcal{P}$. To account for global significance, each hyperlink frequency is scaled by the logarithm of the inverse frequency of the hyperlink occurring in any of the pages in $\mathcal{P}$. The HF-IPF modified frequency $f_{i,j}$

for $\alpha_{i,j}$, the number of hyperlinks between pages p_i and p_j, is calculated by Equation (16) below.

$$f_{i,j} = \alpha_{i,j} \times \log\left(\frac{n}{\sum_{i=1}^{n} \alpha_{i,j}}\right) \tag{16}$$

Again the adjacency matrix $\mathcal{A}$ for $\mathcal{G}_w$ is useful, since $\sum_{i=1}^{n} \alpha_{i,j}$ is simply the sum of the frequencies in column vector $\vec{A}_j$. Using this equation, a HF-IPF vector $\vec{F}_i = \langle f_{i,1}, ..., f_{i,n} \rangle$ can be produced for each page p_i. Now each page p_i is associated with a $\vec{F}_i$ vector of local hyperlink frequencies scaled to their global semantic significance.

4.2.3 *Step 2: Weighting Hyperlinks based on Cosine Similarity*

For the second step, the edge weights for $\mathcal{G}_c$ can now be calculated from the $\vec{F}_i$ vectors associated with each $p_i \in \mathcal{P}$. For this Cosine Similarity (CS), a measure between 0 and 1 indicating the intersection between two vectors was chosen. This translates to a CS score of 0 meaning there is no similarity at all between two pages, and 1 meaning the two pages have exactly the same inbound and outbound edges (as well as frequency of them), suggesting they are semantically equivalent. Since Wikipedia is a scale-free graph (Voss, 2005), $\vec{F}$ vectors are very sparse. This makes CS a very efficient measure of semantic similarity because only non-zero values need to be calculated. CS is defined by Equation (17) which completes the calculation of edge weights. Let E be a matrix that stores these edge weights, indexed in the same way as adjacency matrix A stores edge frequencies.

$$e_{i,j} = \frac{\vec{F}_i \cdot \vec{F}_j}{\|\vec{F}_i\| \|\vec{F}_j\|} \tag{17}$$

Algorithm 1 summarises both of the previous two steps to calculate edge weights. With function `GetHFIPF`(A, i, j, n) representing Equation (16) and `GetCosineSimilarity`(F, i, j) representing Equation (17).

Algorithm 1: Calculating Edge Weights for Semantic Graph $\mathcal{G}_c$

Input: $\mathcal{G}_w = (\mathcal{P}, \mathcal{H})$
Output: E
$n \leftarrow |\mathcal{P}|$;
$A \leftarrow$ `GetAdjacencyMatrix` $(\mathcal{G}_w)$;
$F \leftarrow$ `GetZerosMatrix` (n, n);
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** n **do**
    $f_{i,j} \leftarrow$ `GetHFIPF` (A, i, j, n);
$E \leftarrow$ `GetZerosMatrix` (n, n);
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** n **do**
    $e_{i,j} \leftarrow$ `GetCosineSimilarity` (F, i, j);

There are also the standard functions `GetAdjacencyMatrix`$(\mathcal{G}_w)$ that returns the hyperlink frequencies of $\mathcal{G}_w$ and `GetZerosMatrix`(n, n) that initialises an $n \times n$ matrix of zeros. The final output matrix E that contains the calculated edge weights can be used as the map $m : \mathcal{H} \to [0, 1]$. Therefore Algorithm 1 maps each edge (p_i, p_j) to the edge weight $e_{i,j}$, i.e. $(p_i, p_j) \mapsto e_{i,j}$. This completes the construction of $\mathcal{G}_c = (\mathcal{P}, \mathcal{H}, m)$, which is used to create subgraphs in the following section.

4.3 CONTEXT VISUALISATION RESULTS

The visualisation results that follow were presented in poster form at MLSS 2011 in Bordeaux, France. These results, represented in visual form, demonstrate how semantic subgraphs can indeed hold context, particularly if they are weighted in a meaningful way. These subgraphs could go on to complement Machine Translation (MT) or Information Retrieval (IR) applications. Three cases will now be presented that reflect this potential, with each explained in turn.

4.3.1 *Case 1: Competing Contexts*

The first case put forward constructs a subgraph from $\mathcal{G}_c$ with one single lemma. For example, consider the intended senses of the word "chip", in which a noun *premodifier*[5] determines the type of chip it is (e.g. *potato* vs *chocolate* "chip"). If "chip" is lemmatised as the second word to $chip_{(n),2}$, let $R(\ell_2)$ retrieve from $\mathcal{G}_c$ the set of candidate senses in Equation (18).

$$R(\ell_2 = chip_{(n),2}) = \{\textsc{Chip}_{(n),wood,2}, \textsc{Chip}_{(n),micro,2}, \textsc{Chip}_{(n),potato,2}, \textsc{Chip}_{(n),casino,2}, \textsc{Chip}_{(n),chocolate,2}, \textsc{Chip}_{(n),mod,2}\} \tag{18}$$

Admittedly, there are more senses for the lemma $chip_{(n),2}$, yet for ease of interpretation the number is capped at six. Next, a subtree subgraph as formally described in Section 2.2.1 with $L = 1$ is constructed for each sense in Equation (18). These six subgraphs together make up a supergraph $\mathcal{G}_{chip}$, which is a subgraph of $\mathcal{G}_c$ and takes on its respective edge weights. From $\mathcal{G}_{chip}$ the page cloud produced by Wordle[6] in Figure 12 illustrates six competing contexts of the senses retrieved by $R(\ell_2 = chip_{(n),2})$.

What is immediately striking, is the dominance of the context for the sense $\textsc{Chip}_{(n),micro,2}$, there are more pages for this sense and furthermore some of these pages, such as BCDMOS and Four-phase logic exhibit a higher degree of semantic similarity than observed for the most semantically related pages of other senses (such as Corn chip, XBOX Modchips, and Poker chip). In fact $\textsc{Chip}_{(n),micro,2}$ could be considered as the Most Frequent Sense (MFS) baseline for the lemma $chip_{(n),2}$ in Wikipedia. Also noticeable, is the Zipfian-like distribution in which each consecutively less dominant sense in the subgraph, is much less visually present.

5 Noun premodification, i.e. (modifier) noun + (head) noun sequences, contain only content words, with no function word to show the meaning relationship between the two parts, i.e. Poker chip → a chip used for playing poker (Biber et al., 2002, p272-274).

6 http://www.wordle.net - Wordle Homepage

Figure 12: A colour is associated with each sense (or page p) for the lemma $chip_{(n),2}$. Font size is determined by the edge weights in $\mathcal{G}_{\text{chip}}$ between page p and other pages that it has inbound or outbound links to, which are representative of the sense's context.

4.3.2 *Case 2: Subtle Differences*

The choice of words in a sentence naturally conjure up different images in one's mind, such as when reading the statements made in Figure 13.

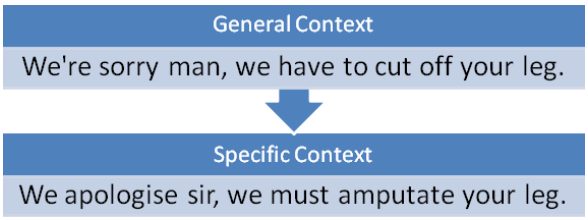


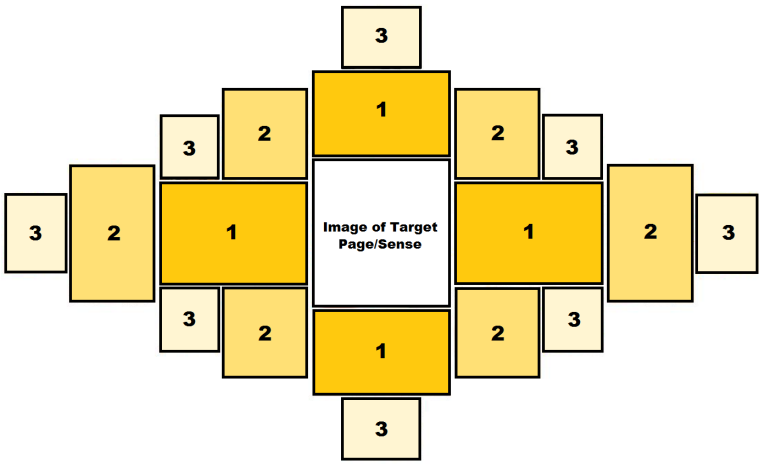
Figure 13: Choosing Words Carefully

In the non-linguistic context of a hospital, naturally patients would be a little alarmed if they heard the first sentence rather than the second. Recall image information was recorded by the data mining tool in Tables 9 and 10. The difference between *cut* and *amputate* is subtle, but very significant. For the next set of results the image files in Wikipedia are utilised.

Again the same type of *subtree* subgraph is created as in Section 4.3.1 with L = 1 and the edge weights of $\mathcal{G}_c$ made available. Except on this occasion,

rather than using page titles, the first image from each page is used. For this some Hyper Text Markup Language (HTML) was generated from the data mining tool output, to arrange these semantically related images into an image cloud viewable in a web browser. The target image is placed in the middle, and it is surrounded by images from other pages that are weighted as the most semantically significant to the target image. The larger and closer each image is to the target image, the more semantically significant the page it comes from is. As seen in Figure 14 (a), the most semantically significant images are grouped into three tiers. The top 4 are in first tier, next the top 6-10 are in the second tier, and finally the top 11-18 are in the third tier[7]. For example see Figure 14 (b), that has the Korean dish "Kimchi" as the target image. Notice there are many other types of Kimchi, Korean dishes, and ingredients that are also associated with it.

7 Note if a semantically related page had no image, it was discarded in order to consider the next most semantically related one.



(a) Tier Organisation of Image Clouds

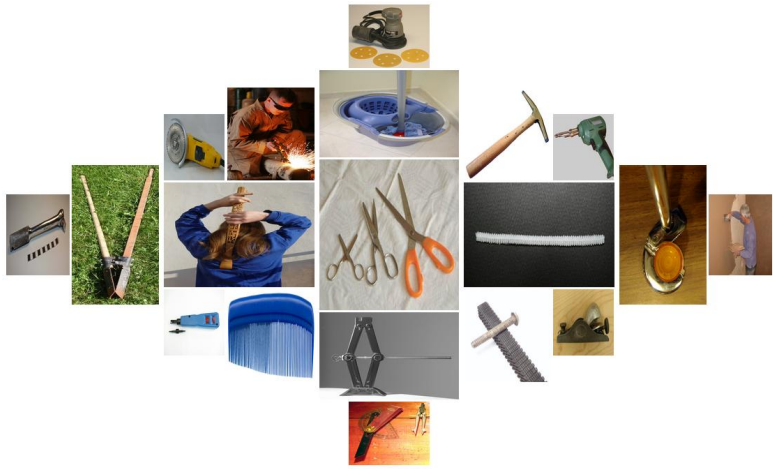
(b) Image Cloud for Subgraph $\mathcal{G}_{\text{kimchi}}$

Figure 14: The tier-based organisation of image clouds, with closer and larger images to the target image in the centre exhibiting more semantic significance according to their edge weights in $\mathcal{G}_c$.

Now to return to the subtle yet significant difference between the intended sense of *amputating* and *cutting off* limbs. A subgraph was made for each sense with the closest Wikipedia page equivalent possible. These were the pages Amputation and Cutting respectively, for which subtree subgraphs $\mathcal{G}_{\text{amputation}}$ and $\mathcal{G}_{\text{cutting}}$ were constructed. As image clouds they can be seen in Figures 15 (a) and (b), illustrating the context of each word and perhaps the images that the reader him/herself would conjure up upon hearing them. A patient would certainly not desire any of the torturous devices such as a *grinder*, *drill*, or *hand plane* found in Figure 15 (a) near him/her for a surgical amputation. In a hospital ward (rather than a workshop or butchery) more precise and hygienic tools would be expected to be in use.

4.3.3 *Case 3: Specified vs Unspecified Contexts*

Finally in Figure 16 two different image clouds are produced from subgraph $\mathcal{G}_{\text{amputation}}$. Figure 16 (a) contains the 18 *most* semantically related images, whereas Figure 16 (b) contains the 19 *least* semantically related (the extra image is accounted for by the least related image taking the centre position of the image cloud). Even though all the images in the visual context given by the page cloud in Figure 16 (b) are hyperlinked to the page Amputation, it would be very difficult to deduce that it was in comparison to Figure 16 (a). This illustrates how a lot of Wikipedia links are very weakly related to they page the are found in, which also highlights the importance of mapping globally-scaled semantic edge weights such as was formalised in this chapter.

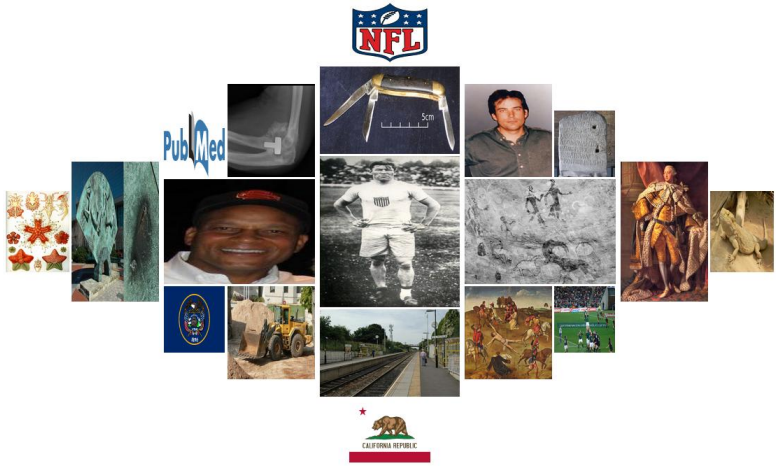


(a) Image Cloud for Subgraph $\mathcal{G}_{\text{cutting}}$

(b) Image Cloud for Subgraph $\mathcal{G}_{\text{amputate}}$

Figure 15: Two different image clouds visually illustrating the subtle yet significant difference in contextual use of the synonymous words cut off and amputate.

(a) Image Cloud for the *Most* Semantically Related Images in Subgraph $\mathcal{G}_{\text{amputate}}$



(b) Image Cloud for the *Least* Semantically Related Images in Subgraph $\mathcal{G}_{\text{amputate}}$

Figure 16: Two image clouds constructed from the same subgraph $\mathcal{G}_{\text{amputate}}$, illustrating the *most* and *least* semantically related images respectively

5

DISAMBIGUATING CONCEPTS THAT ORIGINATE FROM HETEROGENEOUS SEMANTIC GRAPHS

Based on the achievements of mining context from Wikipedia in the previous chapter, the author was offered an internship for 6 months at the company Pingar. This involved developing a system that automatically generates a taxonomy[1] from a document collection. This was a joint project funded by New Zealand's Ministry of Science & Innovation (MSI), between Pingar and Waikato University, enabling the author to work with Dr Alyona Medelyan, Dr Jeen Broekstra, Dr Anna Divoli, Dr Anna Huang, Dr David Milne, and Prof Ian Witten.

This chapter details the disambiguation module developed by the author which was implemented into the taxonomy generation system. Leading up to this module, words (or ngrams) found in the documents have several concepts and named entities mapped to them from a range of heterogeneous semantic graphs. The purpose of the disambiguation module is to ascertain for each word, whether the concepts and named entities that are mapped to it are semantically equivalent.

Finally the taxonomy generation system that makes use of this disambiguation module has been published as a peer reviewed paper (Medelyan et al., 2013) which can be found in Appendix B.1. This paper was presented at the proceedings of ESWC 2013 *in Montpellier, France. More information about the taxonomy generation system as a product can be found by visiting Pingar's company website.*

1 A taxonomy is a hierarchical structure to formally classify a set of concepts or named entities.

5.1 FOCUSED SKOS TAXONOMY EXTRACTION PROCESS (F-STEP)

The F-STEP system is comprised of several processing steps in order to make sure the resulting taxonomy is as comprehensive as possible, exhibiting both breadth and depth that draws from a range of semantic graphs. The focus of this chapter is the disambiguation module (or processing step) of the F-STEP system. However before getting into details, it is worth briefly explaining *all* of the processing steps the F-STEP system has in order to better understand its purpose. Therefore the explanations of each step will be purely non-technical and introduce the technologies and resources utilised.

5.2 A BRIEF STEP BY STEP SYSTEM OVERVIEW

The F-STEP system is designed to help companies organise their internal documents (text, spreadsheets, slides, etc). The motivation is that if a taxonomy can be generated based upon a document collection then that taxonomy could be useful in managing those documents. Generally, information architects design taxonomies which is an expensive task in terms of time and money. Furthermore it is very difficult to keep a taxonomy up to date when documents are constantly added to the collection. F-STEP aims to automate this process, with the end result being that companies will be able to better structure and manage their data instantaneously without such expense and effort.

Taken from (Medelyan et al., 2013), Figure 17 over the following page is a view of the F-STEP system's processing steps and their order of occurrence, for which each will now be briefly explained. It is also worth noting here that the author worked on all parts of the F-STEP system, implementing the code of the other co-authors. As for the focus of this chapter, the disambiguation module, this was the author's own innovation.

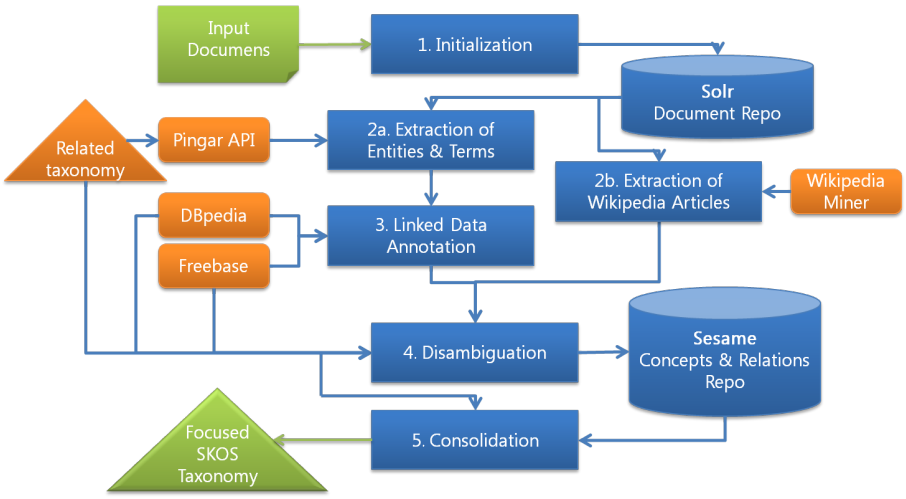


Figure 17: System View of F-STEP

5.2.1 *Processing Step 1: Initialisation*

Processing step 1 achieves *initialisation* by first indexing each document into a Solr[2] repository. Solr has full-text search and indexing features for document collections that F-STEP makes use of, accessing documents via its URL referencing system. Before loading documents into Solr they need to be stripped of any formatting. For this Apache Tika[3] was used, since it is packaged with a number of useful libraries for parsing documents.

In conjunction with loading documents into Solr, a Sesame[4] repository is automatically set up to store semantic relations deduced from the document collection. The Sesame repository can be queried with the use of SPARQL[5] – a query language widely used to add, remove and manipulate semantic data. In fact, several well known semantic graphs including Freebase and DBpedia have SPARQL endpoints, which allow the public to query their semantic data.

2 `http://lucene.apache.org/solr` - Apache Solr Homepage
3 `http://tika.apache.org` - Apache Tika Homepage
4 `http://rdf4j.org` - Sesame Homepage
5 `http://www.w3.org/TR/rdf-sparql-query` - Latest W3C SPARQL Query Language for RDF Recommendation

5.2.2 *Processing Steps 2(a), 2(b) & 3: Extraction & Annotation of Concepts & Named Entities*

Processing steps 2(a) and 2(b) are focused on *extraction* – these being the Entity Extraction service of the Pingar API[6], along with the Wikify service of Wikiminer[7]. Following this, processing step 3 involves linked data *annotation*, which is the querying of the DBpedia and Freebase endpoints using SPARQL to map even more concepts to the target ngrams. These mapping steps are detailed below.

PROCESSING STEP 2(A): PINGAR ENTITY EXTRACTION Pingar Entity Extract (PEE) is one of the many services provided by the Pingar API. PEE achieves two types of entity extraction – the first is the mapping of *named entities* such as locations, dates, and people to *ngrams*, and the second is the mapping of *concepts* from an input taxonomy, again to *ngrams*.

PROCESSING STEP 2(B): WIKIFY EXTRACTION Wikify is one of the many services provided by Wikiminer, in which the key concepts in a block of plain text are hyperlinked to pages in Wikipedia. If Wikipedia pages are considered to represent *concepts* and *named entities*, then Wikify simply achieves the task of mapping them to *ngrams*.

PROCESSING STEP 3: DBPEDIA & FREEBASE ANNOTATION Finally there is the *annotation* processing step, in which each of the *named entities* found by PEE for a target *ngram* are iterated over to see if they can be located in either DBpedia[8] or Freebase[9] by querying their SPARQL endpoints (note the entity type must also match). If a *named entity* exists in DBpedia or Freebase, it is also mapped to the target *ngram*.

6 http://apidemo.pingar.com - Demo Page for the Pingar API
7 http://wikipedia-miner.cms.waikato.ac.nz/services/?wikify - The Wikify service, hosted by SourceForge and the University of Waikato
8 http://dbpedia.org/sparql - DBpedia SPARQL Endpoint
9 http://sparql.freebase.com - A Collection of Freebase SPARQL Endpoints

5.2.3 *Processing Step 4: Disambiguation of Concept & Named Entity Mappings*

The *disambiguation* processing step is yet to be elaborated on in greater detail later, but is mentioned here briefly to continue the flow of F-STEP's description. The previous processing steps mapped *concepts* and *named entities* from a range of semantic graphs to the *ngrams* in the documents. The purpose of the disambiguation processing step is to ascertain whether these particular mappings are correct, then discard those that are not while merging the rest together. This results in one *canonical* concept mapping per *ngram*, that represents one or more semantically equivalent *concepts* or *named entities* that originate a range of heterogeneous semantic graphs.

5.2.4 *Processing Step 5: Consolidation of Taxonomy*

The *taxonomy consolidation* processing step is where all the *extracted*, *annotated* and then *disambiguated* concept and named entity mappings are examined to look for broader/narrower relations in the semantic graphs they originated from. Based on a collection of heuristics, the final taxonomy is populated and consolidated.

For further details on any of these processing steps, reading the paper (Medelyan et al., 2013) contributed to by the author is recommended.

5.3 SKOS: SIMPLE KNOWLEDGE ORGANISATION SYSTEM

To construct a taxonomy, the range of semantic graphs utilised and the documents in the collection need to be described in a well defined semantic vocabulary. For this the Simple Knowledge Organisation System (SKOS[10]) vocabulary was chosen, of which the fraction of it that is used in the F-STEP system is defined in Table 11 on the following page.

10 http://www.w3.org/TR/2009/REC-skos-reference-20090818 - The W3C Recommendation Document for SKOS

Table 11: The SKOS Vocabulary Specifically Employed in F-STEP

SKOS Vocabulary	Meaning	Example
skos:Concept	The concept class	*Beer* is a *concept*
skos:prefLabel	The preferred label for a concept	*Beer* is the preferred label of the *Concept*
skos:altLabel	The alternative label for a concept	A *cold one* is an alternative label for *beer*
skos:related	The related concept of a concept	*Wine* is a related concept to *beer*
skos:closeMatch	The close match of a concept	*Guinness* is a close match to *beer*
skos:exactMatch	The exact match of a concept	*Ale* is an exact match to *beer*
skos:broader	The broader concept of a concept	*Beer* has a broader concept of *alcohol*
skos:narrower	The narrower concept of a concept	*Alcohol* has a narrower concept of *beer*

SKOS is a Resource Description Framework (RDF) based vocabulary designed to represent onotologies, taxonomies, thesauri, among many other semantic graphs. Some semantic graphs are already SKOS formatted, where as the semantic relationships gathered from Wikiminer or the DBpedia and Freebase Annotators require formatting to the SKOS vocabulary before being added to the Sesame repository. Semantic relationships in RDF are denoted as *triples* – a sequence of three Uniform Resource Identifiers (URIs) formatted as `<vertex> <edge> <vertex>`. For example:

<http://en.wikipedia.org/wiki/Beer>
<skos:related>
<http://en.wikipedia.org/wiki/Brewing>

This triple denotes an hyperlink (or graph edge) existing between the two Wikipedia pages for Beer and Brewing.

5.4 FORMALISATION OF F-STEP

A brief description of each processing step has now been covered in Section 5.2, along with an outline of the SKOS vocabulary in Section 5.3 that all the semantic information of the taxonomy is formatted in. Now the F-STEP system can be formalised, with specific attention given to the disambiguation processing step as detailed in Algorithm 2 below.

Algorithm 2: Disambiguating Concepts that Originate from Heterogeneous Semantic Graphs

Input: D
Output: T
`SetUpRepositories` (); `// Step 1`
$R_d \leftarrow$ `IndexDocuments` (D);
$R_t \leftarrow$ `AddPingarEntityExtractions` (); `// Step 2(a)`
$R_t \leftarrow$ `AddWikifyExtractions` (); `// Step 2(b)`
$R_t \leftarrow$ `AddDBpediaAnnotations` (); `// Step 3`
$R_t \leftarrow$ `AddFreebaseAnnotations` ();
`/* ---------- Step 4 Function Expanded (Start) */`
`DisambiguateHeterogeneity` () {
foreach $d \in R_d$ **do**
  foreach η *identified in* d **do** `// Phase 1: Get Canonical Concept`
    $C \leftarrow$ `GetConceptsMappedToNgram` (η);
    $\hat{c}_* \leftarrow \arg\max_{c \in C}$ `GetMeanSimilarity`$(\omega(c), \omega(\eta))$;
    $R_t \leftarrow$ `AddSKOSExactMatch` ($\hat{c}_*$,η);
    foreach $c \in C$ *such that* $c \neq \hat{c}_*$ **do** `// Phase 2: Merge/Discard`
      $\gamma \leftarrow$ `GetMeanSimilarity` ($\omega(c)$,$\omega(\hat{c}_*)$);
      if $\gamma > 0.9$ **then**
        $R_t \leftarrow$ `AddSKOSExactMatch` (c,η);
      else if $0.7 < \gamma \leq 0.9$ **then**
        $R_t \leftarrow$ `AddSKOSCloseMatch` (c,η);
      else
        $R_t \leftarrow$ `DiscardConcept` (c,η);
}
`/* ---------- Step 4 Function Expanded (End) */`
$R_t \leftarrow$ `ConsolidateTaxonomy` (); `// Step 5`
$T \leftarrow$ `ExportTaxonomy` ();

5.4.1 *System Input, Output, & Resources*

The collection of documents D is taken as input, for which the taxonomy T, is output of the F-STEP system. The Solr repository that stores the *documents* is denoted as R_d, while the Sesame repository that stores semantic relationships as *triples* is denoted as R_t. All functions have access to and can manipulate either of these two repositories. For the semantic graphs Wikipedia, DBpedia, Freebase, or any other relevant taxonomies, functions can either remotely access them by their SPARQL endpoints, or locally access them in R_t (i.e. in other words, a subset of relevant nodes and edges from a semantic graph can be locally cached in R_t).

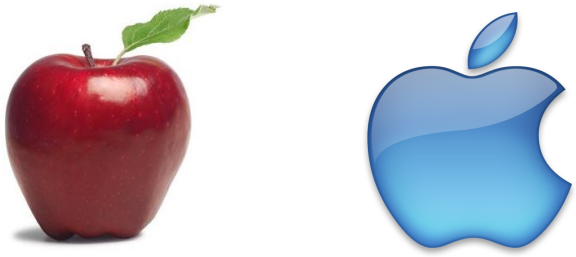
5.4.2 *Leading up to Disambiguation*

First is *initialisation* (Step 1) as described in Section 5.2.1, in which both repositories R_d and R_t are set up. Following this, the document collection, D, is uploaded to Solr via the function `IndexDocuments(D)`.

Next is *extraction* and *annotation* (Steps 2(a), 2(b), and 3) as described in Section 5.2.2. The extraction functions `AddPingarEntityExtractions()` and `AddWikifyExtractions()` map *concepts* and *named entities* from input taxonomies and Wikipedia to *ngrams* in the documents. Then the annotation functions `AddDBpediaAnnotations()` and `AddFreebaseAnnotations()` look for *named entities* mapped to *ngrams* by `AddPingarEntityExtractions()` in DBpedia and Freebase, adding mappings for these *named entities* if they can be found.

5.4.3 *Concept URI Mappings*

At this point all the *ngrams* in each document have several *concepts* and *named entities* mapped to them that each originate from a different semantic graph. From (Medelyan et al., 2013), take for example "Apple". The two



(a) (b)

Figure 18: (a) Apple the *fruit* and (b) Apple the *company*

concept URIs <http://www.freebase.com/view/en/apple> from Freebase and <http://en.wikipedia.org/wiki/Apple_Inc.> from Wikipedia as portrayed by Figure 18, could both be mapped to the the ngram "Apple". Naturally this is undesirable because they are distinct concepts (or senses) that while related, are not semantically equivalent.

This calls for a disambiguation processing step to resolve *concept* to *ngram* mappings that exhibit a semantic conflict. This is achieved with the function `DisambiguateHeterogeneity()` expanded as Step 4 in Algorithm 2, and to be explained in detail henceforth.

5.4.4 *Disambiguating Heterogeneity*

5.4.4.1 *Phase 1: Acquiring the Canonical Concept*

Each document d, is pulled down from the Solr repository R_d. For each ngram η, that is identified in document d, there are a set of concepts C that map to it. The requirement of the first phase is to find the concept (or sense) mapped to the ngram that best fits the context it used in. This is deemed to be the *canonical* concept and denoted as $\hat{c}_{i,*} \in C$. For this chosen concept a new URI for its canonical representation is generated and this is linked to its former concept URI with a skos:exactMatch semantic relation.

5.4.4.2 *Phase 2: Merging of Other Concepts with the Canonical Concept*

Phase 2 deals with merging or discarding the remaining competing concepts. Each remaining concept $c \in C$ such that $c \neq \hat{c}_{i,*}$ is now compared to the canonical concept using the function `GetMeanSimilarity(<Bag of Words#1>, <Bag of Words#2>)`. Again the function $\omega(\texttt{<ngram|concept>})$ is used to generate a bag of words context for the canonical concept $\hat{c}_{i,*}$ and for each concept c it is compared to. Depending on the mean similarity score γ, a competing concept c will be mapped as a skos:exactMatch semantic relation, a skos:closeMatch semantic relation, or discarded as a concept with a semantically different *sense*[11]. Taken from (Medelyan et al., 2013), the thresholds for each action taken are given in Table 12.

Table 12: Similarity Thresholds for Concept Merging

Mean Similarity Score (γ)	**Action Taken**
$\gamma \leq 0.7$	*Discard concept*
$0.7 < \gamma \leq 0.9$	*List as* skos:closeMatch
$\gamma > 0.9$	*List as* skos:exactMatch

The similarity thresholds values were manually selected upon reviewing output. Preferably if a large enough test data set could be prepared, more optimal values could be learned. This marks the end of the *disambiguation* processing step, which results in one canonical concept that represents a particular *sense* of the ngram it is mapped to, based on the particular *context* of the document it is found in. Each canonical concept may even be a merger of several concepts from different semantic graphs, which will naturally make for a much richer taxonomy to be output by the system.

11 To avoid conflict with the notation, the variable γ replaces the variable s in the paper (Medelyan et al., 2013), since s denotes a sense in this thesis.

5.4.5 *The End Result*

After the final processing step of *taxonomy consolidation* has taken place, as described in Section 5.2.4, the final taxonomy T can be exported from the Sesame Repository R_t in the SKOS vocabulary. In this format the taxonomy can be easily explored by a number of programs that can exploit SKOS semantic structures. Furthermore this format ensures the taxonomy is easily interropable to whatever purpose it may be used for on the semantic web.

5.4.6 *Details of Key Functions*

Now the implementation of two key functions ω`(<ngram|concept>)` and `GetMeanSimilarity(<Bag of Words#1>, <Bag of Words#2>)` found in Algorithm 2 will be detailed.

5.4.6.1 *Generating Bag of Words Context*

The purpose of the function ω`(<ngram|concept>)` is to build a *bag of words* context for either an *ngram* or a *concept*. To build a context for an ngram, this is achieved by collecting labels denoted as skos:prefLabel or skos:altLabel of the concepts that map to ngrams in the document[12]. On the other hand, to build a context for a concept (or named entity), this is achieved by collecting labels denoted as skos:prefLabel or skos:altLabel of adjacent concepts in the semantic graph it originates from. The edge types for these adjacent concepts, must be either skos:broader, skos:narrower, or skos:related. For example, a Wikipedia category page would be the equivalent to a skos:broader concept, with the page title and redirects being the equivalent to the skos:prefLabel and skos:altLabels respectively, both of which are included in the returned bag of words.

12 Admittedly this will include the labels of some incorrect concept mappings, however the detrimental effect of this is negligible since often a majority of mappings are correct.

5.4.6.2 *Calculating Mean Similarity*

Finding the intersection between a two bags of words is very reminiscent of the Lesk (1986) algorithm, with the difference that *local context* is matched with *concept labels* found in the relevant semantic graphs, rather than with *definitions* (or glosses) found in dictionaries. The author's implementation of this will now be detailed as the function `GetMeanSimilarity(<Bag of Words#1>, <Bag of Words#2>)`.

This function in its most basic form compares all the labels in the first bag with all the labels in the second bag, returning the total number of matches divided by the total number of comparisons made. Naturally some adjustments were required to address some of the obvious caveats that would affect the accuracy of this overly simplistic function.

Firstly, taking into account that documents vary in content and length, as well as the fact that semantic graphs vary in granularity and coverage, the set size of the bag of words context varies considerably. For example while some Wikipedia concepts have thousands of SKOS equivalent relations, taxonomies and thesauri such as the AGROVOC thesaurus[13] may only have a handful of SKOS relations in comparison. As a consequence, this will severely dilute the mean similarity, even if there are some very relevant label matches. To address this issue the variable $n = min\{|\omega(c_x)|, |\omega(c_y)|\}$ was devised. Now only the mean of the top n (rather than all) similarity scores, is returned by the function. The assumption is, if the two concepts the sets represent are semantically equivalent, then every label in the smaller set should have at least one reasonably similar partner label in the larger set.

Secondly, deviations in word etymology and orthography of concept labels, such as *beer* and *bier* or *neighbour* and *neighbor*, will result in a mismatch. However with the use of string similarity metrics, at least a partial match can be credited. There are several string similarity metrics, as listed on the following page.

13 http://aims.fao.org/agrovoc - The AGROVOC (Multilingual Agricultural) Thesaurus

1. Levenshtein (1966) Distance (LD)
2. Longest Common Subsequence (LCS)
3. Dice (1945) Coefficient (DC) – later known as the Sorensen (1948) index (SSI)

The author chose to use the string similarity metric SSI, denoted in Equation (19). This provides a value between 0 and1, that is sensitive to deviations in spelling at the lexical level.

$$\text{SSI} = \frac{2\,|A \cap B|}{|A| + |B|} \tag{19}$$

For example compare the alternative spellings of *neighbour* and *neighbor* in British and American English. By breaking down each string into a set of character pairs, such that $A = \{ne, ei, ig, gh, hb, bo, ou, ur\}$ and $B = \{ne, ei, ig, gh, hb, bo, or\}$, the intersection between these two sets can be illustrated as the Venn diagram in Figure 19. Based on this, the SSI value for the two alternative spellings can be calculated as seen in Equation (20) below.

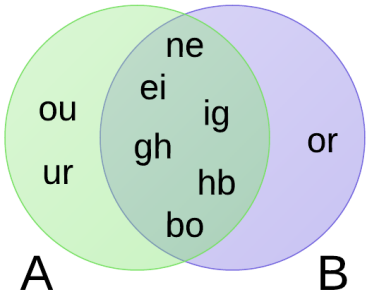


Figure 19: Venn Diagram of A, B, & $A \cap B$

$$\text{SSI} = \frac{2\,|A \cap B|}{|A| + |B|} = \frac{2 \times 6}{8 + 7} = \frac{12}{15} = 0.8 \tag{20}$$

The use of SSI increased the sensitivity of label matching to ensure the labels in the smaller set had the best chance of finding an equivalent in the larger set. Although of course, some matches included in the top n were not valid. For example as Kondrak (2005) also found in his experiments, a worse case scenario for SSI used with character pairs is that a similarity

value of 1.0 is possible for words such as "Xanex" and "Nexan" which have the same character pairs {an, ex, ne, xa} but are not semantically equivalent. Furthermore character pairs can be credited although they appear in very different positions of the words, for example "Voltaren" and "Tramadol". This occasionally led to a concept mapping being included in the final taxonomy when it should have been discarded by the disambiguation processing step, however the cut-off threshold of 0.7 ensured this did not happen often.

5.5 EXAMPLE FROM DISAMBIGUATION RESULTS

During the author's internship period, only an evaluation of the F-STEP system's performance as a whole was conducted. The results and discussion of this can be found in the paper (Medelyan et al., 2013) which is in Appendix B.1. Alternatively, provided here is an example result of the disambiguation module's implementation.

Consider for document d, the ngram η = "beer" is found in the sentence:

"Guests at the festival also got the chance to enjoy some locally sourced food to accompany the free beer."

Assume it has a concept c_w, from Wikipedia and a concept c_a, from the AGROVOC thesaurus, mapped to it by the extraction and annotation functions leading up to the disambiguation module. Then assume for Phase 1 of dismabiguation, it was decided that the canonical concept $\hat{c}_* = c_w$. For Phase 2 it now needs to be decided whether to merge or discard the other concept c_a. This illustrated by Table 13 over the following page.

The concepts $\hat{c}_*$ =<http://en.wikipedia.org/wiki/Beer> from Wikipedia and c_a =<http://aims.fao.org/aos/agrovoc/c_864> from the AGROVOC thesaurus are input for the function `GetMeanSimilarity`$(\omega(\hat{c}_*), \omega(c_a))$. In this instance, $\hat{c}_*$ has over 400 concept relations that are mostly skos:related, where as c_a has only 4 concept relations that are either skos:broader or skos:narrower.

Table 13: SSI Scores for Top n Concept Label Matches

$\omega(\hat{c}_*)$	$\omega(c_a)$	**SSI**
"Alcoholic Beverages"	"Alcoholic Beverages"	**1.000**
"Stout"	"Stouts"	**0.889**
"Lager"	"Lagers"	**0.889**
"Beer"†	"Beers"†	**0.857**
"Cask Ale"	"Ales"	0.500
"Pale Lagers"	"Lagers"	**0.769**
"Vienna Lager"	"Lagers"	0.615
⋮	⋮	⋮
"Saccharomyces cerevisiae"	"Stouts"	0.000
	Total	4.404
	Mean	0.881

Also worth noting is that the dagger† in Table 13 denotes the skos:prefLabel for each concept – "Beer" and "Beers" respectively, which are also included in the bag of words context. Recall that $n = \min\{|\omega(\hat{c}_*)|, |\omega(c_a)|\}$, therefore the mean of the top 5 label matches (shown in bold) will be returned by the function `GetMeanSimilarity`$(\omega(\hat{c}_*),\ \omega(c_a))$. Notice an almost perfect match was found for each of label from $\omega(c_a)$ except for "Ales", validating the author's earlier assumption that if concepts are semantically equivalent then there should be approximately n similar partners found in the larger labels set. Considering the similarity thresholds given in Table 12 and a calculated mean similarity of $\gamma = 0.881$ for the concepts $\hat{c}_*$ and c_a, they would be merged together as a skos:closeMatch.

A skos:exactMatch merger might have been more appropriate. However given that the smaller set of labels were mostly in the plural form whereas the larger set of labels were mostly in the singular form, at least the combination of using SSI with character pairs ensured some partial credit was awarded so the mapping of c_a to η was not discarded. Without this combination, there would have been only one string match (from label "Alcoholic Beverages"), resulting a much lower mean similarity of $\gamma = 1/5$.

In the final taxonomy T, locating the concept for *beer* will have $\hat{c}_*$ listed as an skos:exactMatch and c_a listed as a skos:closeMatch. Furthermore a majority the incorrect concept to ngram mappings should have been removed, effectively pruning away the noisy relations in taxonomy T. This underscores the primary purpose of this disambiguation module described in this chapter.

6

PERIPHERAL DIVERSITY

After completion of the internship at Pingar, the author decided to participate in the SemEval 2013 *task of Multilingual Word Sense Disambiguation*[1]*, for the languages English, French, German, Italian, and Spanish. This involved developing a new graph centrality measure to achieve Word Sense Disambiguation (WSD), dubbed as Peripheral Diversity (PD).*

In the results of this task (Navigli et al., 2013), Peripheral Diversity proved to be a competitive and robust graph centrality measure, managing to defeat the Most Frequent Sense (MFS) baseline for both French and Italian. The content of this chapter has been published as a peer reviewed paper (Manion and Sainudiin, 2013) which can be found in Appendix B.2. This paper was chosen for presentation and also for the poster session at SemEval 2013*, held in conjunction with* *SEM *and* NAACL *in Atlanta, Georgia.*

1 http://www.cs.york.ac.uk/semeval-2013/task12 - The Homepage for the Multilingual WSD task.

6.1 TASK DESCRIPTION

The SEMEVAL[2] 2013 All Words (AW) task of Multilingual Word Sense Disambiguation (WSD) included the languages English, French, German, Italian, and Spanish. The task focused on disambiguating nouns, and systems could be evaluated by their use of WordNet (Fellbaum, 1998), Wikipedia, or both via BabelNet (Navigli and Ponzetto, 2012a) which maps these Lexical Knowledge Bases (LKBs) together as a multilingual sense inventory[3].

6.2 BABEL SYNSETS

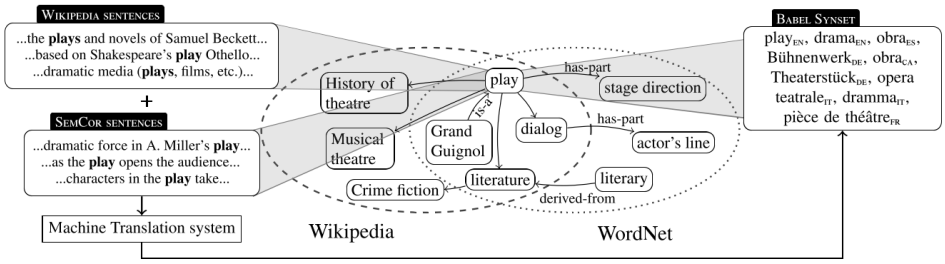


Figure 20: Illustrative View of a Babel Synset

Crucial to this task is understanding Babel synsets, of which Figure 20 presents an example taken from (Navigli and Ponzetto, 2012a) of the synset for the sense PLAY$_{(n)}$ *theatre*. Effectively a Babel synset represents a unified concept, through the mapping of senses found in the two aforementioned heterogeneous semantic graphs WordNet (as a lexicon) and Wikipedia (as a multilingual encyclopedia), making BabelNet a *multilingual encyclopedic dictionary* (Navigli and Ponzetto, 2012c). As a result, each synset is a collection of lexicalisations in several languages, in which its mappings to WordNet and Wikipedia senses are indicated by the keys WIKIWN (Wikipedia + WordNet), WN (WordNet only), and WIKI (Wikipedia only). As these keys

2 http://www.cs.york.ac.uk/semeval-2013/task12/index.php?id=task-description - Online Task Description of SEMEVAL 2013 Task 12: Multilingual Word Sense Disambiguation

3 *See* Section 1.4.2 for sense inventory descriptions

suggest, it is not always possible to produce a mapping since naturally a dictionary and encyclopedia can not be expected to completely overlap.

While the whole mapping process detailed in (Navigli and Ponzetto, 2012a, p220-224) will not be described here, there are some interesting parallels with Chapter 5 that also had to deal with heterogeneity between semantic graphs. Firstly, Navigli and Ponzetto (2012a) needed to create a common contextual representation for concepts from both semantic graphs in order to compare semantic equivalence. A WordNet context was defined by a sense's *synonyms*, *hypernymy/hyponymy*, and *glosses*, on the other hand, a Wikipedia context was defined by page *labels*, *links*, *redirects*, and *categories*. Both of these contexts can be transliterated into the SKOS vocabulary introduced in Table 11 in Section 5.3.

Secondly, the *abundancy* of context for Wikipedia is much greater than that of WordNet making them difficult to compare. This same issue had to be addressed in Section 5.4.6.2 when calculating the mean similarity. While this author would reduce the Wikipedia context with the variable n to match that of WordNet, Navigli and Ponzetto (2012a) instead increase the context of WordNet via graph based means to match that of Wikipedia[4]. While this is not the focus of this chapter, it is interesting to note that establishing a common context and a similar degree of context abundancy are two issues that need to be dealt with when dealing with heterogeneous semantic graphs.

6.3 CREATING SUBGRAPHS WITH BABELNET

For this task, subgraphs were constructed using the Daebak API, that works alongside classes from the BabelNet API to access BabelNet[5] and synset paths indexed into Apache Lucene[6] to ensure speed of subgraph construction. The reader should refer to Section 2.2.1 for formalisations of subgraph

4 Extra context from WordNet is discovered via a Depth First Search (DFS).
5 BabelNet 1.1.1 API, Sense Inventory, & Paths - http://babelnet.org/download
6 Apache Lucene - http://lucene.apache.org

construction, and to (Navigli and Ponzetto, 2012c) for better understanding of how to use the BabelNet API.

6.4 PERIPHERAL DIVERSITY

For this task, the author designed a graph based centrality measure ϕ, named "Peripheral Diversity" (PD). The intuition behind it is, given a subgraph $\mathcal{G}_\mathcal{L}$ that is created based on the local context of a word to be disambiguated, then the sense that is most appropriate and should be chosen by ϕ needs to be a) highly connected to b) a diverse range of peripheral senses, thus *peripheral diversity*. In line with these intuitions, senses that are highly connected to a very similar set of senses (e.g. is part of an isolated but strongly connected component) should be rejected. Again senses that are diverse but are not well connected to the rest of $\mathcal{G}_\mathcal{L}$ should also be rejected. It is hoped that PD can select senses on the appropriate region of this continuum, which is now formalised in the following section (Note that only ϕ is formalised, the rest of the WSD system is formalised the same way as in Section 2.1.2, with $\mathcal{G}_\mathcal{L}$ constructed as a subtree subgraph).

6.4.1 *Pairwise Semantic Dissimilarity*

First, for each synset $s_{i,j} \in R(\ell_i)$, a set of its peripheral synsets needs to be acquired. This is done by travelling a depth of up to d (stopping if the path ends), then adding the synset reached to the set of peripheral synsets $\mathcal{P}^{\leq d} = \{s_{j,1}, s_{j,2}, ..., s_{j,k}\}$. Next for every pair of synsets s and s' that are not direct neighbours in $\mathcal{P}^{\leq d}$ such that $s \neq s'$, their Pairwise Semantic Dissimilarity (PSD) $\delta(s, s')$ is calculated, which is required for a synset's PD score. To generate the results for this task the complement to Cosine

Similarity is used as the PSD measure. Commonly known as the Cosine Distance it is denoted in Equation (21) below.

$$\delta(s, s') = \begin{cases} 1 - \left(\frac{|O(s) \cap O(s')|}{\sqrt{|O(s)|}\sqrt{|O(s')|}} \right), & \text{if } |O(s)||O(s')| \neq 0 \\ 1, & \text{otherwise,} \end{cases} \tag{21}$$

$O(s)$ is the outgoing (out-neighbouring) synsets for $s \in \mathcal{P}^{\leq d}$, and $|O(s)|$ denotes the number of elements in $O(s)$.

6.4.2 *Peripheral Diversity Score*

Once the PSD scores for every permitted pairing of s and s' are calculated, there are a number of ways to generate the $\phi(s_{i,j})$ values. To generate results for this task, synsets were scored on the *sum of their minimum PSD values*, which is expressed formally below.

$$\phi(s_{i,j}) = \sum_{s \in \mathcal{P}^{\leq d}(s_{i,j})} \min_{\substack{s' \neq s \\ s' \in \mathcal{P}^{\leq d}(s_{i,j})}} \delta(s, s') \tag{22}$$

The idea is that this summing over the peripheral synsets in $\mathcal{P}^{\leq d}(s_{i,j})$ accounts for how frequently synset $s_{i,j}$ is used, in which then each *usage* is scaled by a peripheral synset's minimum PSD across all synsets in $\mathcal{P}^{\leq d}(s_{i,j})$. It is hoped that this can score highly the senses that are well connected to a diverse range of senses.

6.4.3 *Strategies, Parameters, & Filters*

WIKIPEDIA'S *did you mean?* Deviations and errors in spelling are accounted for to ensure lemmas have the best chance of being mapped to a synset. Absent synsets in subgraph $\mathcal{G}_{\mathcal{L}}$ will naturally degrade system out-

put. Therefore if $\ell_i \mapsto \varnothing$, then an HTTP call to Wikipedia's *Did you mean?* function was made and the response is parsed for any alternative spellings. For example in the test data set[7] the misspelt lemma: "feu_de_la_rampe" was corrected to "*feux*_de_la_rampe".

CUSTOM BACK-OFF STRATEGY The MFS is a very powerful back-off[8] strategy, yet it relies on having some quantity of hand-tagged data (McCarthy et al., 2004). Therefore a custom back-off strategy was designed. In the event the system provides a `null` result, the Babel synset $s_{i,j} \in R(\ell_i)$ with the most senses associated with it will be chosen with preference to its region in BabelNet such that WIKIWN $\succ$ WN $\succ$ WIKI. Note that as a participant in the task, the MFS and FS had not yet been published, which was the motivation to design an automated back-off strategy.

INPUT PARAMETERS The sliding context window length $(b - a)$ was set to encompass 5 sentences at a time, in which the step size was also 5 sentences. For subgraph construction the maximum path length[9] L was set to 2, with the peripheral search depth d set to 3. BabelNet edge weights, as deduced in (Navigli and Ponzetto, 2012a), do not directly contribute to PD scores in the experimental results that follow. BabelNet edge weights are only used to qualify each path in the construction of $\mathcal{G}_{\mathcal{L}}$, by ensuring that an edge weight of ≥ 0.005 is found for all edges in each contributing path[10]. Finally, all these parameters were set based on results obtained with the trial data issued by the organisers before the evaluation period of the task.

7 Found in sentence `d001.s002.t005` in the French test data set.

8 In the event the WSD technique fails to provide an answer, a back-off strategy provides one for the system to output.

9 In the SEMEVAL workshop proceedings it is reported that $L = 3$, however when the author reproduced the results at a later date it was discovered that in fact $L = 2$. This thesis and the most current version of the task paper on the author's homepage now reflect this to ensure results can be reproduced.

10 This value can be adjusted by changing the `babelnet.minEdgeWeight` variable of the `babelnet.properties` file located in the BabelNet API.

FILTERS Two filters were applied to the subgraphs that ship with the BabelNet API. WordNet contributed domain relations were removed with the `ILLEGAL_POINTERS` filter, and then the `SENSE_SHIFTS` filter (see Section 2.2.3) was applied. For more information on these filters, consult the BabelNet API documentation. Once again, these filters were applied since they demonstrated an improvement on performance for the trial data.

6.5 SEMEVAL RESULTS

6.5.1 *Results of SemEval Submission*

	Language	**DAEBAK!**	**$MFS_{Baseline}$**	**+/-**
DE	*German*	59.10	68.60	-9.50
EN	*English*	60.40	65.60	-5.20
ES	*Spanish*	60.00	64.40	-4.40
FR	*French*	53.80	50.10	+3.70
IT	*Italian*	61.30	57.20	+4.10
	Mean	58.92	61.18	-2.26

Table 14: DAEBAK! vs MFS Baseline on BabelNet

As can be seen in Table 14, the results of the single submission titled "DAEBAK!", were varied and competitive. The worst result was for German in which the system fell behind the MFS baseline by a margin of 9.50. Again for French and Italian the MFS baseline was exceeded by a margin of 3.70 and 4.10 respectively. The Daebak back-off strategy contributed anywhere between 1.12% (for French) to 2.70% (for Spanish) in the results, which means the system outputs a result without the need for a back-off strategy at least 97.30% of the time. Over all languages the system was slightly outperformed by the MFS baseline by a margin of 2.26. Ultimately PD demonstrated to be robust across a range of European languages. With these preliminary results this surely warranted further investigation of what can be achieved with PD.

In Figure 21 the three teams that entered this task had their system's performance evaluated over increasing degrees of polysemy. As expected the more polysemous words induce a decay in performance. Finally PD was also the top performing system for the disambiguation of named entities.

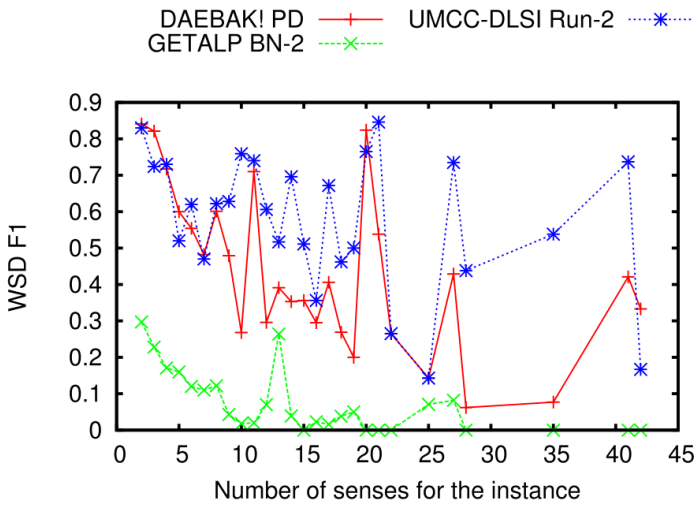


Figure 21: F-Score According to the degree of instance polysemy, reported when at least ten instances have the specified polysemy. (Navigli et al., 2013).

6.5.2 *Exploratory Results*

Some inconsistencies were observed in the task answer keys across different languages as shown in Table 15. For each Babel synset ID found in the answer key, the sense inventory it originated from was recorded, be it Wikipedia (WIKI), WordNet (WN), or both (WIKIWN).

	Language	**WIKI**	**WN**	**WIKIWN**
DE	*German*	43.42%	5.02%	51.55%
EN	*English*	10.36%	32.11%	57.53%
ES	*Spanish*	30.65%	5.40%	63.94%
FR	*French*	40.81%	6.55%	52.64%
IT	*Italian*	38.80%	7.33%	53.87%

Table 15: BabelNet Answer Key Breakdown

This is not a critical observation but rather an empirical enlightenment on the amount of development/translation effort for each language that has gone into the contributing subparts of BabelNet – Wikipedia and WordNet. As was the case in Chapter 5, the heterogeneity of hybrid sense inventories such as BabelNet creates new obstacles for WSD. In the future, disambiguation policies need will to be sensitive to the heterogeneity of such sense inventories.

7

ITERATIVE CONSTRUCTION OF SUBGRAPHS

After competing in the multilingual Word Sense Disambiguation (WSD) task of SemEval 2013, *experimentation with the Peripheral Diversity (PD) framework described in Chapter 6 continued, along with the testing and development of other graph centrality measures, context windows, filters, and subgraph types. All these experiments provided the Daebak Application Programming Interface (API) with many more features as well. After much experimentation, it became clear that PD had its limitations. Furthermore this also appeared to be the case for most other graph centrality measures, subgraph types, window sizes, and filters. Of course certain combinations of these variables proved better than others, but this apparent glass ceiling could not be broken through.*

Why not?

Through re-evaluation of the literature the author noticed that most researchers, just as this author had also done, tended to focus on developing, tweaking, and evaluating ϕ, $\mathcal{L}$, $\mathcal{G}_{\mathcal{L}}$, or other parts of the WSD process individually as ordered atomic steps of the overall process. Therefore this chapter details a shift in focus for the investigation, by stepping back and looking at the bigger picture of how these atomic steps could interact with each other.

Lastly, this chapter has been published as a peer reviewed long paper (Manion and Sainudiin, 2014) which can be found in Appendix B.3. It was presented in the proceedings of *SEM 2014, *held in conjunction with* COLING *and* SemEval *in Dublin, Ireland.*

7.1 THE CONVENTIONAL SUBGRAPH APPROACH

Subgraph-based WSD has been characterised over the last decade by performing the two key steps of (1) subgraph construction and (2) disambiguation via graph centrality measures, in an ordered atomic sequence. This author refers to this characteristic as the *conventional* approach to subgraph-based WSD, while at the same time, proposes an *iterative* approach to subgraph-based WSD that allows for interaction between the two atomic steps in an incremental manner. This chapter will demonstrate its effectiveness across a range of graph-based centrality measures and subgraph construction methods, at both the sentence and document level.

7.1.1 *Algorithm for Conventional Approach*

The two steps of subgraph construction and disambiguation via a graph centrality measure have already been formalised in Chapter 2 in Section 2.2.1 and Section 2.2.2 respectively. In reference to these sections, the conventional subgraph approach can be illustrated by Algorithm 3. Let $\mathcal{L}$ be taken as *input*, and let the disambiguation results $\mathcal{D} = \{\hat{s}_{1,*}, ..., \hat{s}_{m,*}\}$ be produced as *output* to assign to $\mathcal{L} = \{\ell_1, ..., \ell_m\}$.

Algorithm 3: Conventional Approach

Input: $\mathcal{L}$
Output: $\mathcal{D}$
$\mathcal{D} \leftarrow \varnothing$;
$\mathcal{G}_{\mathcal{L}} \leftarrow$ `ConstructSubGraph` $(\mathcal{L})$;
foreach $\ell_i \in \mathcal{L}$ **do**
  $\hat{s}_{i,*} \leftarrow \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$;
  put $\hat{s}_{i,*}$ in $\mathcal{D}$;

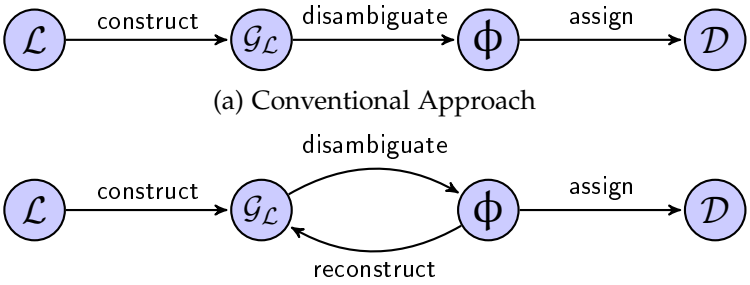
To begin with, $\mathcal{D}$ is initialised as an empty set and `ConstructSubGraph`$(\mathcal{L})$ constructs one of the three subgraphs described in Section 2.2.1. Next for each $\ell_i \in \mathcal{L}$, by running a graph based centrality measure ϕ over $\mathcal{G}_{\mathcal{L}}$, the most appropriate sense $\hat{s}_{i,*}$ is estimated, and placed in set $\mathcal{D}$. Effectively,

$\mathcal{L}$ is a context window based on document or sentence size, therefore this algorithm is run for each context window division. Note that Algorithm 3 would require a little extra complexity to handle *local edge* subgraphs (as described in Section 2.2.1(c)), due to its context window needing to satisfy $\mathcal{L} = \{\ell_{i-D}, ..., \ell_{i+D}\}$.

7.2 THE ITERATIVE SUBGRAPH APPROACH

7.2.1 *What is Iterative WSD?*

As alluded to earlier, the key observation to make about the conventional approach in Algorithm 3, is for input $\mathcal{L}$, constructing subgraph $\mathcal{G}_{\mathcal{L}}$ and performing disambiguation are two ordered atomic steps. Notice that there is no iteration between them, because the first step of subgraph construction is never revisited for each $\mathcal{L}$. For the conventional process to be iterative, then for $\ell_a, \ell_b \in \mathcal{L}$ a previous disambiguation of ℓ_a, would need to influence a consecutive disambiguation of ℓ_b, through an iterative re-construction of $\mathcal{G}_{\mathcal{L}}$ between each disambiguation. This key difference illustrated by Figure 22, is the level of iterative WSD that is aspired to.



(a) Conventional Approach

(b) Interactively Iterative Approach

Figure 22: The Key Difference In Approach

It is important to note, the term *iterative* can already be found in WSD literature, therefore the opportunity is taken here to make a distinction. Firstly, a graph based centrality measure ϕ may be iterative, such as PageRank (Brin and Page, 1998) or HITS (Kleinberg, 1999). In the experiments by

Mihalcea (2005) in which PageRank was run over *local edge* subgraphs, it is easy to perceive the WSD process itself as iterative.

Iteration can again be taken further, as observed with Personalised PageRank in which Agirre and Soroa (2009) apply the idea of biasing values in the random surfing vector, v, (see Haveliwala (2003)). For their run labelled "Ppr_w2w", in order to avoid senses anchored to the same lemma reinforcing each other's ϕ score, the random surfing vector v is iteratively updated as ℓ_i changes, to ensure context senses $s_{a,j} \in v$ such that $a \neq i$ are the only senses that receive probability mass[1].

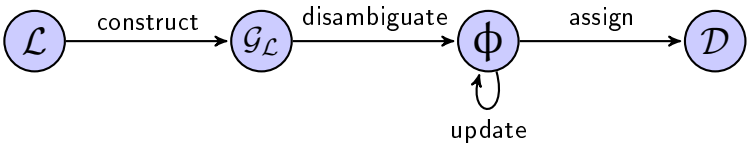


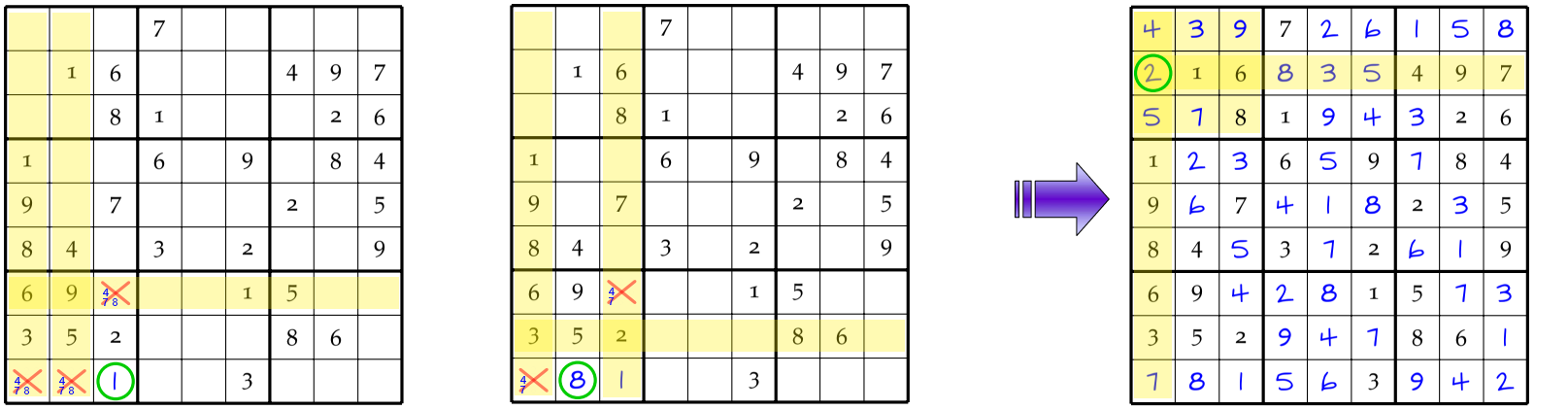
Figure 23: Atomically Iterative Approach

In summary, iteration in the literature either describes ϕ as being iterative or being iteratively adjusted, both of which are contained in the disambiguation step alone as shown in Figure 23. This is iteration at the atomic level and should not be conflated with the interactive level of iteration that is proposed as seen in Figure 22 (b).

7.2.2 *Iteratively Solving a Sudoku Grid*

In Figures 24 (a), (b), and (c) on the following page, the solving of a Sudoku puzzle can be observed, in which the numbers from 1 to 9 must be assigned only once to each *column*, *row*, and 3x3 *square*. Each time a number is assigned and the Sudoku grid is updated, this is an *iteration*. For example, in the south west square of grid (a) (i.e. Figure 24 (a)) unknown cells can be assigned $\{1,4,7,8\}$. Given that 1 has already been assigned to the 7^{th} row and the 1^{st} and 2^{nd} columns, this singles it down to one cell it can be assigned to (at the intersection of the 3^{rd} column and 9^{th} row).

1 This has a similar purpose to the SENSE_SHIFT filter described in Section 2.2.3



(a) 1st Row/Column Elimination

(b) 2nd Row/Column Elimination

(c) Row/Column/Box Completion

Figure 24: Iterative Solving of Sudoku Grids

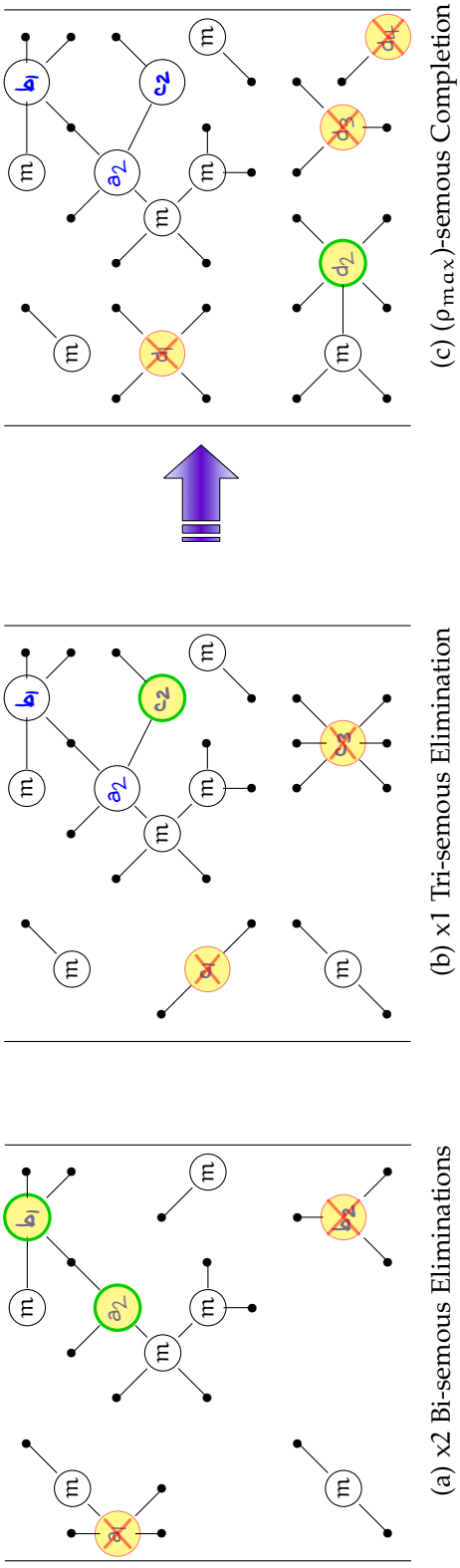
The iteration of grid (a), now makes possible the iteration of grid (b) to eliminate the number 8 as the only possibility for its assigned cell. This iterative process continues until the completed puzzle in grid (c) is reached. Therefore in WSD terminology, with each cell *disambiguated*, a new grid is *constructed*, in which knowledge is passed on to each consecutive iteration.

Continuing with this line of thought, each unsolved cell is *ambiguous*, with a degree of *polysemy* ρ, such that $\rho_{max} \leq 9$. Again, the initial Sudoku grid has pre-solved cells, of which are *monosemous*. This leads to another key observation. Typically in Sudoku, it is necessary to solve the least polysemous cells first, before you can solve the more polysemous cells with a certainty. As the conventional approach exhibits no Sudoku-like iteration, cells are solved without regard to the ρ value of the cell, or any interactive exploitation of previously solved cells.

7.2.3 *Iteratively Constructing a Subgraph*

In the author's 'Sudoku style' approach, it is proposed that each ℓ_i should be disambiguated in order of increasing polysemy ρ, iteratively reconstructing subgraph $\mathcal{G}_\mathcal{L}$ to reflect 1) previous disambiguations and 2) the ρ value of lemmas being disambiguated in the current iteration. This is illustrated in Figures 25 (a), (b), and (c) on the following page.

Let m-labelled vertices describe monosemous lemmas. In subgraph (a) (i.e. Figure 25) the two bi-semous lemmas *a* and *b* can be observed, in which the arbitrary graph-based centrality measure ϕ has selected the second sense of *a* (i.e. a_2) and the first sense of *b* (i.e. b_1) to be placed in $\mathcal{D}$. For the next iteration, notice the alternative senses for *a* and *b* are removed from $\mathcal{G}_\mathcal{L}$ for the disambiguation of tri-semous lemma *c*. The second sense of lemma *c* manages to be selected by ϕ with the help of the previous disambiguation of lemma *a*. This interactive and iterative process continues until the most polysemous lemma is reached, which in this example is lemma *d* with $\rho_{max} = 4$ in subgraph (c).



(a) x2 Bi-semous Eliminations

(b) x1 Tri-semous Elimination

(c) (ρ_{max})-semous Completion

Figure 25: Iterative Disambiguating of Subgraphs

7.2.4 *Algorithm for Iterative Approach*

What is happening in Figure 25 can formally be described with Algorithm 4. Effectively, this is a recreation of Algorithm 3 with the necessary modifications made to upgrade the WSD approach from *conventional* to *iterative*.

Algorithm 4: Iterative Approach

Input: $\mathcal{L}$
Output: $\mathcal{D}$
$\mathcal{D} \leftarrow$ `GetMonosemous` $(\mathcal{L})$;
$\mathcal{A} \leftarrow \varnothing$;
for $\rho \leftarrow 2$ **to** ρ_{max} **do**
  $\mathcal{A} \leftarrow$ `AddPolysemous` $(\mathcal{L}, \rho)$;
  $\mathcal{G}_{\mathcal{L}} \leftarrow$ `ConstructSubGraph` $(\mathcal{A},\mathcal{D})$;
  foreach $\ell_i \in \mathcal{A}$ **do**
    $\hat{s}_{i,*} \leftarrow \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$;
    if $\hat{s}_{i,*}$ *exists* **then**
      remove ℓ_i from $\mathcal{A}$;
      put $\hat{s}_{i,*}$ in $\mathcal{D}$;

Firstly as it reads, `GetMonosemous`$(\mathcal{L})$ places all the senses of the monosemous lemmas into $\mathcal{D}$. This is the equivalent of copying out an unsolved Sudoku grid onto a piece of paper and adding in all the initial hint numbers. Next the set $\mathcal{A}$ which holds all *ambiguous* lemmas of polysemy $\leq \rho$ is initialised as an empty set. Now the values of ρ can be iterated through, beginning with $\rho = 2$ to add all bi-semous lemmas to $\mathcal{A}$ with the function `AddPolysemous`$(\mathcal{L}, \rho)$. Notice ρ places a restriction on the degree of polysemy a lemma $\ell_i \in \mathcal{L}$ can have before being added to $\mathcal{A}$.

Next with the function `ConstructSubGraph`$(\mathcal{A}, \mathcal{D})$ the first subgraph $\mathcal{G}_{\mathcal{L}}$ can be constructed. This previously used function in Algorithm 3, is now modified to take the ambiguous lemmas of polysemy $\leq \rho$ in set $\mathcal{A}$ and previously disambiguated lemma senses in set $\mathcal{D}$. The resulting graph has a limited degree of polysemy and is constructed based on previous disambiguations.

From this point on the given graph centrality measure ϕ is run over $\mathcal{G}_{\mathcal{L}}$. For the lemmas that are disambiguated, they are removed from $\mathcal{A}$ and the selected sense is added to $\mathcal{D}$. For those lemmas that are not (i.e. $\hat{s}_{i,*}$ does not exist[2]) they remain in $\mathcal{A}$ to be involved in reattempted disambiguations in consecutive iterations. As more lemmas are disambiguated, it is more likely that previously difficult to disambiguate lemmas become much easier to solve, just like at the end of a Sudoku puzzle it gets easier as you get closer to completing it.

7.3 EXPERIMENTAL RESULTS

These experiments set out to understand a number of aspects. The first experiment is a *proof of concept*, to understand whether an iterative approach to subgraph WSD can in fact achieve better performance than the conventional approach. The second set of experiments seeks to understand how the iterative approach works and the performance *benefits* and *penalties* of implementing the iterative approach. Finally the third experiment is an *elementary attempt* at optimising the iterative approach to defeat the Most Frequent Sense (MFS) baseline.

7.3.1 *LKB & Dataset for All Experiments*

All experiments are conducted on the most recent SEMEVAL WSD dataset, of which is the SEMEVAL 2013 Task 12 Multilingual WSD (English) data set described in Section 6.1. Once again the multilingual sense inventory known as BabelNet (Navigli and Ponzetto, 2012a) is used along with the Daebak API. Also the same filters were applied as described in the experiments of Chapter 6, along with BabelNet edge weights only being used to qualify an indexed path (such that `babelnet.minEdgeWeight`= 0.005).

2 This can happen if ℓ_i does not map to any senses, or alternatively all the senses that are mapped to are filtered out of the subgraph before disambiguation.

7.3.2 *Experiment 1: Proof of Concept*

7.3.2.1 *Experiment 1: Setup*

For this experiment the outset was simply to see how the iterative approach performed compared to the conventional approach in a range of experimental conditions. Directed and unweighted subgraphs were used, namely *subtree* and *shortest* paths subgraphs with L = 2. Disambiguation was attempted at the document and sentence level. For the graph centrality measures evaluated, none were optimised to avoid masking the total effect of the iterative approach. For this reason, while all the graph centrality measures formalised in Section 2.2.2 were applied to $\mathcal{G}_{\mathcal{L}}$, Personalised PageRank (Agirre and Soroa, 2009) was not. This is because its bias random surfing vector is considered to be an optimisation. As for traditional PageRank, it was allowable since it takes on a uniform random surfing vector. Also default values[3] of 0.85 and 30 for damping factor and maximum iterations were set respectively.

7.3.2.2 *Experiment 1: Observations*

First and foremost, it is clear from Table 16 and 17 on the following two pages that the iterative approach outperforms the conventional approach, regardless of the subgraph used, level of disambiguation, or the graph centrality measure employed. Since no graph centrality measure or subgraph type was optimised, let this experiment prove that the iterative approach has the potential to improve any WSD system that implements it.

At the document level for both subgraphs the F-Scores were very close to the MFS baseline for this task of 66.50. It is notoriously hard to beat and only one team (Gutiérrez et al., 2013) managed to beat it for this task.

3 The default values are the same as applied by the original PageRank paper (Brin and Page, 1998) and in the Personalised PageRank paper (Agirre and Soroa, 2009) for comparative purposes.

$\mathcal{G}_{\mathcal{L}}$	ϕ	Conventional Doc			Iterative Doc			Improvement		
		P	R	F	P	R	F	ΔP	ΔR	ΔF
SubTree Paths	In-Degree	**61.70**	**55.51**	**58.44**	**65.39**	**63.74**	**64.55**	+3.69	+8.23	+6.11
	Out-Degree	54.23	48.78	51.36	57.70	56.23	56.96	+3.47	+7.45	+5.59
	Betweenness Centrality	59.29	53.34	56.15	63.43	61.82	62.61	+4.14	+8.48	+6.46
	Sum Inverse Path Length	56.58	50.90	53.59	58.86	57.37	58.11	+2.28	+6.47	+4.51
	HITS(hub)	54.69	49.20	51.80	59.71	58.20	58.95	**+5.03**	**+9.00**	**+7.15**
	HITS(authority)	57.45	51.68	54.41	61.62	60.06	60.83	+4.18	+8.38	+6.42
	PageRank	60.09	54.06	56.92	64.07	62.44	63.24	+3.97	+8.38	+6.33
Shortest Paths	In-Degree	**63.06**	**56.08**	**59.36**	65.36	63.06	64.19	+2.30	+6.98	+4.83
	Out-Degree	57.07	50.75	53.72	61.14	58.92	60.01	+4.07	+8.17	+6.29
	Betweenness Centrality	60.33	53.65	56.79	**65.52**	**63.22**	**64.35**	**+5.20**	**+9.57**	**+7.56**
	Sum Inverse Path Length	57.53	51.16	54.16	61.19	58.98	60.06	+3.66	+7.81	+5.90
	HITS(hub)	57.48	51.11	54.11	62.14	59.96	61.03	+4.67	+8.85	+6.92
	HITS(authority)	60.91	54.16	57.34	63.54	61.30	62.40	+2.63	+7.14	+5.06
	PageRank	60.33	53.65	56.79	64.83	62.55	63.67	+4.50	+8.90	+6.87

Table 16: Improvements of using the Iterative Approach at the Document Level

$\mathcal{G}_{\mathcal{L}}$	ϕ	Conventional Sent			Iterative Sent			Improvement		
		P	R	F	P	R	F	ΔP	ΔR	ΔF
SubTree Paths	In-Degree	**60.83**	**50.70**	**55.30**	**61.80**	**56.23**	**58.88**	+0.96	+5.54	+3.58
	Out-Degree	56.18	46.82	51.07	59.64	54.11	56.74	+3.46	+7.29	+5.67
	Betweenness Centrality	59.40	49.51	54.01	61.66	56.08	58.74	+2.26	+6.57	+4.73
	Sum Inverse Path Length	56.67	47.23	51.52	59.45	54.01	56.60	+2.78	+6.78	+5.08
	HITS(hub)	55.49	46.25	50.45	59.51	54.06	56.65	**+4.02**	**+7.81**	**+6.20**
	HITS(authority)	56.80	47.34	51.64	60.30	54.84	57.44	+3.50	+7.50	+5.80
	PageRank	59.71	49.77	54.29	60.56	55.04	57.67	+0.84	+5.28	+3.38
Shortest Paths	In-Degree	**58.13**	**32.75**	**41.89**	63.79	42.11	50.73	+5.66	+9.36	+8.84
	Out-Degree	54.64	30.78	39.38	61.79	40.66	49.05	**+7.15**	+9.88	+9.67
	Betweenness Centrality	57.94	32.64	41.76	**64.11**	**42.32**	**50.98**	+6.17	+9.68	+9.22
	Sum Inverse Path Length	55.65	31.35	40.11	62.39	41.02	49.50	+6.74	+9.67	+9.39
	HITS(hub)	56.11	31.61	40.44	62.74	41.28	49.80	+6.63	+9.67	+9.36
	HITS(authority)	55.74	31.40	40.17	62.75	41.39	49.88	+7.01	**+9.98**	**+9.70**
	PageRank	56.84	32.02	40.97	63.17	41.70	50.23	+6.33	+9.67	+9.27

Table 17: Improvements of using the Iterative Approach at the Sentence Level

For all subtree subgraphs, it was observed that In-Degree is clearly the best choice of centrality measure, while HITS (hub) enjoys the most improvement. It was also observed that applying the iterative approach to Betweenness Centrality on shortest paths is a great combination at both the document and sentence level, most probably due to the measure being based on shortest paths. Furthermore it is worth noting, the results at the sentence level for all graph centrality measures on shortest path subgraphs are quite poor, but highly improved, this is likely due to the restriction of L = 2 causing the subgraphs to be much sparser and broken up into many components.

Also provided here is an example from the data set in which the incorrect disambiguation of the lemma *cup* via the conventional approach was corrected by the iterative approach. This example is the seventh sentence in the eleventh document (`d011.s007`). Each word's degree of polysemy is denoted in square brackets.

> "Spanish **[1]football players** playing in the All-Star **[4]League** and in powerful **[12]clubs** of the **[2]Premier League** of **[9]England** are during the **[5]year** very active in **[4]league** and local **[8]cup** **[7]competitions** and there are high-level **[25]shocks** in the **[10]European Cups** and **[2]European Champions League**."

The potential graph constructed from this sentence is illustrated in Figure 26 over the following page as a shortest paths subgraph. The darker edges portray the subgraph iteratively constructed up to a polysemy $\rho \leq 8$ (in order to disambiguate *cup*), whereas the lighter edges portray the greater subgraph constructed if the conventional approach is employed.

Note that although the lemma *cup* has eight senses, only three are shown due to the application of the previously mentioned `SENSE_SHIFTS` filter. The remaining five senses of *cup* were filtered out since they were not able to link to a sense up to L = 2 hops away that is anchored to an alterative lemma.

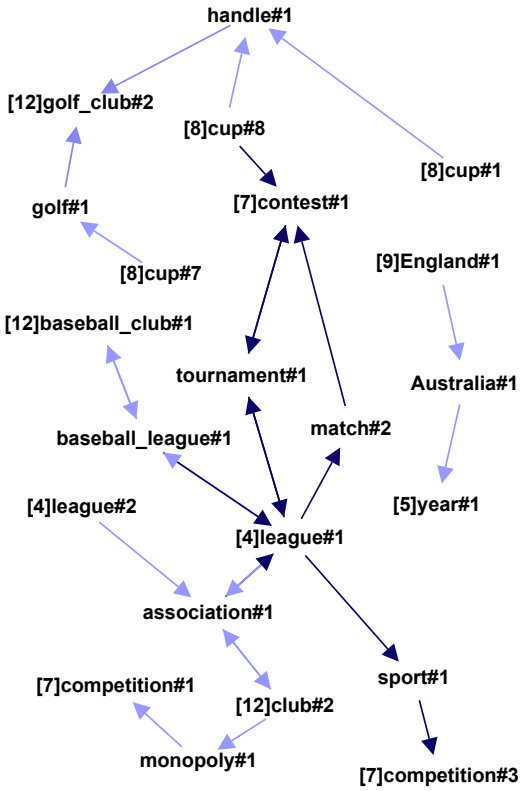


Figure 26: Conventional vs Iterative Subgraph Example

- **cup#1** - A small open container usually used for drinking; usually has a handle.
- **cup#7** - The hole (or metal container in the hole) on a golf green.
- **cup#8** - A large metal vessel with two handles that is awarded as a trophy to the winner of a competition.

Given the context, the 8th sense of cup is the correct sense, the type known as a trophy. For the conventional approach, if ϕ is the centrality measure of Out-Degree then the 8th sense of cup is easily chosen by having one extra outgoing edge than the other two senses for *cup*. Yet if ϕ is the centrality measure of In-Degree or Betweenness Centrality, all three senses of cup now have the same score, zero. Therefore in the results the first sense is chosen which is incorrect. On the other hand, if the subgraph was constructed iteratively with disambiguation results providing feedback to

consecutive constructions, this could have been avoided. The shortest paths **cup#1→handle#1→golf_club#2** and **cup#7→golf#1→golf_club#2** only exist because the sense **golf_club#2** (anchored to the more polysemous lemma *club*) is present, if it was not then the `SENSE_SHIFTS` filter would have removed these alternative senses. This demonstrates that if the senses of more polysemous lemmas are introduced into the subgraph too soon, they can interfere rather than help with disambiguation.

Secondly with each disambiguation at lower levels of polysemy, a more stable context is constructed to perform the disambiguation of much more polysemous lemmas later. Therefore in Figure 26 an iteratively constructed subgraph with *cup* already disambiguated, would mean the other two senses of *cup* would no longer be present. This ensures that **club#2** (the correct answer) would have a much stronger chance of being selected than **golf_club#2**, which would have only one incoming edge from **handle#1**. Note the conventional approach would lend **golf_club#2** one extra incoming edge than **club#2** has, which could be problematic if ϕ is the centrality measure of In-Degree.

7.3.3 *Experiment 2: Performance of the Iterative Approach*

7.3.3.1 *Experiment 2: Setup*

An obvious caveat of the iterative approach is that it requires the construction of several subgraphs as ρ increases, which of course will require extra computation and time which is a penalty for the improved precision and recall. It was decided the extent to which this happens should be investigated. Betweenness Centrality and PageRank were selected from Experiment 1, in which both use shortest path subgraphs at the document level. This is because a) they acquired good results at the document level and b) with only 13 documents there are less data points on the plots making it easier to read as opposed to the hundreds of data points produced by sentences.

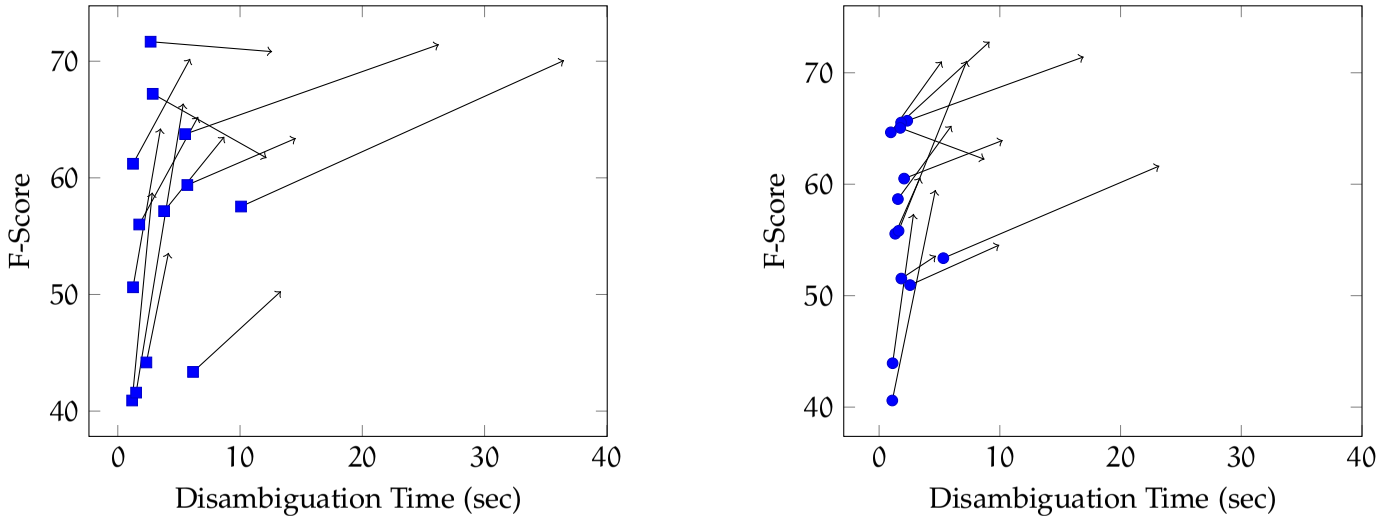
7.3.3.2 *Experiment 2: Observations*

Firstly from Figures 27 (a) and (b) on the next page, it can be seen that there is a substantial improvement in F-Score for almost all documents, with the exception of two for ϕ = Betweenness Centrality and one for ϕ = PageRank. For most documents the increased amount of time to disambiguate is not excessive. For this experiment, applying the iterative approach to Betweenness Centrality resulted in a mean 231% increase in processing time, from 3.54 to 11.73 seconds to acquire a mean F-Score improvement of +8.85. Again for PageRank, a mean increase of 343% in processing time, from 1.95 to 8.64 seconds to acquire a F-Score improvement of +7.16 was observed.

It was also investigated why in some cases, the iterative approach can produce poorer results than the conventional approach. Aspects such as order, size, density, and number of components were looked at for subgraphs that produced a correct, incorrect, or no disambiguation result. One aspect that stood out, was that a higher number of monosemous lemmas associated with the initial subgraph construction led to better disambiguation results. From this it was suspected from that, just like in a Sudoku puzzle, if there are not enough hints to start with, the possibility of finishing the puzzle successfully becomes slim. In other words, even though monosemous lemmas are by nature *salient*, if they were not *abundant* enough[4] in the construction of the initial $\mathcal{G}_{\mathcal{L}}$, then the effectiveness of the iterative approach could be negated.

On observing Figures 28 (a) and (b) over the following two pages, evidently monosemy does effect the outcome of the iterative approach. On the horizontal axis, document monosemy represents the percentage of lemmas in a document, not counting duplicates, that are monosemous. The vertical axis on the other hand represents the difference in F-Score between conventional and iterative approach. Through a simple linear regression of each scatter plot, an increased effectiveness of the iterative approach is observed with each slope of regression.

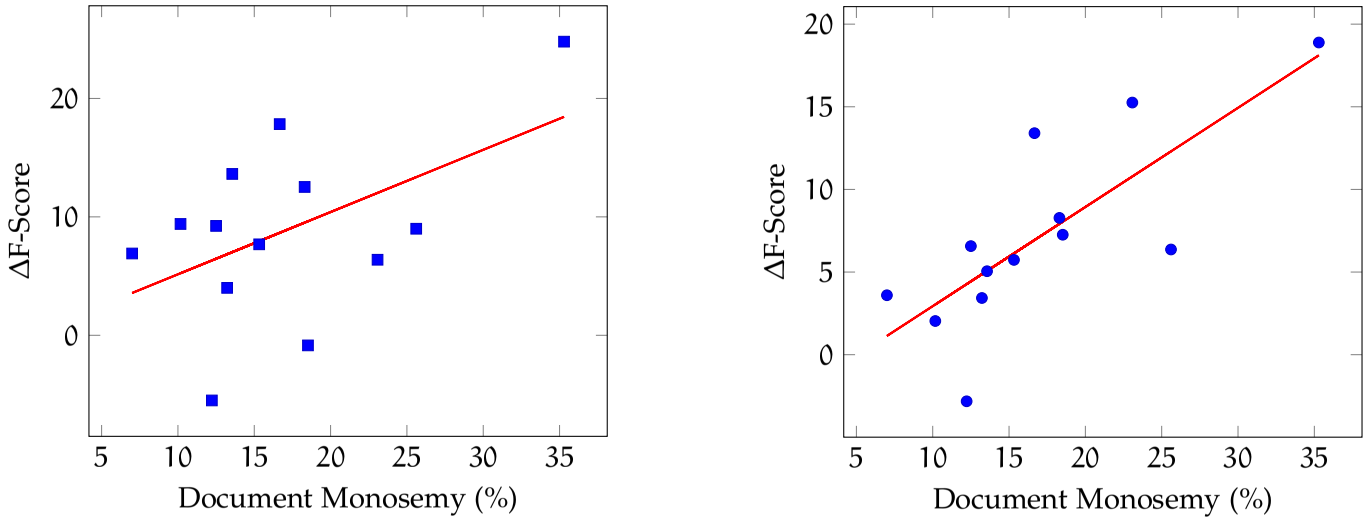
4 See context under-specification in Section 1.2.2 for an elaboration on context *abundancy* and *saliency*.



(a) ϕ = Betweenness Centrality

(b) ϕ = PageRank

Figure 27: For each of the 13 documents, performance (F-Score) is plotted against time to disambiguate, for $\mathcal{G}_{\mathcal{L}}$ = Shortest Paths. The squares (PageRank) and circles (Betweenness Centrality) plot the conventional approach. The arrows show the effect caused by applying the iterative approach, with the arrow head marking the *shift* in F-Score and time to disambiguate.



(a) ϕ = Betweenness Centrality

(b) ϕ = PageRank

Figure 28: Both PageRank (squares) and Betweenness Centrality (circles) are plotted. Each data plot represents the *change* in F-Score when the iterative approach replaces the conventional approach with respect to the monosemy of the document.

With m representing document monosemy, and ΔF representing the change in F-Score induced by the iterative approach, the slopes observed in Figures 28 (a) and (b) are denoted by Equations (23) and (24) respectively.

$$\Delta F = 0.53m - 0.11 \tag{23}$$

$$\Delta F = 0.60m - 3.07 \tag{24}$$

By taking document monosemy into account, these functions could make for a useful cut-off mechanism in terms of deciding whether to use the conventional or iterative approach for each document.

Finally, a Sudoku puzzle is a deterministic problem. The rules of Sudoku dictate which cells/rows/columns the numbers 1 to 9 can be allocated to, and they are therefore the constraints of the problem. Furthermore a Sudoku grid needs at least 17 hint cells to ensure it has one *unique* solution (and qualify as a valid puzzle)[5]. In contrast to Sudoku, when performing WSD a panel of human judges in disagreement with each other will find there are *multiple* disambiguation solutions. In spite of this fundamental difference between Sudoku and WSD, experimental results so far have shown that treating WSD as a deterministic problem is advantageous.

Nonetheless the stricter constraints of a Sudoku puzzle, means that just one misallocation of a number to a cell will lead to further misallocations and ultimately an incorrect final solution. Therefore in drawing parallels to Sudoku, is the iterative approach also susceptible to this by performing WSD in a deterministic manner? If a sense is incorrectly allocated to a lemma, can this lead to further sense misallocations in later iterations?

To see if this actually happens, the author deliberately allocated incorrect senses to all lemmas with a polysemy of $\rho \leq t$, then observed the difference in F-Score calculated for the disambiguation of all lemmas with a polysemy of $\rho > t$. By increasing the value of t, more misallocations of senses to lem-

5 Latest research maintains and proves by brute-force that at least 17 hint cells are required for a Sudoku grid to have a unique solution, see (McGuire et al., 2012).

mas occur in the initial iterations. The implementation of this is formally described in Algorithm 5 below.

Algorithm 5: Misallocated Iterative Approach

Input: $\mathcal{L}$
Output: $\mathcal{D}$
$\mathcal{D} \leftarrow$ `GetMonosemous` $(\mathcal{L})$;
$\mathcal{A} \leftarrow \varnothing$;
for $\rho \leftarrow 2$ **to** ρ_{max} **do**
  $\mathcal{A} \leftarrow$ `AddPolysemous` $(\mathcal{L}, \rho)$;
  $\mathcal{G}_{\mathcal{L}} \leftarrow$ `ConstructSubGraph` $(\mathcal{A},\mathcal{D})$;
  foreach $\ell_i \in \mathcal{A}$ **do**
    if $\rho \leq t$ **then**
      $\hat{s}_{i,*} \leftarrow$ `MisallocateSense` (ℓ_i);
    else
      $\hat{s}_{i,*} \leftarrow \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$;
    if $\hat{s}_{i,*}$ *exists* **then**
      remove ℓ_i from $\mathcal{A}$;
      put $\hat{s}_{i,*}$ in $\mathcal{D}$;

This algorithm is a modification of Algorithm 4 that describes the iterative approach, with the addition of the function `MisallocateSense`(ℓ_i). This function when given a lemma ℓ_i, will return a randomly selected sense that is known to be incorrect[6]. Implemented by a simple **if/else** statement, the decision is made whether to misallocate or to genuinely attempt a disambiguation based on what value t is set to.

Over the following two pages Figures 29 and 30 illustrate the observed effect of misallocating senses compared to both the *conventional* and *regular iterative* approach. For a fair comparison between the approaches, with each increase in t, F-Scores are calculated *only* for disambiguated lemmas that have a polysemy of $\rho > t$. Furthermore since a randomly selected incorrect sense is returned every time the function `MisallocateSense`(ℓ_i) is called, then each run of Algorithm 5 will almost certainly produce different results. Therefore the mean F-Score of 25 consecutive runs is taken as the final F-Score for the *misallocated iterative* approach in the Figure 29.

6 The sense's absence in the answer key for SemEval Task 12 is used as an indication that it would be an incorrect allocation to the input lemma.

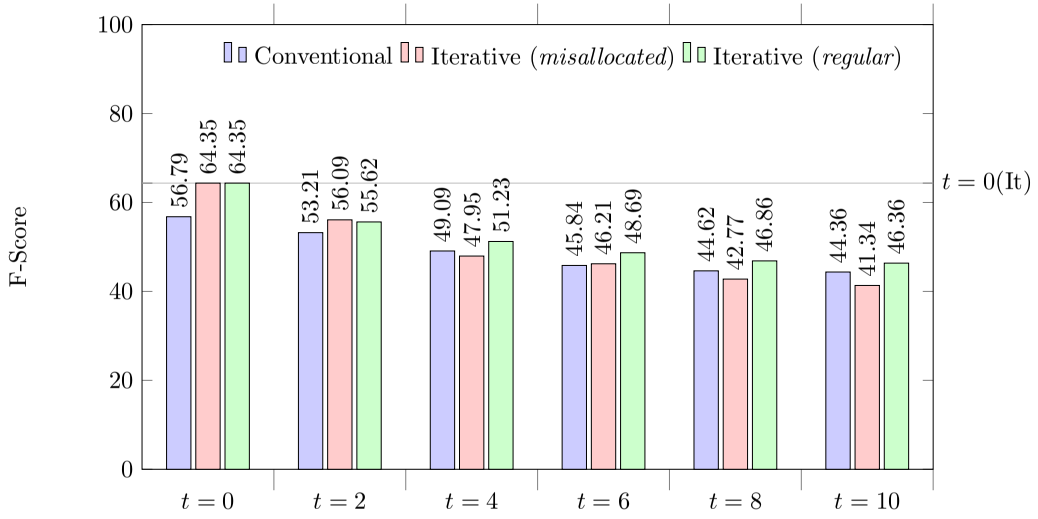


Figure 29: The respective F-Scores for all lemmas with polysemy $\rho > t$ are shown for the set of 13 documents. WSD is conducted at the document level with ϕ = Betweenness Centrality and $\mathcal{G}_{\mathcal{L}}$ = Shortest Paths. At increasing values of t, results are shown (from left to right) for the conventional approach (blue), the misallocated iterative approach (red) described by Algorithm 5 that has deliberate sense misallocations for all lemmas with polysemy $\rho \leq t$, and the regular iterative approach (green) described by Algorithm 4.

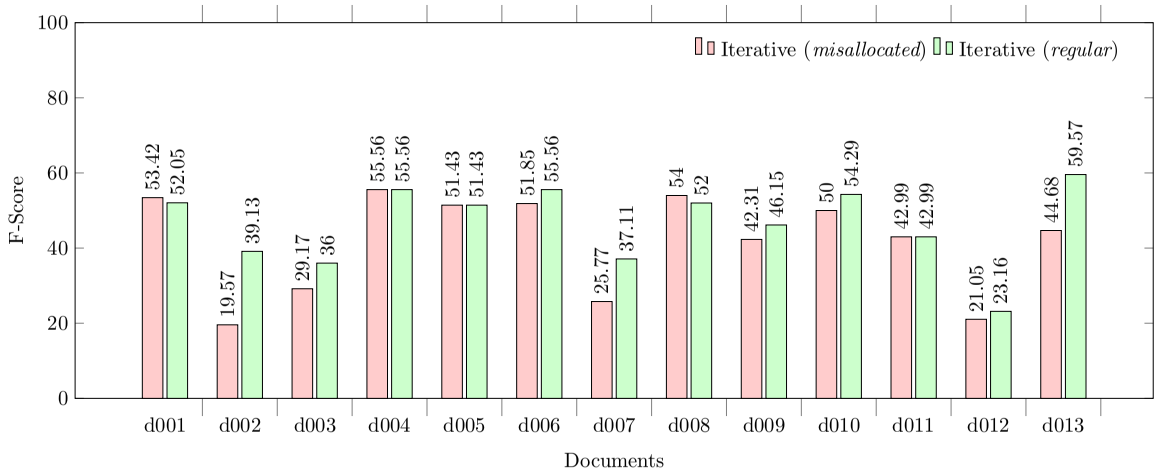


Figure 30: For a *single run* with t = 10, each of the 13 documents has the F-Score for the misallocated (on left in red) and regular (on right in green) iterative approach calculated only for all lemmas with a polysemy of $\rho > t$. Once again WSD is conducted at the document level with ϕ = Betweenness Centrality and $\mathcal{G}_{\mathcal{L}}$ = Shortest Paths.

In Figure 29 it appears that misallocations, similar to the Sudoku problem, do affect the final disambiguation solutions. As t increases there is a growing gap in F-Scores between the misallocated and regular iterative approach, yet it is not that large. While one wrong misallocation of a number to a cell in a Sudoku puzzle can be catastrophic, it is interesting to note that for iterative WSD this is not always the case. A closer inspection of Figure 30 shows that even though for documents d002, d007, and d013 the misallocated senses were significantly detrimental to the success of the iterative approach, a majority of documents remained unaffected such as d004, d005 and d011. To the author's surprise this demonstrates how robust the iterative approach is even when provided with a large amount of sense misallocations.

7.3.4 *Experiment 3: Adding a Little Optimisation*

Results for this chapter were processed right up to the point of submitting this thesis. Therefore briefly, an effort was made into optimising the iterative approach with subtree subgraphs. These results were compared with systems from SEMEVAL 2013 Task 12 (Navigli et al., 2013) in Table 18 below.

Team	System	P	R	F
UMCC-DLSI	Run-2$^+$	68.50	68.50	68.50
UMCC-DLSI	Run-3$^+$	68.00	68.00	68.00
UMCC-DLSI	Run-1$^+$	67.70	67.70	67.70
SUDOKU	It-PPR[M]$^+$	67.62	67.51	67.56
MACHINE	**MFS**	66.50	66.50	66.50
SUDOKU	It-PPR[M]	67.20	65.49	66.33
SUDOKU	It-PR[U]	64.07	62.44	63.24
SUDOKU	It-PD	63.58	61.41	62.47
DAEBAK!	PD$^+$	60.47	60.37	60.42
GETALP	BN-1$^+$	58.30	58.30	58.30
SUDOKU	PR[U]	60.09	54.06	56.91
GETALP	BN-2$^+$	56.80	56.80	56.80

Table 18: SemEval 2013 Task 12 Participant vs Iterative Results

Firstly, the author's original results as team DAEBAK! (Manion and Sainudiin, 2013) were able to be marginally improved, by applying the iterative approach to the Peripheral Diversity (see Chapter 6) graph-based centrality measure (It-PD). Next Personalised PageRank (It-PPR[M]) was tried with a surfing vector biased towards only *M*onosemous senses. Also included was regular PageRank (It-PR[U]) with a *U*niform surfing vector as a reference point. It-PPR[M] almost defeated the MFS baseline of 66.50, but lacked recall. To rectify this, the MFS baseline was used as a back-off strategy (It-PPR[M]$^{+}$)[7], which then led to the MFS baseline being beaten. As for the other teams, GETALP (Schwab et al., 2013) made use of an Ant Colony algorithm, while UMCC-DLSI (Gutiérrez et al., 2013) also made use of PPR, except they based the surfing vector on SemCor (Miller et al., 1993) sense frequencies, set L = 5 for shortest paths subgraphs, and disambiguated using resources external to BabelNet. Since their implementation of PPR beats this author's, it would be interesting to see how effective the iterative approach could be on the results of those teams.

7 Note that plus^{+} implies the use of a back-off strategy.

Part III

FRUITIONS & FUTURE WORK

Finally in Part III, the results of this research and future work are discussed in Chapter 8. This consolidates the achievements of each branch of this research and how the objectives stated in Chapter 3 have been met. Based on this, the future directions of research the author intends to pursue are then detailed.

8

CONCLUSIONS & FUTURE WORK

8.1 CONCLUSIONS

8.1.1 *Visualising Context in Semantic Subgraphs*

Mining Wikipedia and producing a large semantically weighted graph is not by itself a contribution to the field of Natural Language Processing (NLP). In fact slightly alternative methods in using edge-based TF-IDF vectors in conjunction with Cosine Similarity have already proven to be quite successful in creating semantic graphs. For example the thesauri developed by Nakayama et al. (2007) and evaluations completed by Milne and Witten (2008). The only real point of difference between each method in these works is the treatment of link direction.

In-bound links are computationally expensive to process and are not as semantically significant as out-bound links. Imagine the number and nature of article pages in Wikipedia that point to the article pages for *Europe* or *Barrack Obama*. The aforementioned works either reduce the influence of in-bound links on edge weights, or they simply do not process them at all. For the results produced in Chapter 4, all in-bound links were processed in the same way as out-bound links, purely out of curiosity. The author found that while they are expensive to process they do not appear to have any negative affect the semantic edge weights of $\mathcal{G}_c$.

Aside from link direction, there are also a range of link types in Wikipedia as illustrated by Figure 10 on page 44. The author decided not to give preference to, or make any distinction, between link types when calculating edge weights. This resulted in images more frequently appearing

as semantically *similar* rather than semantically *related* in the image clouds. In other words, images were almost always taken from another *article* page, rather than from alternative page types such as *category* and *portal* pages. This most likely happened because pages such as these have a higher degree out-bound and in-bound links, therefore receive lower edge weights. Gathering images from pages that share a more diverse set of semantic relationships with the target page could shift the visualisation of context in the image clouds to being more semantically related rather than similar. In future a restructured image cloud with tiers reserved for images from particular page types, could be a means of achieving this.

Overall, by mining Wikipedia to build a semantic graph, the author achieved the first objective of the thesis – to create a large graph of concepts and semantic relationships in order to perform subgraph-based WSD. While BabelNet was adopted in later chapters, through the experiments of Chapter 4 the existence and richness of context in semantic subgraphs was able to be visualised, giving further support to the subgraph based approach to WSD.

8.1.2 *Disambiguating Heterogeneity*

Admittedly, the internship at Pingar led the author a little astray from the objectives set out for this research. However experience in attempting disambiguation when merging heterogeneous semantic graphs as described in Chapter 5, later become beneficial when dealing with the heterogeneity of BabelNet (for example, the author's customised back-off strategy in Section 6.4.3 that takes heterogeneity into account when choosing a sense).

With continued advancements towards the semantic web, the pooling together heterogeneous structured knowledge will increase. To achieve this as an automated process, in which the concepts from one semantic graph are mapped to the concepts of another, this advocates another significant use for WSD.

Two key obstacles that had to be overcome for taxonomy generation, as well as for the creation of BabelNet (Navigli and Ponzetto, 2012a) were:

- Establishing a method to generate a universal context for concepts in a collection of heterogeneous semantic graphs
- Handling richer and poorer levels of concept context that can be extracted from heterogeneous semantic graphs

The author believes addressing these two obstacles and providing an automated means of merging together heterogeneous semantic graphs would make for an interesting SEMEVAL task in the future.

8.1.3 *Peripheral Diversity for WSD*

In Chapter 6 the author has contributed a new graph-based centrality measure to WSD literature. Dubbed with the name Peripheral Diversity (PD), it has demonstrated to be a competitive graph based measure in its early stages. For its entry into the SEMEVAL Task 12: Multilingual WSD (Navigli et al., 2013) it even outperformed the Most Frequent Sense (MFS) baseline in both French and Italian. Additionally, by implementing and testing PD the author needed to construct a complete WSD system, therefore achieving the second objective of this research.

Further experimentation on PD was undertaken after after SEMEVAL in which various combinations of subgraph filters, strategies, and input parameters were tried. Improvement was minimal, however application of the iterative approach was able to take PD's SEMEVAL task score from 60.42 to 62.47. Finally PD is a framework, in which a range of Pairwise Semantic Dissimilarity (PSD) measures could be tried in place of the Cosine Distance, again edge weights could also be factored in too. Thus there is still many more aspects for the author to improve PD from in future.

8.1.4 *The Iterative Approach*

The third objective of this research – to understand optimal conditions for unsupervised subgraph based WSD, is achieved by the extensive evaluations conducted in Chapter 7. The iterative approach demonstrated in a number of experiments that it can significantly improve the results of conventional subgraph-based WSD, even to the point of defeating the MFS baseline without doing anything complicated. This is regardless of the subgraph, graph centrality measure, or level of disambiguation. The performance of the iterative approach has also shown to not be excessive in processing time. Simultaneously, while the iterative approach benefits from higher levels of document monosemy, it also demonstrated how well it can recover when it is deliberately given a set of incorrect disambiguations in earlier iterations.

The iterative approach can still be extended much further, and the author encourages other researchers to rethink their own approaches to unsupervised knowledge-based WSD, particularly in regards to the interaction of subgraphs and graph centrality measures. In fact Agirre and Edmonds (2007) suggested a while ago, that exploiting interdependencies in context was a promising direction for WSD, which the experimental results of Section 7.3 have validated. For a long time the conventional approach to subgraph based WSD has been treated as a classification task. However the author has shown that by simply reducing the polysemy of subgraphs and re-constructing them based on previous disambiguations, that subgraph based WSD is better achieved as optimisation of the classification process. Given all of the above, this definitely warrants further investigation into the full potential of the iterative approach.

8.2 FUTURE WORK

Experiments on the iterative approach were also conducted on two other SENSEVAL/SEMEVAL tasks. These were:

- SENSEVAL 2004: The English All-Words Task (Snyder and Palmer, 2004)
- SEMEVAL-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain (Agirre et al., 2010)

The results were not published in this thesis, since they are very recent and not yet fully understood. However to summarise the results of both these evaluations, for each document the results tended to be better, yet compared to the improvements observed in Chapter 7 were less impressive. At first this was a little confounding, until the author recalled that both of these tasks only made use of WordNet (Fellbaum, 1998) where as the SEMEVAL 2013 multilingual WSD task a) only disambiguated nouns, and b) made use of BabelNet (Navigli and Ponzetto, 2012a) which is a marriage of both dictionary and encyclopedia.

Firstly, the encyclopedic senses included in BabelNet from Wikipedia are mostly named entities and therefore more likely to be monosemous. If Zipf's law is taken into account, the more monosemous nature of Wikipedia's named entities can be confirmed by the very high MFS baseline observed for the Wikipedia-focused WSD evaluation in (Navigli et al., 2013). Therefore while experiments need to be conducted to prove this, the author suspects that one of the key reasons the iterative approach is the most successful on the multilingual WSD task is because of BabelNet. WordNet as a lexicon contains senses that are more polysemous, therefore if this is the only sense inventory employed in a WSD evaluation task, it is less likely that an iterative approach which thrives on monosemy will perform as well as it could. BabelNet's inclusion of monosemous named entities, increases document monosemy which results in better performance of the iterative approach as seen in Figure 28 on page 103. In fact if other tasks had their

datasets re-tagged with BabelNet as the sense inventory, it would be interesting to see if the iterative approach could also perform better.

Finally, at the same time the iterative approach was published[1] in the conference proceedings of *SEM, so was the paper (Basile et al., 2014) in the co-located conference proceedings of COLING. With the use of distributional semantics – of which relies on characterising word senses as a distribution of words it keeps company with, the authors of this paper were not only able to defeat the MFS baseline, but also the beat the best result achieved in English by Gutiérrez et al. (2013) for the multilingual WSD task. While the use of distributional semantics is in fact a corpus-based supervised approach that resides outside the scope of this thesis, the author would also like to investigate whether its performance can benefit from an implementation of the iterative approach.

To conclude, the author hopes to have completed a more thorough analysis of the iterative approach for future publication and to extend this branch of research. Questions to begin with are:

- What constraints other than ρ could be exploited by the iterative approach?
- Could the iterative approach improve the performance of any particular NLP system under an in-vivo evaluation framework?
- What influence does the subgraph construction method have on the outcome of the iterative approach?
- Could other non-subgraph based methods of achieving WSD also benefit from the iterative approach?

This is now the end of this thesis. The author sincerely hopes it has provided the reader with an interesting and unique insight into the field of subgraph based WSD and the promise it holds for the future of NLP.

1 The author's publication on the iterative approach (Manion and Sainudiin, 2014) can be found in Appendix B.3

Part IV

APPENDICES

Part IV consists of both Appendix A and B. The former provides the original WSD system description that was submitted to the task organisers of SemEval Task 12. The latter contains the three peer reviewed publications that stem from Chapters 5, 6, and 7.

A

APPENDIX: PROJECT RESOURCES

A.1 SEMEVAL 2013 TASK 12 SYSTEM DESCRIPTION ON SUBMISSION

- Primary Researcher: Steve Manion, PhD candidate, University of Canterbury, New Zealand (slmanion@gmail.com)
- Supervisor: Dr Raazesh Sainudiin, Senior Lecturer, University of Canterbury, New Zealand (r.sainudiin@math.canterbury.ac.nz)

(1) Sense Inventory Used:

- BabelNet Core Lucene 1.1.1 (in conjunction with BabelNet Indexed Paths 1.0.1)

(2) WSD System Description: task12-DAEBAK!-PD (Peripheral Diversity)

This system constructs subgraphs from BabelNet Path Indexes v1.0.1 via use of the BabelNet API and BabelNet (both v1.1.1). The measure of “Peripheral Diversity” is built on several assumptions and observations about word senses by the authors and the

literature.

a) A sliding window of 5 sentences was used for the construction of the subgraphs in which there is no overlap. It has been demonstrated in research that local context (that of +/-2 words from the target word provide a bulk of its context). However since we are addressing nouns only and applying it to a semantic graph, from observations we consider 5 sentences as an appropriate window of local context (based on the English/French trial data). This approximately the size of a paragraph in any particular language, but is likely to vary of course.

b) We compensate for deviations in spelling for better mapping of synsets, the absence of a very important concept can drastically change the results of a subgraph structure.

c) The most frequent sense is a hard contender to beat in WSD, therefore we reward synsets that are more frequently used.

d) The more diversely a synset can be used also indicates its dominance as the most frequent sense, therefore we reward the diverse use of a synset (with respect to other synsets).

e) BabelNet is the combination of two different semantic resources, whatever nature of the WSD

algorithm, we must be careful not to be bias towards either of these resources.

f) Most importantly of all the above points, a relatively connected subgraph should reward the connectivity of synsets connected from different lemmas, as this is effectively the embodiment in graph form of the context we seek out from the subgraph.

To briefly describe our “Peripheral Diversity” algorithm, it does a breadth first search to up to 3 synsets, from there it gathers the peripheral nodes. Then we run metrics on how much these peripheral nodes diversify from each other and how often they are used. We assign the result of the metric as a score to the central synset.

A synset that is well connected and diversifies will often be picked, however there is also room for less frequent synsets to also be selected if they connect with other synsets from different lemmas. Likewise, more common synsets can be ignored if they fail to connect to other synsets, and as a result their peripheral nodes are all nodes internal to it and not diverse. A more detailed description with references to the above assumptions will be given in the paper. We have several more “Peripheral Diversity” algorithms we would like to explore after SemEval that are based on this investigation.

(3) Resources Used:

- BabelNet Core Lucene 1.1.1
- BabelNet Indexed Paths 1.0.1
- BabelNet API 1.1.1 and all its referenced libraries
- Self-constructed automated HTTP call to Wikipedia's "Did you mean" functionality to remove noise caused by misspellings and inconsistent formatting, mainly in the French and Italian test files.

(4) Languages Annotated:

- English
- French
- German
- Italian
- Spanish

B

APPENDIX: PUBLICATIONS

Peer reviewed publications produced from this thesis can be found over the following pages. By order of chapter occurrence, they are:

...from Chapter 5

Alyona Medelyan, Steve L. Manion, Jeen Broekstra, Anna Divoli, Anna-lan Huang, and Ian H. Witten (2013). Constructing a Focused Taxonomy from a Document Collection. *In Proceedings of the 10th Extended Semantic Web Conference (ESWC'13)*, pages 367-381, Montpellier, France. Springer, Heidelberg.

...from Chapter 6

Steve L. Manion and Raazesh Sainudiin (2013). DAEBAK!: Peripheral Diversity for Multilingual Word Sense Disambiguation. *In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13).*, pages 250-254, Atlanta, Georgia. ACL.

...from Chapter 7

Steve L. Manion and Raazesh Sainudiin. An Iterative 'Sudoku Style' Approach to Subgraph-based Word Sense Disambiguation (2014). *In Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (*SEM'14)*, pages 40–50, Dublin, Ireland. ACL.

Constructing a Focused Taxonomy from a Document Collection

Olena Medelyan,[1] Steve Manion,[1] Jeen Broekstra,[1] Anna Divoli,[1]
Anna-Lan Huang,[2] Ian H. Witten[2]

[1] Pingar Research, Auckland, New Zealand
(alyona.medelyan|steve.manion|anna.divoli)@pingar.com,
jeen@rivuli-development.com
[2] University of Waikato, Hamilton, New Zealand
(ahuang|ihw)@cs.waikato.ac.nz

Abstract. We describe a new method for constructing custom taxonomies from document collections. It involves identifying relevant concepts and entities in text; linking them to knowledge sources like Wikipedia, DBpedia, Freebase, and any supplied taxonomies from related domains; disambiguating conflicting concept mappings; and selecting semantic relations that best group them hierarchically. An RDF model supports interoperability of these steps, and also provides a flexible way of including existing NLP tools and further knowledge sources. From 2000 news articles we construct a custom taxonomy with 10,000 concepts and 12,700 relations, similar in structure to manually created counterparts. Evaluation by 15 human judges shows the precision to be 89% and 90% for concepts and relations respectively; recall was 75% with respect to a manually generated taxonomy for the same domain.

1 Introduction

Domain-specific taxonomies constitute a valuable resource for knowledge-based enterprises: they support searching, browsing, organizing information, and numerous other activities. However, few commercial enterprises possess taxonomies specialized to their line of business. Creating taxonomies manually is laborious, expensive, and unsustainable in dynamic environments (e.g. news). E↵ective automatic methods would be highly valued.

Automated taxonomy induction has been well researched. Some approaches derive taxonomies from the text itself [1], some from Wikipedia [2], while others combine text, Wikipedia and possibly WordNet to either extend these sources with new terms and relations [3] or carve a taxonomy tailored to a particular collection [4,5]. Our research falls into the last category, but extends it by defining a framework through which any combination of knowledge sources can drive the creation of document-focused taxonomies.

We regard taxonomy construction as a process with five clearly defined stages. The first, initialization, converts documents to text. The second extracts concepts and named entities from text using existing NLP tools. The third connects

named entities to Linked Data sources like Freebase and DBpedia. The fourth identifies conflicting concept mappings and resolves them with an algorithm that disambiguates concepts that have matching labels but different URIs. The fifth connects the concepts into a single taxonomy by carefully selecting semantic relations from the original knowledge sources, choosing only relations that create meaningful hierarchies given the concept distribution in the input documents. These five stages interoperate seamlessly thanks to an RDF model, and the output is a taxonomy expressed in SKOS, a standard RDF format.

The method itself is domain independent—indeed the resulting taxonomy may span multiple domains covered by the document collection and the input knowledge sources. We have generated and made available several such taxonomies from publicly available datasets in five different domains.[3] This paper includes an in-depth evaluation of a taxonomy generated from news articles. Fifteen human judges rated the precision of concepts at 89% and relations at 90%; recall was 75% with respect to a manually built taxonomy for the same domain. Many of the apparently missing concepts are present with different—and arguably more precise—labels.

Our contribution is threefold: (a) an RDF model that allows document-focused taxonomies to be constructed from any combination of knowledge sources; (b) a flexible disambiguation technique for resolving conflicting mappings and finding equivalent concepts from different sources; and (c) a set of heuristics for merging semantic relations from different sources into a single hierarchy. Our evaluation shows that current state-of-the-art concept and entity extraction tools, paired with heuristics for disambiguating and consolidating them, produce taxonomies that are demonstrably comparable to those created by experts.

2 Related Work

Automatic taxonomy induction from text has been studied extensively. Early corpus-based methods extract taxonomic terms and hierarchical relations that focus on the intrinsic characteristics of a given corpus; external knowledge is rarely consulted. For example, hierarchical relations can be extracted based on term distribution statistics [6] or using lexico-syntactic patterns [7,1]. These methods are usually unsupervised, with no prior knowledge about the corpus. However, they typically assume only a single sense per word in the corpus, and produce taxonomies based on words rather than word senses.

Research has been conducted on leveraging knowledge bases to facilitate taxonomy induction from both closed- and open-domain text collections. Some researchers derive structured taxonomies from semi-structured knowledge bases [2,8] or from unstructured content on the Web at large [9]. Others expand knowledge bases with previously unknown terms and relations discovered from large corpora—for example, Matuszek et al. enrich the Cyc knowledge base with information extracted from the Web [10], while Snow et al. expand WordNet with new synsets by using statistical classifiers built from lexical information extracted

[3] http://bit.ly/f-step

from news articles [3]. Still others interlink documents and knowledge bases: they match phrases in the former with concepts in the latter [11,12] and identify taxonomic relations between them [4,5]. These studies do address the issue of sense ambiguity: polysemous phrases are resolved to their intended senses while synonyms are mapped to the same concept. However, they typically only consult a single source and users do not intervene in the taxonomy construction process.

The Castanet project [4] and Dakka and Ipeirotis's research [5] relate closely to our work. They both derive hierarchical metadata structures from text collections and both consult external sources—WordNet in the former case and Wikipedia, WordNet and the Web in the latter—to find important concepts in documents. Castanet identifies taxonomic relations based on WordNet's `is-a` relations, whereas Dakka and Ipeirotis use subsumption rules [6]. The latter only select those taxonomic concepts for final groupings that occur frequently in the documents in non-related contexts. In contrast to our work, both studies represent the extracted information as hierarchical faceted metadata: the outcome is no longer a single taxonomy but is instead split into separate facets. Although Dakka and Ipeirotis consult multiple sources, they do not check which concepts are the same and which are different. In contrast, we explicitly address the problem of sense disambiguation and consolidation with multiple sources.

Our work also intersects with research on relation extraction and ontology induction from text, the closest being [13], which also links phrases in text to Wikipedia, DBpedia and WordNet URIs, extracts relations, and represents them as RDF. However, their input is a single short piece of text, whereas we analyze an entire document collection as a whole, and focus on organizing the information hierarchically.

3 Architecture of the Taxonomy Generator

The primary input to our taxonomy generator is a collection of documents and, optionally, a taxonomy for a related domain (e.g., the Agrovoc thesaurus or the Gene ontology). Our system automatically consults external knowledge sources, and links concepts extracted from the documents to terminology in these sources. By default we use Freebase, DBpedia and Wikipedia, but domain-specific linked data sources like Geonames, BBC Music, or the Genbank Entrez Nucleotide database can also be consulted.[4] Finally, a small taxonomy with preferred root nodes can be supplied to guide the upper levels of the generated taxonomy.

3.1 Defining Taxonomies in SKOS

The result of each step of the taxonomy generation process is stored as an RDF data structure, using the Simple Knowledge Organization System vocabulary. SKOS is designed for sharing and linking thesauri, taxonomies, classification schemes and subject heading systems via the Web.[5] An SKOS model consists

[4] Suitable linked data sources can be found at `http://thedatahub.org/group/lodcloud`
[5] See `http://www.w3.org/2004/02/skos`

of a hierarchical collection of *concepts*, defined as "units of thought"—abstract entities representing ideas, objects or events. A concept is modeled as an instance of the class `skos:Concept`. An `skos:prefLabel` attribute records its preferred name and `skos:altLabel` attributes record optional synonyms. Concepts are linked via semantic relations such as `skos:broader` (to indicate that one concept is broader in meaning than another) and its inverse `skos:narrower`. These relations allow concepts to be structured into a taxonomic hierarchy.

Our goal is to produce a new knowledge organization system (a taxonomy) based on heterogeneous sources, including concepts extracted from text as well as concepts in existing sources, and SKOS is a natural modeling format. Also, many existing public knowledge systems are available online as SKOS data,[6] and reusing these sources ensures that any taxonomy we generate is immediately linked via concept mappings to third-party data sources on the Web.

3.2 Information Model

We have built a set of loosely coupled components that perform the individual processing steps. Each component's results are stored as RDF data in a central repository using the OpenRDF Sesame framework [14].

Figure 1 shows the information model. The central class is `pw:Ngram`, which represents the notion of an extracted string of N words. The model records every position of the ngram in the input text, and each occurrence of the same ngram in the same document is a single instance of the `pw:Ngram` class.

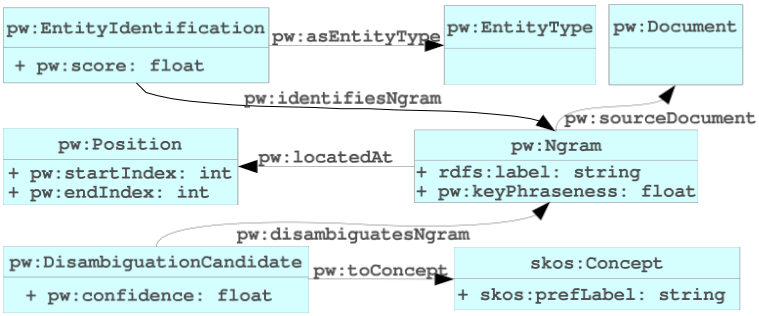


Fig. 1. Shared RDF model for ngram and entity information

The `pw:EntityType` class supports entity typing of ngrams. It has a fixed number of instances representing types such as people, organizations, locations, events, etc. In order to be able to record the relation between an ngram and its type, as well as an identification score reported by the extraction tool, the relation is modeled as an an object, of type `pw:EntityIdentification`.

[6] See a.o. `http://www.w3.org/2001/sw/wiki/SKOS/Datasets`

`pw:DisambiguationCandidate` is introduced to allow ngrams to be annotated with corresponding concepts from external sources. This class records the relation (and the system's confidence in it) between an extracted ngram and an external source. These external sources are modeled as instances of `skos:Concept`. They are the building blocks of the taxonomy we generate.

Using a shared RDF model to hold extracted data ensures that components can interoperate and reuse each other's results. This is a significant advantage: it facilitates the use of di↵erent language processing tools in a single system by mapping their outputs to a common vocabulary. Moreover, users can add other Linked Data sources, and insert and remove processing steps, as they see fit. It can also be used for text annotation.[7]

In addition, the use of an RDF repository allows one to formulate SPARQL[8] queries over the aggregated data. Using these, data from di↵erent components can be analyzed quickly and e ciently at each processing step.

4 Generating the Taxonomy

Figure 2 shows the processing steps in our system, called F-STEP (Focused SKOS Taxonomy Extraction Process). Existing tools are used to extract entities and concepts from document text (steps 2a and 2b respectively in the Figure). Purpose-built components annotate entities with information contained in Linked Data sources (step 3), disambiguate concepts that are mapped to the same ngram (step 4), and consolidate concepts into a hierarchy (step 5).

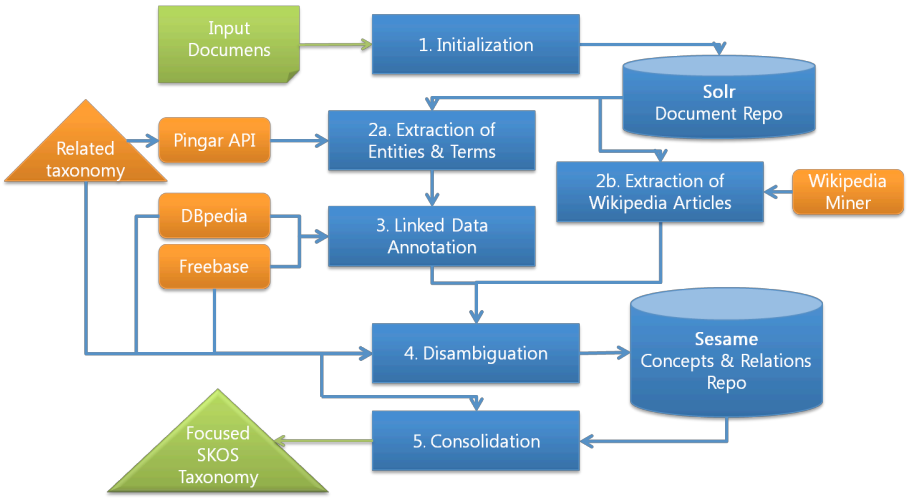


Fig. 2. Automated workflow for turning input documents into a focused taxonomy

[7] A possible alternative is the recently-defined NLP2RDF format `http://nlp2rdf.org`.
[8] See `http://www.w3.org/TR/sparql11-query/`

4.1 Initialization

Taxonomies organize knowledge that is scattered across documents. To federate inputs stored on file systems, servers, databases and document management systems, we use Apache Tika to extract text content from various file formats and Solr for scaleable indexing.[9] Solr stores multiple document collections in parallel, each document being referenced via a URL, which allows concepts to be linked back to the documents containing them in our RDF model.

4.2 Extracting Named Entities and Concepts

Extraction step 2a in Figure 2 uses a text analytics API[10] to identify names of people, organizations and locations, and to identify relevant terms in an existing taxonomy if one is supplied. Step 2b uses the Wikipedia Miner toolkit [15] to relate documents to relevant concepts in Wikipedia.

Named Entities. Names of people, organizations, and locations are concepts that can usefully be included in a taxonomy; existing systems extract such entities with an accuracy of 70%–80% [16]. We extract named entities from the input documents using the text analytics API and convert its response to RDF. Named entities are represented by a `pw:EntityIdentification` relation between the original ngram and an entity type. The entities are passed to the annotation step to disambiguate any matches to Linked Data concepts.

Concepts from Related Taxonomies. As mentioned in Section 3, the input can include one or more taxonomies from related domains. The same text analytics API records any concepts in a related taxonomy that appear in the input documents, maps them to SKOS, and links to the source document ngram via a `pw:DisambiguationCandidate` relation.

Concepts from Wikipedia. Each Wikipedia article is regarded as a "concept." Articles describe a single concept, and for (almost) any concept there exists a Wikipedia article. We use the Wikipedia Miner toolkit to annotate ngrams in the text with corresponding Wikipedia articles. This toolkit allows the number of annotations to be controlled, and disambiguates ngrams to their correct meaning—for example, the word *kiwi* may refer to a.o. a bird, a fruit, a person from NZ, or the NZ national rugby league team, all of which have distinct Wikipedia entries. The approach is described in detail in [15].

The user determines what kind of concepts will be included in the taxonomy. For example, if no related taxonomies are available, only named entities and Wikipedia articles returned by the Wikification process will be included in the final taxonomy.

[9] See `http://tika.apache.org/` and `http://lucene.apache.org/solr/`
[10] See `http://apidemo.pingar.com`

4.3 Annotating with Linked Data

Once entities such as people, places, and organisations have been extracted, the annotation step queries Freebase [17] and DBpedia [18] for corresponding concepts (Figure 2, step 3). The queries are based on the entity's type and label, which is the only structured information available at this stage. Other Linked Data sources can be consulted in this step, either by querying via a SPARQL endpoint,[11] which is how we consult DBpedia, or by accessing the Linked Data source directly over the HTTP protocol.

We define mappings of our three entity types to Linked Data concept classes. For example, in the case of Freebase, our entity type "Person" (`pw:person`) is mapped to `http://rdf.freebase.com/ns/people/person`, and for each extracted *person* entity Freebase is queried for lexically matching concepts of the mapped type. Several candidate concepts may be selected for each entity (the number is given as a configuration parameter). These matches are added as disambiguation candidates to every ngram that corresponds to the original entity.

4.4 Disambiguation

The preceding processing steps use various techniques to determine relevant concepts in documents. A direct consequence is that a given ngram may be mapped to more than one concept: a taxonomy term, a Wikipedia article, a Freebase or a DBpedia concept. Although the Wikipedia Miner incorporates its own built-in disambiguation component, this merely ensures that at most one Wikipedia concept corresponds to each ngram. A second disambiguation step (Figure 2, step 4) determines whether concepts from *di↵erent* sources share the same meaning and whether their meaning is contextually relevant.

The disambiguation is performed for each document, one ngram at a time. If an ngram has a single concept mapping, it is considered unambiguous and this concept is added to the final taxonomy. If an ngram has multiple mappings, the conflicting concepts are inspected first. Here, we compare the context of the ngram with the contexts of each concept, as it is defined in its original source. The context of the ngram is as a set of labels of concepts that co-occur in the same document, whereas the context of each concept is a set of labels derived from its associated concepts, computed in a way that depends on the concept's origin. In SKOS taxonomies, associated concepts are determined via `skos:broader`, `skos:narrower`, and `skos:related` relations. For each associated concept we collect the `skos:prefLabel` and one or more `skos:altLabels`. In Wikipedia, these labels are sourced from the article's redirects, its categories, the articles its abstract links to, and other linked articles whose semantic relatedness [15] exceeds a certain threshold (we used 0.3, which returns 27 linked articles on average). In the case of Freebase and DBpedia, we utilize the fact that many Freebase concepts have mappings to DBpedia, which in turn are (practically all) mapped to Wikipedia articles. We locate the corresponding Wikipedia article and use the above method to determine the concepts.

[11] A SPARQL endpoint is a web service that implements the W3C SPARQL protocol

Once all related labels have been collected we calculate the distance between every pair of labels. To account for lexical variation between the labels, we use the Dice coe cient between the sets of bigrams that represent the labels. We then compute a final similarity score by averaging the distance over the top n scoring pairs. n is chosen as the size of the smaller set, because if the concepts the sets represent are truly identical, every label in the smaller set should have at least one reasonably similar partner in the other set; larger values of n tend to dilute the similarity score when one of the concepts has many weakly associated concept labels, which is often the case for Wikipedia concepts.

Given this similarity metric, disambiguation proceeds as follows. First, we choose the concept with the greatest similarity to the ngram's context to be the canonical concept. (This assumes that there is at least one correct concept among the conflicting ones.) Second, we compare the similarity of every other candidate concept to the canonical one and, depending on its similarity score s, list it as an `skos:exactMatch` (if $s > 0.9$), an `skos:closeMatch` (if $0.9 \quad s \quad 0.7$), or discard it (if $s < 0.7$). The thresholds were determined empirically.

As an example of disambiguation, the ngram *oceans* matches three concepts: *Ocean*, *Oceanography* (both Wikipedia articles), and *Marine areas* (a taxonomy concept). The first is chosen as the canonical concept because its similarity with the target document is greatest. *Marine areas* is added as `skos:closeMatch`, because its similarity with *Ocean* is 0.87. However, *Oceanography*'s similarity falls below 0.7, so it is discarded. As a another example, the ngram *logged* is matched to both *Logs* (a taxonomy concept) and *Deforestation* (a Wikipedia article). *Logs* is semantically connected to another taxonomy concept, which is why it was not discarded by the text analytics API, but it is discarded by the disambiguation step because it is not su ciently closely related to other concepts that occur in the same document.

4.5 Consolidation

The final step is to unite all unambiguous and disambiguated concepts found in documents into a single taxonomy. Each concept lists several URIs under `skos:exactMatch` and (possibly) `skos:closeMatch` that define it in other sources: the input taxonomy, Wikipedia, Freebase and DBpedia. These sources already organize concepts into hierarchies, but they di↵er in structure. The challenge is to consolidate these hierarchies into a single taxonomy.

Sources of Relations. Taxonomies from related domains, as optional inputs, already define the relations we seek: `skos:broader` and `skos:narrower`. However, they may cover certain areas in more or less detail than what we need, which implies that some levels should be flattened while others are expanded. Because *broader* and *narrower* are transitive relations, flattening is straightforward. For expansion, concepts from other sources are needed.

Wikipedia places its articles into categories. For example, the article on George Washington belongs to 30 categories; some useful, e.g. *Presidents of the*

US and *US Army generals*, and others that are unlikely to be relevant in a taxonomy, e.g. *1732 births*. Some articles have corresponding categories (e.g., there is a category "George Washington"), which lead to further broader categories. Furthermore, names may indicate multiple relations (e.g. *Politicians of English descent* indicates that *George Washington* is both a *Politician* and *of English descent*). Wikipedia categories tend to be fine-grained, and we discard information to create broader concepts. We remove years (*1980s TV series* becomes *TV series*), country and language identifiers (*American sitcoms* becomes *Sitcoms*; *Italian-language comedy films* becomes *Comedy films*), and verb and prepositional phrases that modify a head noun (*Educational institutions established in the 1850s* becomes *Educational institutions*; *Musicians by country* becomes *Musicians*). The entire Wikipedia category structure is available on DBpedia in SKOS format, which makes it easy to navigate. We query the SPARQL DBpedia endpoint to determine categories for a given Wikipedia article.

Other potential sources are Freebase, where categories are defined by users, and DBpedia, which extracts relations from Wikipedia infoboxes. We plan to use this information in future when consolidating taxonomies.

Consolidation Rules. F-STEP consolidates the taxonomy that has been generated so far using a series of rules. First, direct relations are added between concepts. For each concept with a SKOS taxonomy URI, if its broader and narrower concepts match other input concepts, we connect these concepts, e.g. *Air transport* `skos:narrower` *Fear of flying*. If a concept has a Wikipedia URI and its immediate Wikipedia categories match an existing concept, we connect these concepts, e.g. *Green tea* `skos:narrower` *Pu-erh tea*.

Following the intuition that some concepts do not appear in the documents, but may useful for grouping others that do, we iteratively add such concepts. For each concept with a SKOS taxonomy URI, we use a transitive SPARQL query to check whether it can be connected by new intermediate concepts to other concepts. If a new concept is found, it is added to the taxonomy and its relations are populated for all further concepts. For example, this rule connects concepts like *Music* and *Punk rock* via a new concept *Music genres*, whereupon a further relation is added between *Music genres* and *Punk rock*.

Next, the Wikipedia categories are examined to identify those of interest. The document collection itself is used to quantify the degree of interest: categories whose various children co-occur in many documents tend to be more relevant. Specifically, a category's "quality" is computed by iterating over its children and checking how many documents contain them. If this score, normalized by the total number of comparisons made, exceeds a given threshold, the category is added to the output taxonomy. This helps eliminate categories that combine too many concepts (e.g. *Living people* in a news article) or that do not group co-occurring concepts, and singles out useful categories instead (e.g. *Seven Summits* might connect *Mont Blanc*, *Puncak Jaya*, *Aconcagua*, and *Mount Everest*). Next, we retrieve broader categories for these newly added categories and check whether their names match existing concepts, allowing us to add new

relations. One could continue up the Wikipedia category tree, but the resulting categories are less satisfactory. For example, *Music* belongs to *Sound*, which in turn belongs to *Hearing*, but the relation between *Music* and *Hearing* is associative rather than hierarchical. In fact, unlike conventional SKOS taxonomies, the Wikipedia category structure is not, in general, transitive.

Parentheses following some Wikipedia article names indicate possible groupings for a concept, e.g. *Madonna_(entertainer)* is placed under *Entertainers*, if such a concept exists. We also match each category name's last word against existing concept names, but choose only the most frequent concepts to reduce errors introduced by this crude technique.

We group all named entities that are found in Freebase using the Freebase categories, and all those found in DBpedia using the corresponding Wikipedia categories. The remainder are grouped by their type, e.g. *John Doe* under *Person*.

These techniques tend to produce forests of small subtrees, because general concepts rarely appear in documents. We check whether useful general terms can be found in a related taxonomy, and also examine the small upper-level taxonomy that a user may provide, as mentioned in Section 1. For example, a media website may divide news into *Business*, *Technology*, *Sport* and *Entertainment*, with more specific areas underneath, e.g. *Celebrities*, *Film*, *Music*—a two-level taxonomy of broad categories. For each input concept we retrieve its broadest concept—the one below the root—and add it, skipping intermediate levels. This rule adds relations like *Cooperation* `skos:broader` *Business and industry*.

Pruning Heuristics. Prunning can make a taxonomy more usable, and eliminate redundancies. First, following [4], who extract a taxonomy from WordNet, we elide parent–child links for single children. If a concept has a single child that itself has one or more children, we remove the child and point its children directly to its parent.

Second, we eliminate multiple inheritance that repeats information in the same taxonomy subtree, which originates from redundancy in the Wikipedia category structure. We identify cases where either relations or concepts can be removed without compromising the tree's informativeness. Figure 3 shows examples. In (a) the two-parent concept *Manchester United FC* is reduced to a single parent by removing a node that does not otherwise contribute to the structure. In (b) the two-parent concept *Tax* is reduced to a single parent by removing a small redundant subtree. In (c) a common parent of the two-parent concepts *The Notorious B.I.G.* and *Tupac Shakur* is pruned.

5 Evaluation and Discussion

Domain-specific taxonomies (and ontologies) are typically evaluated by (a) comparing them to manually-built taxonomies, (b) evaluating the accuracy of their concepts and relations, and (c) soliciting feedback from experts in the field. This section evaluates our system's ability to generate a taxonomy from a news collection. We give an overview of the dataset used, compare the dimensions of the

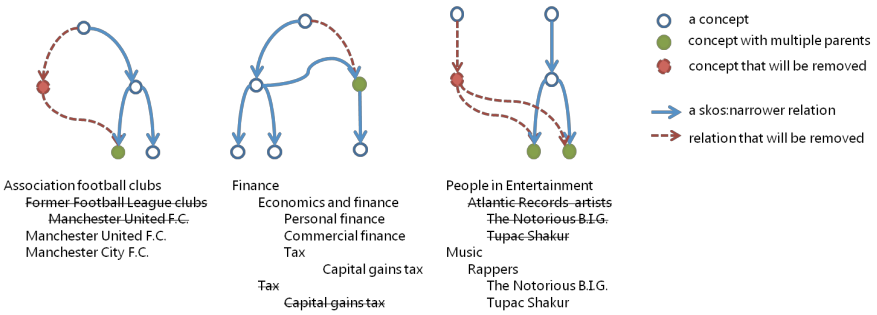


Fig. 3. Pruning concepts and relations to deal with multiple inheritance

taxonomy generated with other taxonomies, assess its coverage by comparing it with a hand-built taxonomy for the domain, and determine the accuracy of both its concepts and its relations with respect to human judgement.

5.1 The Domain

Fairfax Media is a large media organization that publishes hundreds of news articles daily. Currently, these are stored in a database, organized and retrieved according to manually assigned metadata. Manual assignment is time-consuming and error-prone, and automatically generated metadata, organized hierarchically for rapid access to news on a particular topic or in a general field, would be of great benefit.

We collected 2000 news articles (4.3MB of uncompressed text) from December 2011, averaging around 300 words each. We used the UK Integrated Public Service Sector vocabulary (`http://doc.esd.org.uk/IPSV/2.00.html`) as an input taxonomy. A taxonomy was extracted using the method described in Section 4 and can be viewed at `http://bit.ly/f-step`. It contains 10,150 concepts and 12,700 relations and is comparable in size to a manually-constructed taxonomy for news, the New York Times taxonomy (`data.nytimes.com`), which lists 10,400 *People*, *Organizations*, *Locations* and *Descriptors*. The average depth of the tree is 2.6, with some branches being 10 levels deep. Each concept appears in an average of 2 news articles. The most frequent, *New Zealand*, appears as metadata for 387 articles; the most topical, *Christmas*, is associated with 127 articles. About 400 concepts were added during the consolidation phase to group other concepts, and do not appear as metadata.

5.2 Coverage Comparison

To investigate the coverage of the automatically-generated taxonomy, we compared it with one comprising 458 concepts that Fairfax librarians had constructed manually to cover all existing and future news articles. Interestingly, this taxonomy was never completed, most likely because of the labor involved. Omissions

tend to be narrower concepts like individual sports, movie genres, music events, names of celebrities, and geographic locations. In order to evaluate our new taxonomy in terms of recall, we checked which of the 458 manually assigned concepts have labels that match labels in the new taxonomy (considering both preferred or alternative labels in both cases). There were a total of 271 such "true positives," yielding a recall of 59%. However, not all the manually assigned concepts are actually mentioned in the document set used to generate our taxonomy, and are therefore, by definition, irrelevant to it. We used Solr to seek concepts for which at least one preferred or alternative label appears in the document set, which reduced the original 458 concepts to 298 that are actually mentioned in the documents. Re-calculating the recall yields a figure of 75% (224 out of 298).

Inspection shows that some of the missing concepts are present but with different labels—instead of *Drunk*, the automatically generated taxonomy includes *Drinking alcohol* and *Alcohol use and abuse*. Others are present in a more specific form—instead of *Ethics* it lists *Ethical advertising* and *Development ethics*. Nevertheless, some important concepts are missing—for example, *Immigration*, *Laptop* and *Hospitality*.

5.3 Accuracy of Concepts

Fifteen human judges were used to evaluate the precision of the concepts present in the taxonomy generated from the documents. Each judge was presented with the text of a document and the taxonomy concepts associated with it, and asked to provide yes/no decisions on whether the document refers to each term. Five documents were chosen and given to all judges; a further 300 documents were distributed equally between the judges.

Looking first at the five common documents, the system extracted 5 to 30 concepts from each, with an average of 16. Three judges gave low scores, agreeing with only 74%, 86% and 90% of the concepts respectively, averaged over the five documents. The remaining 12 each agreed with virtually all—more than 97%—of the concepts identified by the system. The overall precision for automatic identification of concepts, averaged over all 15 judges, was 95.2%.

Before these figures were calculated the data was massaged slightly to remove an anomaly. It turned out that the system identified for each article the name of the newspaper in which it was published (e.g. *Taranaki Daily News*), but the human judges disagreed with one another on whether that should be counted as a valid concept for the article. A decision was taken to exclude the name of the newspaper from the first line of the article.

Turning now to the 300 documents that were examined by one judge each, the system identified a total of 3,347 concepts. Of these, 383 were judged incorrect, yielding an overall precision of 88.6%. (In 15 cases the judge was unwilling to give a yes/no answer; these were counted as incorrect.) Table 1 shows the source of the errors. Note that any given concept may originate in more than one source, which explains the discrepancy in the total of the Errors column (393, not 383). The most accurate concepts are ones that describe people. The most error-prone ones emanate from the input taxonomy, 26% of which are incorrect. This taxonomy

Table 1. Sources of error in concept identification

Type	Number	Errors	Rate
People	1145	37	3.2%
Organizations	496	51	10.3%
Locations	988	114	11.5%
Wikipedia named entities	832	71	8.5%
Wikipedia other entities	99	16	16.4%
Taxonomy	868	229	26.4%
DBPedia	868	81	8.1%
Freebase	135	12	8.9%
Overall	3447	393	11.4%

describes rather general concepts, which introduces more ambiguity than the other sources.

5.4 Accuracy of Relations

The same fifteen judges were used to evaluate the precision of the hierarchical relations present in the taxonomy. Each judge received 100 concept pairs and was asked for a yes/no decision as to whether that relation makes sense—i.e., whether the first concept really is narrower than the second. A total of 750 relations were examined, each adjudicated by two different judges.

The overall precision figure was 90%—that is, of the 1500 decisions, judges expressed disagreement in 150 cases. The interannotator agreement, calculated as the number of relationships that both judges agreed on expressed as a proportion of all relationships, was 87%.

An examination of where the two judges made different decisions revealed that some were too strict, or simply wrong (for example, *Acid @ base chemistry*, *Leeds @ North Yorkshire*, *History of Israel @ Israel*, where @ means "has parent"). Indeed, it appears that, according to some judges, polio is not an infectious disease and Sweden is not in Scandinavia! It is interesting to analyze the clear errors, discarding cases where the judges conflicted. Of the 25 situations where both judges agreed that the system was incorrect, ten pairs were related but not in a strict hierarchical sense (e.g., *Babies 6@ school children*), four were due to an overly simplistic technique that we use to identify the head of a phrase (e.g. *Daily Mail 6@ Mail*), two could have (and should have) been avoided (e.g. *League 6@ League*), and nine were clearly incorrect and correspond to bugs that deserve further investigation (e.g. *Carter Observatory 6@ City*).

6 Conclusions

This paper has presented a new approach to analyzing documents and generating taxonomies focused on their content. It combines existing tools with new

techniques for disambiguating concepts originating from various sources and consolidating them into a single hierarchy. A highlight of the scheme is that it can be easily extended. The use of RDF technology and modeling makes coupling and reconfiguring the individual components easy and flexible. The result, an SKOS taxonomy that is linked to both the documents and Linked Data sources, is a powerful knowledge organization structure that can serve many tasks: browsing documents, fueling facetted search refinements, question answering, finding similar documents, or simply analyzing one's document collection.

The evaluation has shown that in one particular scenario in the domain of news, the taxonomy that is generated is comparable to manually built exemplars in the dimensions of the hierarchical structure and in its coverage of the relevant concepts. Recall of 75% was achieved with respect to a manually generated taxonomy for the same domain, and inspection showed that some of the apparently missing concepts are present but with di↵erent—and arguably more precise—labels. With respect to multiple human judgements on five documents, the accuracy of concepts exceeded 95%; the figure decreased to 89% on a larger dataset of 300 documents. The accuracy of relations was measured at 90% with respect to human judgement, but this is diluted by human error. Analysis of cases where two judges agreed that the system was incorrect revealed that at least half were anomalies that could easily be rectified in a future version. Finally, although we still plan to perform an evaluation in an application context, initial feedback from professionals in the news domain is promising. Some professionals expect to tweak the taxonomy manually by renaming some top concepts, removing some irrelevant relations, or even re-grouping parts of the hierarchy, and we have designed a user interface that supports this.

Compared to the e↵ort required to come up with a taxonomy manually, a cardinal advantage of the automated system is speed. Given 10,000 news articles, corresponding to one week's output of Fairfax Media, a fully-fledged taxonomy is generated in hours. Another advantage is that the taxonomy focuses on what actually appears in the documents. Only relevant concepts and relations are included, and relations are created based on salience in the documents (e.g. occurrence counts) rather than background knowledge. Finally, because Wikipedia and Freebase are updated daily by human editors, the taxonomy that is produced is current, which is important for ever-changing domains such as news.

Finally, the approach is applicable to any domain. Every knowledge-based organization deals with mountains of documents. Taxonomies are considered a very useful document management tool, but uptake has been been slow due to the e↵ort involved in building and maintaining them. The scheme described in this paper reduces that cost significantly.

Acknowledgements. This work was co-funded by New Zealand's Ministry of Science and Innovation. We also thank David Milne and Shane Stuart from the University of Waikato and Reuben Schwarz from Fairfax Media NZ.

References

1. Caraballo, S.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proc. of the 37th Annual Meeting of the ACL, ACL (1999) 120–126
2. Ponzetto, S., Strube, M.: Deriving a large scale taxonomy from wikipedia. In: Proc. of the 22nd National Conference on Artificial Intelligence, AAAI Press (2007) 1440–1445
3. Snow, R., Jurafsky, D., Ng, A.: Semantic taxonomy induction from heterogenous evidence. In: Proc. of the 21st Intl. Conf. on Computational Linguistics, ACL (2006) 801–808
4. Stoica, E., Hearst, M.A.: Automating creation of hierarchical faceted metadata structures. In: In Procs. of the HLT/NAACL Conference. (2007)
5. Dakka, W., Ipeirotis, P.: Automatic extraction of useful facet hierarchies from text databases. In: Proc. of the 24th IEEE Intl. Conf. on Data Engineering, IEEE (2008) 466–475
6. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: Proc. of the 22nd Annual Intl. Conf. on R&D in Information Retrieval, ACM (1999) 206–213
7. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proc. of the 14th Conference on Computational linguistics, ACL (1992) 539–545
8. Suchanek, F., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proc. of the 16th Intl. Conference on World Wide Web, ACM (2007) 697–706
9. Wu, W., Li, H., Wang, H., Zhu, K.: Probase: A probabilistic taxonomy for text understanding. In: Proc. of the 2012 ACM Intl. Conf. on Management of Data, ACM (2012) 481–492
10. Matuszek, C., Witbrock, M., Kahlert, R., Cabral, J., Schneider, D., Shah, P., Lenat, D.: Searching for common sense: Populating cyc from the web. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence, AAAI Press (2005) 1430–1435
11. Milne, D., Witten, I.: Learning to link with wikipedia. In: Proc. of the 17th Conference on Information and Knowledge Management, ACM (2008) 509–518
12. Mendes, P., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proc. of the 7th Intl. Conf. on Semantic Systems, ACM (2011) 1–8
13. Augenstein, I., Pado, S., Rudolph, S.: LODifier: Generating Linked Data from Unstructured Text. In: Proc. of the 9th Extended Semantic Web Conference (ESWC 2012). Number 7295 in LNCS, Springer Verlag, Heidelberg (2012) 210–224
14. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In: Proc. of the 1st Intl. Semantic Web Conference. Number 2342 in LNCS, Springer Verlag, Heidelberg Germany (2002) 54–68
15. Milne, D., Witten, I.H.: An open-source toolkit for mining Wikipedia. Artificial Intelligence (2012)
16. Marrero, M., Sanchez-Cuadrado, S., Lara, J., Andreadakis, G.: Evaluation of Named Entity Extraction Systems. In: Proc. of the Conference on Intelligent Text Processing and Computational Linguistics, CICLing. (2009) 47–58
17. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proc. of Intl. Conf. on Management of Data. SIGMOD '08, New York, NY, USA, ACM (2008) 1247–1250
18. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: In 6th Intl. Semantic Web Conference, Busan, Korea, Springer (2007) 11–15

DAEBAK!: Peripheral Diversity for Multilingual Word Sense Disambiguation

Steve L. Manion
University of Canterbury
Christchurch, New Zealand
steve.manion
@pg.canterbury.ac.nz

Raazesh Sainudiin
University of Canterbury
Christchurch, New Zealand
r.sainudiin
@math.canterbury.ac.nz

Abstract

We introduce Peripheral Diversity (PD) as a knowledge-based approach to achieve multilingual Word Sense Disambiguation (WSD). PD exploits the frequency and diverse use of word senses in semantic subgraphs derived from larger sense inventories such as BabelNet, Wikipedia, and WordNet in order to achieve WSD. PD's f-measure scores for SemEval 2013 Task 12 outperform the Most Frequent Sense (MFS) baseline for two of the five languages: English, French, German, Italian, and Spanish. Despite PD remaining under-developed and under-explored, it demonstrates that it is robust, competitive, and encourages development.

1 Introduction

By reading out aloud "A *minute* is a *minute* division of time" (Nelson, 1976), we can easily make the distinction between the two *senses* of the homograph *minute*. For a machine this is a complex task known as Word Sense Disambiguation (WSD). Task 12 of SemEval 2013 (Navigli et al., 2013) calls for a language-independent solution to WSD that utilises a multilingual sense inventory.

Supervised approaches to WSD have dominated for some time now (Màrquez et al., 2007). Homographs such as *minute* are effortlessly disambiguated and more polysemous words such as *bar* or *line* can also be disambiguated with reasonable competence (Agirre and Edmonds, 2007). However our approach is purely knowledge-based and employs semantic graphs. This allows us to avoid the notorious predicament Gale et al. (1992) name the *information bottleneck*, in which supervised approaches fail to be portable across alternative languages and domains if the annotated corpora do not exist. Conversely, knowledge-based approaches for WSD are usually applicable to all words in unrestricted text (Mihalcea, 2007). It is this innate scalability that motivates us to pursue knowledge-based approaches. Regardless of whether sense inventories can maintain *knowledge-richness* as they grow, their continued refinement by contributors is directly beneficial.

Knowledge-based approaches that employ semantic graphs increasingly rival leading supervised approaches to WSD. They can beat a Random or LESK (Lesk, 1986) baseline (*see* Mihalcea (2005), Navigli and Lapata (2007), Sinha and Mihalcea (2007), Navigli and Lapata (2010)) and can compete with or even beat the Most Frequent Sense (MFS) baseline in certain contexts which is by no means an easy task (*see* Navigli et al. (2007), Eneko Agirre and Aitor Soroa (2009), Navigli and Ponzetto (2012a)).

2 Methodology

PD is a framework for knowledge-based WSD approaches that employ semantic graphs. However before we can elaborate we must first cover the fundamental resources it is built upon.

2.1 Fundamental Resource Definitions

2.1.1 Lemma Sequences

At a glance across the text of any language, we absorb meaning and new information through its *lexical composition*. Depending on the length of text

we are reading, we could interpret it as one of many structural subsequences of writing such as a *paragraph*, *excerpt*, *quote*, *verse*, *sentence*, among many others. Let $\mathcal{W} = (w_a, ..., w_b)$ be this subsequence of words, which we will utilise as a sliding window for PD. Again let $\mathbb{W} = (w_1, ..., w_m)$ be the larger body of text of length m, such as a *book*, *newspaper*, or *corpus of text*, that our sliding window of length $b-a$ moves through.

In SemEval Task 12 on Multilingual Word Sense Disambiguation all words are *lemmatised*, which is the process of unifying the different inflected forms of a word so they can be analysed as a consolidated *lemma* (or *headword*). Therefore words (or *lexemes*) such as *runs* and *ran* are all mapped to their unifying lemma *run*[1].

To express this, let $\ell_w : \mathcal{W} \rightarrow \mathcal{L}$ be a *many-to-one* mapping from the sequence of words $\mathcal{W}$ to the sequence of lemmas $\mathcal{L}$, in which $(w_a, ..., w_b) \mapsto (\ell_{w_a}, ..., \ell_{w_b}) = (\ell_a, ..., \ell_b)$. To give an example from the test data set[2], the word sequence $\mathcal{W}$ = (*And, it, 's, nothing, that, runs, afoul, of, ethics, rules,* .) maps to the lemma sequence $\mathcal{L}$ = (*and, it, be, nothing, that, run, afoul, of, ethic, rule,* .). In order to complete this SemEval task we disambiguate a large sequence of lemmas $\mathbb{L} = (\ell_1, ..., \ell_m)$, via our lemma-based sliding window $\mathcal{L} = (\ell_a, ..., \ell_b)$.

2.1.2 Synsets

Each lemma $\ell_i \in \mathcal{L}$ may refer up to k senses in $S(\ell_i) = \{s_{i,1}, s_{i,2}, ..., s_{i,k}\} = \mathcal{S}$. Furthermore each sense $s_{i,j} \in \mathcal{S}$ maps to a set of unique concepts in the human lexicon. To clarify let us consider one of the earliest examples of modern ambiguity taken from Bar-Hillel's (1960) critique of Machine Translation: $\mathcal{W}$ = (*The, box, was, in, the, pen,* .). The sense of *pen* could be either *a)* a certain writing *utensil* or *b)* an *enclosure* where small children can play, therefore $\{s_{\text{enclosure}}, s_{\text{utensil}}\} \subset S(\ell_{\text{pen}}) = \mathcal{S}$. Humans can easily resolve the ambiguity between the possible senses of *pen* by accessing their own internal lexicon and knowledge of the world they have built up over time.

In the same vein, when accessing sense inventories such as BabelNet, WordNet (Fellbaum, 1998), and Wikipedia which are discrete representations of the human lexicon, we refer to each sense $s_{i,j} \in \mathcal{S}$ as a synset. Depending on the sense inventory the synset belongs to, it may contain alternative or translated lexicalisations, glosses, links to other semantic resources, among a collection of semantically defined relations to other synsets.

2.1.3 Subgraphs

PD makes use of subgraphs derived from a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that can be crafted from a sense inventory, such as BabelNet, WordNet, or Wikipedia. We construct subgraphs using the BabelNet API which accesses BabelNet[3] and Babel synset paths[4] indexed into Apache Lucene[5] to ensure speed of subgraph construction. This process is described in Navigli and Ponzetto (2012a) and demonstrated in Navigli and Ponzetto (2012b). Our formalisation of subgraphs is adapted into our own notation from the original papers of Navigli and Lapata (2007) and Navigli and Lapata (2010). We refer the reader to these listed sources if they desire an extensive explanation of our subgraph construction as we have built PD on top of the same code base therefore we do not deviate from it.

For a given lemma sequence $\mathcal{L} = (\ell_i, ..., \ell_n)$ and directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we construct our subgraph $\mathcal{G}_\mathcal{L} = (\mathcal{V}_\mathcal{L}, \mathcal{E}_\mathcal{L})$ in two steps:

1. Initialize $\mathcal{V}_\mathcal{L} := \bigcup_{i=1}^{n} S(\ell_i)$ and $\mathcal{E}_\mathcal{L} := \emptyset$.

2. For each node $v \in \mathcal{V}_\mathcal{L}$, we perform a depth-first search (DFS) of $\mathcal{G}$, such that, every time we encounter a node $v' \in \mathcal{V}_\mathcal{L}$ $(v' \neq v)$ along a path $v, v_1, ..., v_k, v'$ of length $\leq L$ in $\mathcal{G}$, we add all intermediate nodes and edges on the path from v to v', i.e., $\mathcal{V}_\mathcal{L} := \mathcal{V}_\mathcal{L} \cup \{v_1, ..., v_k\}$ and $\mathcal{E}_\mathcal{L} := \mathcal{E}_\mathcal{L} \cup \{\{v, v_1\}, ..., \{v_k, v'\}\}$.

2.2 Interpretation of Problem

For the lemmatisation of any word $w_i \mapsto \ell_i : w_i \in \mathcal{W}, \ell_i \in \mathcal{L}$, we must estimate the most appropriate synset $s_{i,*} \in S(\ell_i) = \{s_{i,1}, s_{i,2}, ..., s_{i,k}\}$. Our system associates a PD score $\phi(s_{i,j})$ for each

[1] While all words are lemmatised, this task strictly focuses on the WSD of noun phrases.

[2] This is sentence `d010.s014` in the English test data set.

[3] BabelNet 1.1.1 API & Sense Inventory - `http://lcl.uniroma1.it/babelnet/download.jsp`

[4] BabelNet 1.0.1 Paths - `http://lcl.uniroma1.it/babelnet/data/babelnet_paths.tar.bz2`

[5] Apache Lucene - `http://lucene.apache.org`

$s_{i,j} \in S(\ell_i)$ by taking $\mathcal{G}_\mathcal{L}$ as input. We estimate $s_{i,*}$, the most appropriate sense for ℓ_i, by $\hat{s}_{i,*} = \arg\max_{s_{i,j} \in S(\ell_i)} \phi(s_{i,j})$. It's worth noting here that $\mathcal{G}_\mathcal{L}$ ensures the estimation of $\hat{s}_{i,*}$ is not an independent scoring rule, since $\mathcal{G}_\mathcal{L}$ embodies the context surrounding ℓ_i via our sliding lemma-based window $\mathcal{L}$.

2.3 Peripheral Diversity Framework

PD is built on the following two ideas that are explained in the following subsections:

1. For a subgraph derived from one lone lemma ℓ_i, in which no other lemmas can provide context, the synset $s_{i,j} \in \mathcal{G}_{\ell_i}$ that has the largest and most semantically diverse set of peripheral synset nodes is assumed to be the MFS for ℓ_i.

2. For a larger subgraph derived from a sliding lemma window $\mathcal{L}$, in which other lemmas can provide context, the synset $s_{i,j} \in \mathcal{G}_\mathcal{L}$ that observes the largest increase in size and semantic diversity of its peripheral synset nodes is estimated to be $s_{i,*}$, the most appropriate synset for lemma ℓ_i.

Therefore PD is merely a framework that exploits these two assumptions. Now we will go through the process of estimating $s_{i,*}$ for a given lemma ℓ_i.

2.3.1 Pairwise Semantic Dissimilarity

First, for each synset $s_{i,j} \in \mathcal{S}$, we need to acquire a set of its peripheral synsets. We do this by travelling a depth of up to d (stopping if the path ends), then adding the synset we reach to our set of peripheral synsets $\mathcal{P}^{\leq d} = \{s_{j,1}, s_{j,2}, ..., s_{j,k'}\}$.

Next for every pair of synsets v and v' that are not direct neighbours in $\mathcal{P}^{\leq d}$ such that $v \neq v'$, we calculate their Pairwise Semantic Dissimilarity (PSD) $\delta(v, v')$ which we require for a synset's PD score. To generate our results for this task we have used the complement to Cosine Similarity, commonly known as the Cosine Distance as our PSD measure:

$$\delta(v, v') = \begin{cases} 1 - \left(\frac{|O(v) \cap O(v')|}{\sqrt{|O(v)|}\sqrt{|O(v')|}} \right), & \text{if } |O(v)||O(v')| \neq 0 \\ 1, & \text{otherwise,} \end{cases}$$

where $O(v)$ is the outgoing (out-neighbouring) synsets for $v \in \mathcal{P}^{\leq d}$, and $|O(v)|$ denotes the number of elements in $O(v)$.

2.3.2 Peripheral Diversity Score

Once we have PSD scores for every permitted pairing of v and v', we have a number of ways to generate our $\phi(s_{i,j})$ values. To generate our results for this task, we chose to score synsets on the *sum of their minimum PSD values*, which is expressed formally below:

$$\phi(s_{i,j}) = \sum_{v \in \mathcal{P}^{\leq d}(s_{i,j})} \min_{\substack{v' \neq v \\ v' \in \mathcal{P}^{\leq d}(s_{i,j})}} \delta(v, v')$$

The idea is that this summing over the peripheral synsets in $\mathcal{P}^{\leq d}(s_{i,j})$ accounts for how frequently synset $s_{i,j}$ is used, then each increment in size is weighted by a peripheral synset's minimum PSD across all synsets in $\mathcal{P}^{\leq d}(s_{i,j})$. Therefore peripheral set size and semantic diversity are rewarded simultaneously by ϕ. To conclude, the final estimated synset sequence for a given lemma sequence $(\ell_1, ..., \ell_m)$ based on ϕ is $(\hat{s}_{1,*}, \hat{s}_{2,*}, ..., \hat{s}_{m,*})$.

2.3.3 Strategies, Parameters, & Filters

Wikipedia's *Did You Mean?* We account for deviations and errors in spelling to ensure lemmas have the best chance of being mapped to a synset. Absent synsets in subgraph $\mathcal{G}_\mathcal{L}$ will naturally degrade system output. Therefore if $\ell_i \mapsto \emptyset$, we make an HTTP call to Wikipedia's *Did you mean?* and parse the response for any alternative spellings. For example in the test data set[6] the misspelt lemma: "feu_de_la_rampe" is corrected to "*feux*_de_la_rampe".

Custom Back-off Strategy As *back-off strategies*[7] have proved useful in (Navigli and Ponzetto, 2012a) and (Navigli et al., 2007), we designed our own back-off strategy. In the event our system provides a `null` result, the Babel synset $s_{i,j} \in S(\ell_i) = \mathcal{S}$ with the most senses associated with it will be chosen with preference to its region in BabelNet such that WIKIWN $\succ$ WN $\succ$ WIKI.

[6] Found in sentence `d001.s002.t005` in the French test data set.

[7] In the event the WSD technique fails to provide an answer, a back-off strategy provides one for the system to output.

Input Parameters We set our sliding window length ($b - a$) to encompass 5 sentences at a time, in which the step size is also 5 sentences. For subgraph construction the maximum length $L = 2$. Finally we set our peripheral search depth $d = 3$.

Filters For the purposes of reproducibility only we briefly mention two filters we apply to our subgraphs that ship with the BabelNet API. We remove WordNet contributed domain relations with the `ILLEGAL_POINTERS` filter and apply the `SENSE_SHIFTS` filter. For more information on these filters we suggest the reader consult the BabelNet API documentation.

3 Results & Discussion

3.1 Results of SemEval Submission

	Language	DAEBAK!	$\text{MFS}_{\text{Baseline}}$	+/-
DE	*German*	59.10	68.60	-9.50
EN	*English*	60.40	65.60	-5.20
ES	*Spanish*	60.00	64.40	-4.40
FR	*French*	53.80	50.10	+3.70
IT	*Italian*	61.30	57.20	+4.10
	Mean	58.92	61.18	-2.26

Table 1: DAEBAK! vs MFS Baseline on BabelNet

As can be seen in Table 1, the results of our single submission were varied and competitive. The worst result was for German in which our system fell behind the MFS baseline by a margin of 9.50. Again for French and Italian we exceeded the MFS baseline by a margin of 3.70 and 4.10 respectively. Our Daebak back-off strategy contributed anywhere between 1.12% (for French) to 2.70% (for Spanish) in our results, which means our system outputs a result without the need for a back-off strategy at least 97.30% of the time. Overall our system was slightly outperformed by the MFS baseline by a margin of 2.26. Overall PD demonstrated to be robust across a range of European languages. With these preliminary results this surely warrants further investigation of what can be achieved with PD.

3.2 Exploratory Results

The authors observed some inconsistencies in the task answer keys across different languages as Table 2 illustrates. For each Babel synset ID found in the answer key, we record where its original source synsets are from, be it Wikipedia (WIKI), WordNet (WN), or both (WIKIWN).

	Language	WIKI	WN	WIKIWN
DE	*German*	43.42%	5.02%	51.55%
EN	*English*	10.36%	32.11%	57.53%
ES	*Spanish*	30.65%	5.40%	63.94%
FR	*French*	40.81%	6.55%	52.64%
IT	*Italian*	38.80%	7.33%	53.87%

Table 2: BabelNet Answer Key Breakdown

This is not a critical observation but rather an empirical enlightenment on the varied mechanics of different languages and the amount of development/translation effort that has gone into the contributing subparts of BabelNet: Wikipedia and WordNet. The heterogeneity of hybrid sense inventories such as BabelNet creates new obstacles for WSD, as seen in (Medelyan et al., 2013) it is difficult to create a disambiguation policy in this context. Future work we would like to undertake would be to investigate the heterogeneous nature of BabelNet and how this affects various WSD methods.

4 Conclusion & Future Directions

To conclude PD has demonstrated in its early stages that it can perform well and even outperform the MFS baselines in certain experimental contexts. Furthermore it leaves a lot left to be explored in terms of what this approach is capable of via adjusting subgraph filters, strategies, and input parameters across both heterogeneous and homogeneous semantic graphs.

Acknowledgments

This research was completed with the help of the Korean Foundation Graduate Studies Fellowship[8].

5 Resources

The code base for this work can be found in the near future at `http://www.stevemanion.com/`.

[8]KF Graduate Studies Fellowship - `http://www.kf.or.kr/eng/01_sks/sks_fel_sfb01.asp`

References

Eneko Agirre and Philip Edmonds. 2007. Introduction. *Word Sense Disambiguation Algorithms and Applications*, Chapter 1:1-28. Springer, New York.

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. *In Proceedings of the 12th Conference of the European Chapter of the ACL*, April:33–41. Association for Computational Linguistics.

Yehoshua Bar-Hillel. 1960. The Present Status of Automatic Translation of Languages. *Advances in Computers*, 1:91–163.

Christiane Fellbaum. 1998, ed. *WordNet: An Electronic Lexical Database.*, Cambridge, MA: MIT Press.

William A Gale, Kenneth W Church, David Yarowsky. 1992. A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, 26(5–6):415–439.

Michael Lesk. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. *Proceedings of the 5th Annual International Conference on System Documentation.*, 24–26. ACM.

Llus Màrquez, Gerard Escudero, David Martínez, German Rigau. 2007. Supervised Corpus-Based Methods for WSD. *Word Sense Disambiguation Algorithms and Applications*, Chapter 7:167-216. Springer, New York.

Rada Mihalcea. 2005. Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 411-418. Association for Computational Linguistics.

Rada Mihalcea. 2007. Knowledge-Based Methods for WSD. *Word Sense Disambiguation Algorithms and Applications*, Chapter 5:107–131. Springer, New York.

Alyona Medelyan, Steve Manion, Jeen Broekstra, Anna Divoli, Anna-lan Huang, and Ian H Witten. 2013. Constructing a Focused Taxonomy from a Document Collection *Extended Semantic Web Conference*, (Accepted, in press)

Roberto Navigli and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. *IJCAI'07 Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 1683–1688.

Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. 2007. SemEval-2007 Task 07: Coarse-Grained English All-Words Task. *In Proceedings of the 4th International Workshop on Semantic Evaluations*, 30–35.

Roberto Navigli and Mirella Lapata. 2010. An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):678–692.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Multilingual WSD with Just a Few Lines of Code: the BabelNet API. *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 67–72.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (*SEM 2013)*.

Frederic Nelson. 1976. Homographs *American Speech*, 51(3):296–297.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. *Proceedings of IEEE International Conference on Semantic Computing*.

An Iterative 'Sudoku Style' Approach to Subgraph-based Word Sense Disambiguation

Steve L. Manion
University of Canterbury
Christchurch, New Zealand
steve.manion
@pg.canterbury.ac.nz

Raazesh Sainudiin
University of Canterbury
Christchurch, New Zealand
r.sainudiin
@math.canterbury.ac.nz

Abstract

We introduce an *iterative* approach to subgraph-based Word Sense Disambiguation (WSD). Inspired by the Sudoku puzzle, it significantly improves the *precision* and *recall* of disambiguation. We describe how *conventional* subgraph-based WSD treats the two steps of (1) subgraph construction and (2) disambiguation via graph centrality measures as ordered and atomic. Consequently, researchers tend to focus on improving either of these two steps individually, overlooking the fact that these steps can complement each other if they are allowed to interact in an iterative manner. We tested our iterative approach against the conventional approach for a range of well-known graph centrality measures and subgraph types, at the sentence and document level. The results demonstrated that an average performing WSD system which embraces the iterative approach, can easily compete with state-of-the-art. This alone warrants further investigation.

1 Introduction

Explicit WSD is a two-step process of analysing a word's contextual use then deducing its intended sense. When Kilgarriff (1998) established SENSEVAL, the collaborative framework and forum to evaluate WSD, unsupervised systems performed poorly in comparison to their supervised counterparts (Palmer et al., 2001; Snyder and Palmer, 2004). A review of the literature shows there has been a healthy *rivalry* between the two, in which proponents of unsupervised WSD have long sought to vindicate its potential since two decades ago (Yarowsky, 1995) to even more recent times (Ponzetto and Navigli, 2010).

As Pedersen (2007) rightly states, supervised systems are bound by their training data, and therefore are limited in portability and flexibility in the face of new domains, changing applications, or different languages. This *knowledge acquisition bottleneck*, coined by Gale et al. (1992), can be alleviated by unsupervised systems that exploit the portability and flexibility of Lexical Knowledge Bases (LKBs). As of 2007, SENSEVAL became SEMEVAL, offering a more diverse range of semantic tasks. Unsupervised knowledge-based WSD has since had its performance evaluated in terms of *granularity* (Navigli et al., 2007), *domain* (Agirre et al., 2010), and *cross/multi-linguality* (Lefever and Hoste, 2010; Lefever and Hoste, 2013; Navigli et al., 2013). Results from these tasks have demonstrated unsupervised systems are now a competitive and robust alternative to supervised systems, especially given the ever changing task-orientated settings WSD is evaluated in.

One such class of unsupervised knowledge-based WSD systems that we seek to improve in this paper constructs semantic subgraphs from LKBs, and then runs graph-based centrality measures such as PageRank (Brin and Page, 1998) over them to finally select the senses (as nodes) ranked as the most relevant. This class is known as *subgraph-based* WSD, characterised over the last decade by performing the two key steps of (1) subgraph construction and (2) disambiguation via graph centrality measures, in an ordered atomic sequence. We refer to this characteristic as the *conventional* approach to subgraph-based WSD. We propose an *iterative* approach to subgraph-based WSD that allows for interaction between the two major steps in an incremental manner

and demonstrate its effectiveness across a range of graph-based centrality measures and subgraph construction methods at the sentence and document levels of disambiguation.

2 The Conventional Subgraph Approach

The *conventional* approach to subgraph WSD firstly benefits from some preprocessing, in which words in a sequence $\mathcal{W}$, are mapped to their lemmatisations[1] in a set $\mathcal{L}$, such that $(w_1, ..., w_m) \mapsto \{\ell_1, ..., \ell_m\}$. This facilitates better lexical alignment with the LKB to be exploited. Let this LKB be a large semantic graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, such that $\mathcal{S}$ is a set of vertices representing all known word senses, and $\mathcal{E}$ be a set of edges defining semantic relationships that exist between senses. Now given we wish to disambiguate $\ell_i \in \mathcal{L}$, let $R(\ell_i)$ be a function that *Retrieves* from $\mathcal{G}$, all the senses, $\{s_{i,1}, s_{i,2}, ..., s_{i,k}\}$, that ℓ_i could refer to, noting that i is an anchor to the original word w_i.

2.1 Step 1: Subgraph Construction

For unsupervised subgraph-based WSD, the key publications that have advanced the field broadly construct subgraph, $\mathcal{G}_\mathcal{L}$, as either a union of *subtree paths*, *shortest paths*, or *local edges*[2]. First we initialise $\mathcal{G}_\mathcal{L}$, by setting $\mathcal{S}_\mathcal{L} := \bigcup_{i=1}^{n} R(\ell_i)$ and $\mathcal{E}_\mathcal{L} := \emptyset$. Next we add edges to $\mathcal{E}_\mathcal{L}$, depending on the desired subgraph type, by adding either the:

(a) *Subtree paths* of up to length L, via a Depth-First Search (DFS) of $\mathcal{G}$. In brief, **for each** sense $s_a \in \mathcal{S}_\mathcal{L}$, **if** a new sense $s_b \in \mathcal{S}_\mathcal{L}$, i.e. $s_b \neq s_a$, is encountered along a path $P_{a \to b} = \{\{s_a, s\}, ..., \{s', s_b\}\}$ with path-length $|P_{a \to b}| \leq L$, **then** add $P_{a \to b}$ to $\mathcal{G}_\mathcal{L}$. [cf. Navigli and Velardi (2005), Navigli and Lapata (2007), or Navigli and Lapata (2010)]

(b) *Shortest paths*, via a Breadth-First Search (BFS) of $\mathcal{G}$. In brief, **for each** sense pair $s_a, s_b \in \mathcal{S}_\mathcal{L}$, find the shortest path $P_{a \to b} = \{\{s_a, s\}, ..., \{s', s_b\}\}$; **if** such a path $P_{a \to b}$ exists and (optionally) $|P_{a \to b}| \leq L$, **then** add $P_{a \to b}$ to $\mathcal{G}_\mathcal{L}$ [cf. Agirre and Soroa (2008), Agirre and Soroa (2009), or Gutiérrez et al. (2013)]

(c) *Local edges* up to a local distance D. In brief, **for each** sense pair $s_a, s_b \in \mathcal{S}_\mathcal{L}$, **if** the distance in the text $|b - a|$ between the corresponding words w_a and w_b satisfies $|b - a| \leq D$, **then** add edge $\{s_a, s_b\}$ to $\mathcal{G}_\mathcal{L}$ (preferably with edge-weights). [cf. Mihalcea (2005) or Sinha and Mihalcea (2007)] (Note that this subgraph is a hybrid, because only its vertices belong to $\mathcal{G}$)

In practice, subgraph edges may be *directed*, *weighted*, *collapsed*, or *filtered*. However to keep the distinctions between subgraph types simple, we do not include this in our formalisation.

2.2 Step 2: Disambiguation

To disambiguate each lemma $\ell_i \in \mathcal{L}$, its corresponding senses, $R(\ell_i) = \{s_{i,1}, s_{i,2}, ..., s_{i,k}\}$, are scored by a graph-based centrality measure ϕ, over subgraph $\mathcal{G}_\mathcal{L}$, to estimate the most appropriate sense, $\hat{s}_{i,*} = \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$. The estimated sense $\hat{s}_{i,*}$ is then assigned to word w_i.

2.3 Algorithm for Conventional Approach

With both steps formalised, we can now illustrate the conventional subgraph approach in Algorithm 1. Let $\mathcal{L}$ be taken as *input*, and let the disambiguation results $\mathcal{D} = \{\hat{s}_{1,*}, ..., \hat{s}_{m,*}\}$ be produced as *output* to assign to $\mathcal{W} = (w_1, ..., w_m)$.

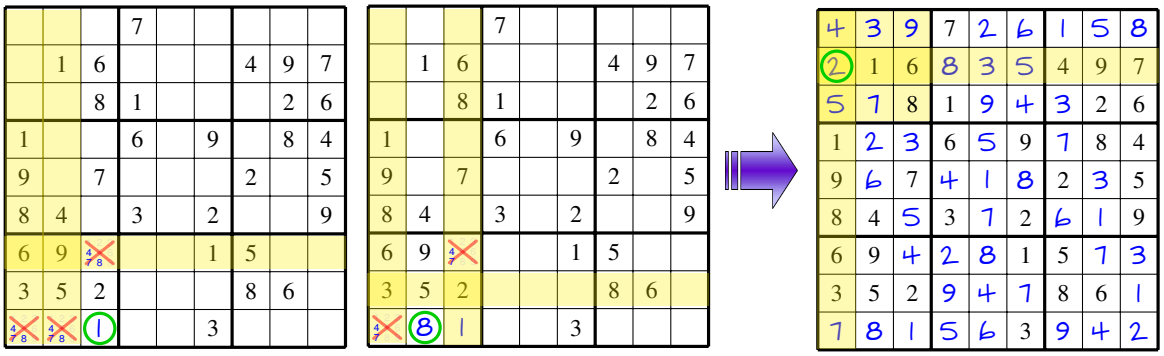
Algorithm 1: Conventional Approach

Input: $\mathcal{L}$
Output: $\mathcal{D}$
$\mathcal{D} \leftarrow \emptyset$;
$\mathcal{G}_\mathcal{L} \leftarrow$ `ConstructSubGraph` $(\mathcal{L})$;
foreach $\ell_i \in \mathcal{L}$ **do**
  $\hat{s}_{i,*} \leftarrow \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$;
  put $\hat{s}_{i,*}$ in $\mathcal{D}$;

To begin with, $\mathcal{D}$ is initialised as an empty set and `ConstructSubGraph`$(\mathcal{L})$ constructs one of the three subgraphs described in section 2.1. Next for each $\ell_i \in \mathcal{L}$, by running a graph based centrality measure ϕ over $\mathcal{G}_\mathcal{L}$, the most appropriate sense $\hat{s}_{i,*}$ is estimated, and placed in set $\mathcal{D}$. Effectively, $\mathcal{L}$ is a context window based on document or sentence size, therefore this algorithm is run for each context window division. Note that Algorithm 1 would require a little extra complexity to handle local edge subgraphs, due to its context window needing to satisfy $\mathcal{L} = \{\ell_{i-D}, ..., \ell_{i+D}\}$.

[1] For a detailed explanation of the processes leading up to lemmatisation (and beyond), *see* Navigli (2009, p12)

[2] 'Local' describes the *local context*, typically this is the 2 or 3 words either side of a word, *see* Yarowsky (1993)



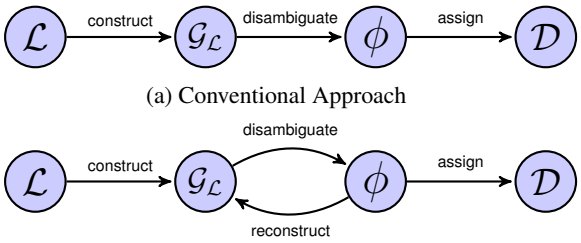
(a) 1st Row/Column Elimination (b) 2nd Row/Column Elimination (c) Row/Column/Box Completion

Figure 1: Iterative Solving of Sudoku Grids

3 The Iterative Subgraph Approach

3.1 What is Iterative WSD?

The key observation to make about the conventional approach in Algorithm 1, is for input $\mathcal{L}$, constructing subgraph $\mathcal{G}_\mathcal{L}$ and performing disambiguation are two ordered atomic steps. Notice that there is no iteration between them, because the first step of subgraph construction is never revisited for each $\mathcal{L}$. For the conventional process to be iterative, then for $\ell_a, \ell_b \in \mathcal{L}$ a previous disambiguation of ℓ_a, would need to influence a consecutive disambiguation of ℓ_b, through an iterative re-construction of $\mathcal{G}_\mathcal{L}$ between each disambiguation. This key difference illustrated by Figure 2, is the level of iterative WSD we aspire to.



(a) Conventional Approach

(b) Interactively Iterative Approach

Figure 2: The Key Difference In Approach

It is important to note, the term *iterative* can already be found in WSD literature, therefore we take the opportunity here to make a distinction. Firstly, a graph based centrality measure ϕ may be iterative, such as PageRank (Brin and Page, 1998) or Hyperlink-Induced Topic Search (HITS) (Kleinberg, 1999). In the experiments by Mihalcea (2005) in which PageRank was run over *local edge* subgraphs (as described in 2.1 (c)), it is easy to perceive the WSD process itself as iterative.

Iteration can again be taken further, as observed with Personalised PageRank in which Agirre and Soroa (2009) apply the idea of biasing values in the random surfing vector, v, (*see* (Haveliwala, 2003)). For their run labelled "Ppr_w2w", in order to avoid senses anchored to the same lemma assisting each other's ϕ score, the random surfing vector v is iteratively updated as ℓ_i changes, to ensure context senses $s_{a,j} \in v$ such that $a \neq i$ are the only senses that receive probability mass.

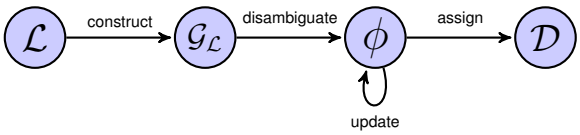
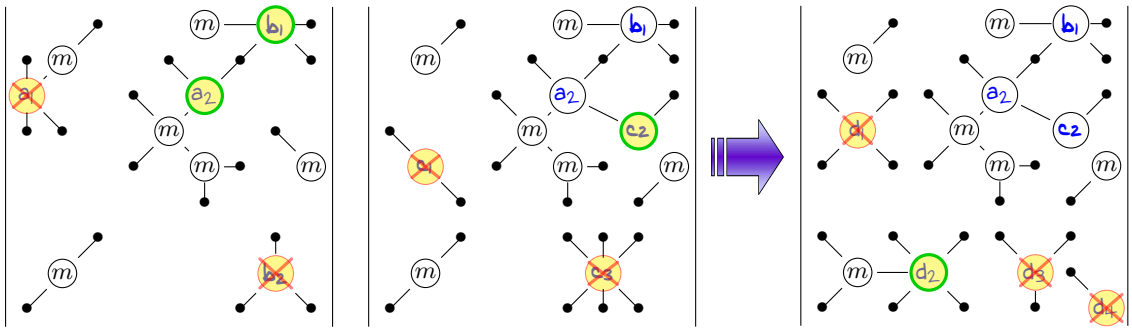


Figure 3: Atomically Iterative Approach

In summary, iteration in the literature either describes ϕ as being iterative or being iteratively adjusted, both of which are contained in the disambiguation step alone as shown in Figure 3. This is iteration at the atomic level and should not be conflated with the interactive level of iteration that we propose as seen in Figure 2 (b).

3.2 Iteratively Solving a Sudoku Grid

In Figures 1 (a), (b), and (c), we observe the solving of a Sudoku puzzle, in which the numbers from 1 to 9 must be assigned only once to each *column*, *row*, and 3x3 *square*. Each time a number is assigned and the Sudoku grid is updated, this is an *iteration*. For example, in the south west square of grid (a) (i.e. Figure 1 (a)) unknown cells can be assigned $\{1, 4, 7, 8\}$. Given that 1 has already been assigned to the 7th row and the 1st and 2nd columns, this singles it down to one cell it can be assigned to. The iteration of grid



(a) x2 Bi-semous Eliminations (b) x1 Tri-semous Elimination (c) (ρ_{max})-semous Completion

Figure 4: Iterative Disambiguating of Subgraphs

(a), now makes possible the iteration of grid (b) to eliminate the number 8 as the only possibility for its assigned cell. This iterative process continues until we reach the completed puzzle in grid (c). Therefore in WSD terminology, with each cell we *disambiguate*, a new grid is *constructed*, in which knowledge is passed on to each consecutive iteration.

Continuing with this line of thought, each unsolved cell is *ambiguous*, with a degree of *polysemy* ρ, such that $\rho_{max} \leq 9$. Again, the initial Sudoku grid has pre-solved cells, of which are *monosemous*. This brings us to another key observation. Typically in Sudoku, it is necessary to solve the least polysemous cells first, before you can solve the more polysemous cells with a certainty. As the conventional approach exhibits no Sudoku-like iteration, cells are solved without regard to the ρ value of the cell, or any interactive exploitation of previously solved cells.

3.3 Iteratively Constructing a Subgraph

In our 'Sudoku style' approach, we propose disambiguating each ℓ_i in order of increasing polysemy ρ, iteratively reconstructing subgraph $\mathcal{G}_\mathcal{L}$ to reflect 1) previous disambiguations and 2) the ρ value of lemmas being disambiguated in the current iteration. This is illustrated in Figures 4 (a), (b), and (c) above.

Let m-labelled vertices describe monosemous lemmas. In graph (a) (i.e. Figure 4) we observe two bi-semous lemmas, a and b, in which our arbitrary graph-based centrality measure ϕ has selected the second sense of a (i.e. a_2) and the first sense of b (i.e. b_1) to be placed in $\mathcal{D}$. For the next iteration, you will notice the alternative senses for a and b are removed from $\mathcal{G}_\mathcal{L}$ for the disambiguation of tri-semous lemma c. The second sense of lemma c manages to be selected by ϕ with the help of the previous disambiguation of lemma a. This interactive and iterative process continues until we reach the most polysemous lemma, which in our example is d with $\rho_{max} = 4$ in graph (c).

3.4 Algorithm for Iterative Approach

We can formally describe what is happening in Figure 4 with Algorithm 2. Effectively, this is a recreation of Algorithm 1, which highlights the differences in the conventional and iterative approach.

Algorithm 2: Iterative Approach

Input: $\mathcal{L}$
Output: $\mathcal{D}$
$\mathcal{D} \leftarrow$ `GetMonosemous` $(\mathcal{L})$;
$\mathcal{A} \leftarrow \emptyset$;
for $\rho \leftarrow 2$ **to** ρ_{max} **do**
  $\mathcal{A} \leftarrow$ `AddPolysemous` $(\mathcal{L}, \rho)$;
  $\mathcal{G}_\mathcal{L} \leftarrow$ `ConstructSubGraph` $(\mathcal{A},\mathcal{D})$;
  foreach $\ell_i \in \mathcal{A}$ **do**
    $\hat{s}_{i,*} \leftarrow \arg\max_{s_{i,j} \in R(\ell_i)} \phi(s_{i,j})$;
    if $\hat{s}_{i,*}$ *exists* **then**
      remove ℓ_i from $\mathcal{A}$;
      put $\hat{s}_{i,*}$ in $\mathcal{D}$;

Firstly, as it reads `GetMonosemous`$(\mathcal{L})$ places all the senses of the monosemous lemmas into the set of *disambiguated* lemmas $\mathcal{D}$. This is the equivalent of copying out an unsolved Sudoku grid onto a piece of paper and adding in all the initial hint numbers. Next the set $\mathcal{A}$ which holds all *ambiguous* lemmas of polysemy $\leq \rho$ is initialised as an empty set. Now we are ready to iterate through values of ρ, beginning from the first iteration, by adding all bi-semous lemmas to

$\mathcal{A}$ with the function `AddPolysemous`$(\mathcal{L}, \rho)$, notice ρ places a restriction on the degree of polysemy a lemma $\ell_i \in \mathcal{L}$ can have before being added to $\mathcal{A}$.

We are now ready to create the first subgraph $\mathcal{G}_\mathcal{L}$ with function `ConstructSubGraph`$(\mathcal{A}, \mathcal{D})$. This previously used function in Algorithm 1, is now modified to take the ambiguous lemmas of polysemy $\leq \rho$ in set $\mathcal{A}$ and previously disambiguated lemma senses in set $\mathcal{D}$. The resulting graph has a limited degree of polysemy and is constructed based on previous disambiguations.

From this point on the given graph centrality measure ϕ is run over $\mathcal{G}_\mathcal{L}$. For the lemmas that are disambiguated, they are removed from $\mathcal{A}$ and the selected sense is added to $\mathcal{D}$. For those lemmas that are not (i.e. $\hat{s}_{i,*}$ does not exist[3]) they remain in $\mathcal{A}$ to be involved in reattempted disambiguations in consecutive iterations. As more lemmas are disambiguated, it is more likely that previously difficult to disambiguate lemmas become much easier to solve, just like at the end of a Sudoku puzzle it gets easier as you get closer to completing it.

4 Evaluations

In our evaluations we set out to understand a number of aspects. The first evaluation is a *proof of concept*, to understand whether an iterative approach to subgraph WSD can in fact achieve better performance than the conventional approach. The second set of experiments seeks to understand how the iterative approach works and the performance *benefits* and *penalties* of implementing the iterative approach. Finally the third experiment is an *elementary attempt* at optimising the iterative approach to defeat the MFS baseline.

4.1 LKB & Dataset

For an evaluation, we have chosen the multilingual LKB known as BabelNet (Navigli and Ponzetto, 2012a). It weaves together several other LKBs, most notably WordNet (Fellbaum, 1998) and Wikipedia. It also can be easily accessed with the BabelNet API, of which we have built our code base around. All experiments are conducted on the most recent SemEval WSD dataset, of which is the SemEval 2013 Task 12 Multilingual WSD (English) data set.

[3]This can happen if ℓ_i does not map to any senses, or alternatively all the senses that are mapped to are filtered out of the subgraph before disambiguation (explained later).

4.2 Graph Centrality Measures Evaluated

To demonstrate the effectiveness of our iterative approach, we selected a range of WSD graph-based centrality measures often experimented with in the literature. Firstly ϕ does not need to be a complicated measure, this is demonstrated by the success of ranking senses by their number of incoming and outgoing edges. Even though it is very simple, it performs surprisingly well against others for both In-Degree (Navigli and Lapata, 2007) and Out-Degree (Navigli and Ponzetto, 2012a)

Next we employ graph centrality measures that are primarily used to disambiguate the *semantic web*, such as PageRank (Brin and Page, 1998), HITS Kleinberg (1999), and a *personalised* PageRank (Haveliwala, 2003); which have since been applied to WSD by Mihalcea (2005), Navigli and Lapata (2007), and Agirre and Soroa (2009) respectively. We also include Betweeness Centrality (Freeman, 1979) which is taken from the analysis of social networks.

These methods are well known and applied across many disciplines, therefore we will leave it to the reader to follow up on the specifics of these graph centrality measures. However we do explicitly define our last measure, Sum Inverse Path Length (Navigli and Ponzetto, 2012a; Navigli and Ponzetto, 2012b) in Equation (1) which was designed with WSD in mind, thus is less well known.

$$\phi(s) = \sum_{p \in P_{s \to c}} \frac{1}{e^{|p|-1}} \quad (1)$$

This measure scores a sense by summing up the scores of all paths that connect to other senses in $\mathcal{G}_\mathcal{L}$ (i.e. senses that are not intermediate nodes, but have a mapping back to a lemma in the context window $\mathcal{L}$). In the words of Navigli and Ponzetto (2012a), $P_{s \to c}$ is the set of paths connecting s to other senses of context words, with $|p|$ as the number of edges in the path p and each path is scored with the exponential inverse decay of the path length.

4.3 Experiment 1: Proof of Concept

4.3.1 Experiment 1: Setup

For this experiment we simply set out to see how the iterative approach performed compared to the conventional approach in a range of experimental conditions. Directed and unweighted subgraphs were used, namely subtree paths and shortest paths subgraphs with $L = 2$. To address the issue of

$\mathcal{G}_\mathcal{L}$	ϕ	Conventional Doc			Iterative Doc			Improvement		
		P	R	F	P	R	F	ΔP	ΔR	ΔF
SubTree Paths	In-Degree	**61.70**	**55.51**	**58.44**	**65.39**	**63.74**	**64.55**	+3.69	+8.23	+6.11
	Out-Degree	54.23	48.78	51.36	57.70	56.23	56.96	+3.47	+7.45	+5.59
	Betweenness Centrality	59.29	53.34	56.15	63.43	61.82	62.61	+4.14	+8.48	+6.46
	Sum Inverse Path Length	56.58	50.90	53.59	58.86	57.37	58.11	+2.28	+6.47	+4.51
	HITS(hub)	54.69	49.20	51.80	59.71	58.20	58.95	**+5.03**	**+9.00**	**+7.15**
	HITS(authority)	57.45	51.68	54.41	61.62	60.06	60.83	+4.18	+8.38	+6.42
	PageRank	60.09	54.06	56.92	64.07	62.44	63.24	+3.97	+8.38	+6.33
Shortest Paths	In-Degree	**63.06**	**56.08**	**59.36**	65.36	63.06	64.19	+2.30	+6.98	+4.83
	Out-Degree	57.07	50.75	53.72	61.14	58.92	60.01	+4.07	+8.17	+6.29
	Betweenness Centrality	60.33	53.65	56.79	**65.52**	**63.22**	**64.35**	**+5.20**	**+9.57**	**+7.56**
	Sum Inverse Path Length	57.53	51.16	54.16	61.19	58.98	60.06	+3.66	+7.81	+5.90
	HITS(hub)	57.48	51.11	54.11	62.14	59.96	61.03	+4.67	+8.85	+6.92
	HITS(authority)	60.91	54.16	57.34	63.54	61.30	62.40	+2.63	+7.14	+5.06
	PageRank	60.33	53.65	56.79	64.83	62.55	63.67	+4.50	+8.90	+6.87

Table 1: Improvements of using the Iterative Approach at the Document Level

$\mathcal{G}_\mathcal{L}$	ϕ	Conventional Sent			Iterative Sent			Improvement		
		P	R	F	P	R	F	ΔP	ΔR	ΔF
SubTree Paths	In-Degree	**60.83**	**50.70**	**55.30**	**61.80**	**56.23**	**58.88**	+0.96	+5.54	+3.58
	Out-Degree	56.18	46.82	51.07	59.64	54.11	56.74	+3.46	+7.29	+5.67
	Betweenness Centrality	59.40	49.51	54.01	61.66	56.08	58.74	+2.26	+6.57	+4.73
	Sum Inverse Path Length	56.67	47.23	51.52	59.45	54.01	56.60	+2.78	+6.78	+5.08
	HITS(hub)	55.49	46.25	50.45	59.51	54.06	56.65	**+4.02**	**+7.81**	**+6.20**
	HITS(authority)	56.80	47.34	51.64	60.30	54.84	57.44	+3.50	+7.50	+5.80
	PageRank	59.71	49.77	54.29	60.56	55.04	57.67	+0.84	+5.28	+3.38
Shortest Paths	In-Degree	**58.13**	**32.75**	**41.89**	63.79	42.11	50.73	+5.66	+9.36	+8.84
	Out-Degree	54.64	30.78	39.38	61.79	40.66	49.05	**+7.15**	+9.88	+9.67
	Betweenness Centrality	57.94	32.64	41.76	**64.11**	**42.32**	**50.98**	+6.17	+9.68	+9.22
	Sum Inverse Path Length	55.65	31.35	40.11	62.39	41.02	49.50	+6.74	+9.67	+9.39
	HITS(hub)	56.11	31.61	40.44	62.74	41.28	49.80	+6.63	+9.67	+9.36
	HITS(authority)	55.74	31.40	40.17	62.75	41.39	49.88	+7.01	**+9.98**	**+9.70**
	PageRank	56.84	32.02	40.97	63.17	41.70	50.23	+6.33	+9.67	+9.27

Table 2: Improvements of using the Iterative Approach at the Sentence Level

senses anchored to the same lemma assisting each other's ϕ score (as discussed in Section 3.1), the `SENSE_SHIFTS` filter that is provided by the BabelNet API was also applied. This filter removes any path $P_{a\to b}$ such that $s_a, s_b \in R(\ell_i)$. Disambiguation was attempted at the document and sentence level, making use of the eight well-known graph centrality measures listed in section 4.2. For this experiment no means of optimisation were applied. Therefore Personalised PageRank was not used, and traditional PageRank took on a uniform random surfing vector. Default values of 0.85 and 30 for damping factor and maximum iterations were set respectively.

4.3.2 Experiment 1: Observations

First and foremost, it is clear from Table 1 and 2 that the iterative approach outperforms the conventional approach, regardless of the subgraph used, level of disambiguation, or the graph centrality measure employed. Since no graph centrality measure or subgraph were optimised, let this experiment prove that the iterative approach has the potential to improve any WSD system that implements it.

At the document level for both subgraphs the F-Scores were very close to the Most Frequent Sense (MFS) baseline for this task of 66.50. It is notoriously hard to beat and only one team (Gutiérrez et al., 2013) managed to beat it for this task. For all subtree subgraphs, we observe that In-Degree is clearly the best choice of centrality measure, while HITS (hub) enjoys the most improvement. We also observe that applying the iterative approach to Betweenness Centrality on shortest paths is a great combination at both the document and sentence level, most probably due to the measure being based on shortest paths. Furthermore it is

worth noting, the results at the sentence level for all graph centrality measures on shortest path subgraphs are quite poor, but highly improved, this is likely to our restriction of $L = 2$ causing the subgraphs to be much sparser and broken up into many components.

We also provide here an example from the data set in which the incorrect disambiguation of the lemma *cup* via the conventional approach was corrected by the iterative approach. This example is the seventh sentence in the eleventh document (`d011.s007`). Each word's degree of polysemy is denoted in square brackets.

"Spanish **[1]football players** playing in the All-Star **[4]League** and in powerful **[12]clubs** of the **[2]Premier League** of **[9]England** are during the **[5]year** very active in **[4]league** and local **[8]cup** **[7]competitions** and there are high-level **[25]shocks** in the **[10]European Cups** and **[2]European Champions League**."

The potential graph constructed from this sentence is illustrated in Figure 5 as a shortest paths subgraph. The darker edges portray the subgraph iteratively constructed up to a polysemy $\rho \leq 8$ (in order to disambiguate *cup*), whereas the lighter edges portray the greater subgraph constructed if the conventional approach is employed. Note that although the lemma *cup* has eight senses, only three are shown due to the application of the previously mentioned `SENSE_SHIFTS` filter. The remaining five senses of *cup* were filtered out since they were not able to link to a sense up to $L = 2$ hops away that is anchored to an alterative lemma.

- **cup#1** - A small open container usually used for drinking; usually has a handle.
- **cup#7** - The hole (or metal container in the hole) on a golf green.
- **cup#8** - A large metal vessel with two handles that is awarded as a trophy to the winner of a competition.

Given the context, the eighth sense of cup is the correct sense, the type we know as a trophy. For the conventional approach, if ϕ is a centrality measure of Out-Degree then the eighth sense of cup is easily chosen by having one extra outgoing edge than the other two senses for *cup*. Yet if ϕ is a centrality measure of In-Degree or Betweenness Centrality, all three senses of cup now have the same score, zero. Therefore in our results the first sense is chosen which is incorrect. On the other hand, if

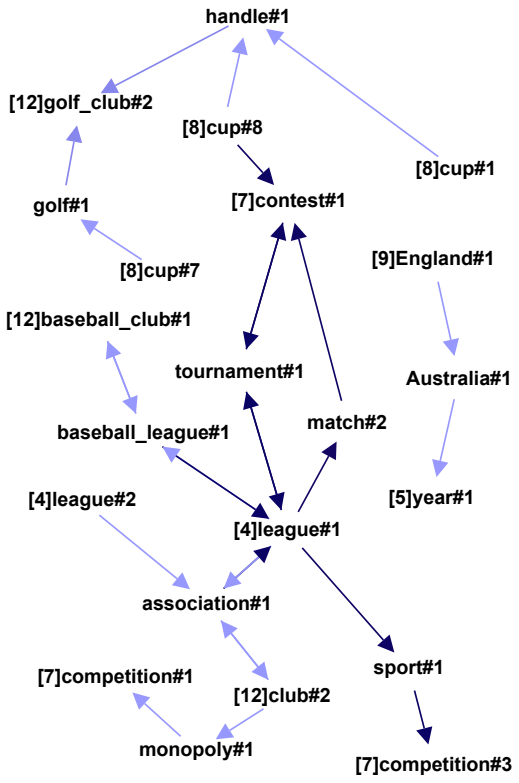
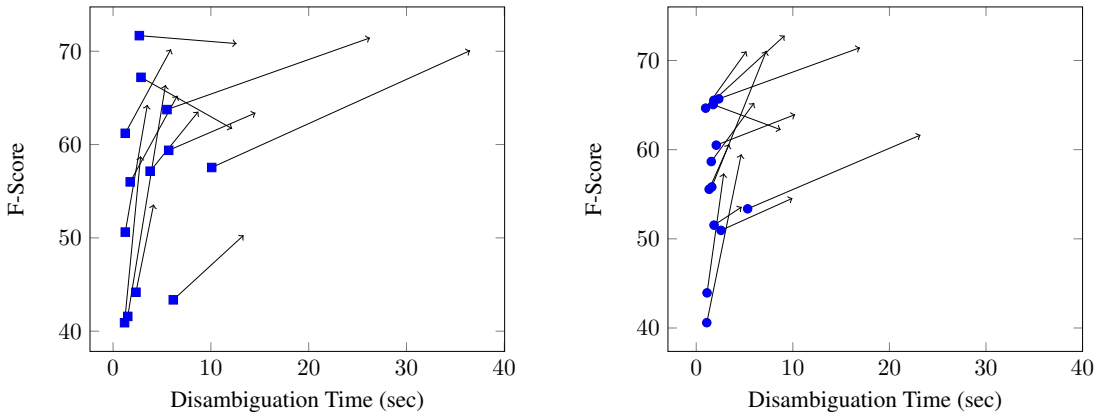


Figure 5: Conventional vs Iterative Subgraph

the subgraph was constructed iteratively with disambiguation results providing feedback to consecutive constructions, this could have been avoided. The shortest paths cup#1→handle#1→golf_club#2 and cup#7→golf#1→golf_club#2 only exist because the sense golf_club#2 (anchored to the more polysemous lemma *club*) is present, if it was not then the `SENSE_SHIFTS` filter would have removed these alternative senses. This demonstrates that if the senses of more polysemous lemmas are introduced into the subgraph too soon, they can interfere rather than help with disambiguation.

Secondly with each disambiguation at lower levels of polysemy, a more stable context is constructed to perform the disambiguation of much more polysemous lemmas later. Therefore in Figure 5 an iteratively constructed subgraph with *cup* already disambiguated, would mean the other two senses of *cup* would no longer be present. This ensures that club#2 (the correct answer) would have a much stronger chance of being selected than golf_club#2, which would have only one incoming edge from handle#1. Note the conventional approach would lend golf_club#2 one extra incoming edge than club#2 has, which could be problematic if ϕ is a centrality measure of In-Degree.



(a) ϕ = Betweenness Centrality

(b) ϕ = PageRank

Figure 6: For each of the 13 documents, performance (F-Score) is plotted against time to disambiguate, for $\mathcal{G}_{\mathcal{L}}$ = Shortest Paths. The squares (PageRank) and circles (Betweenness Centrality) plot the conventional approach. The arrows show the effect caused by applying the iterative approach, with the arrow head marking its F-Score and time to disambiguate.

4.4 Experiment 2: Performance

4.4.1 Experiment 2: Setup

An obvious caveat of the iterative approach is that it requires the construction of several subgraphs as ρ increases, which of course will require extra computation and time which is a penalty for the improved precision and recall. We decided to investigate the extent to which this happens. We selected Betweenness Centrality and PageRank from Experiment 1, in which both use shortest path subgraphs at the document level. This is because a) they acquired good results at the document level and b) with only 13 documents there are less data points on the plots making it easier to read as opposed to the hundreds of sentences.

4.4.2 Experiment 2: Observations

Firstly from Figures 6(a) and (b) we see that there is a substantial improvement in F-Score for almost all documents, except for two for ϕ = Betweenness Centrality and one for ϕ = PageRank. With some exceptions, for most documents the increased amount of time to disambiguate is not unreasonable. For this experiment, applying the iterative approach to Betweenness Centrality resulted in a mean 231% increase in processing time, from 3.54 to 11.73 seconds to acquire a mean F-Score improvement of +8.85. Again for PageRank, a mean increase of 343% in processing time, from 1.95 to 8.64 seconds to acquire a F-Score improvement of +7.16 was observed.

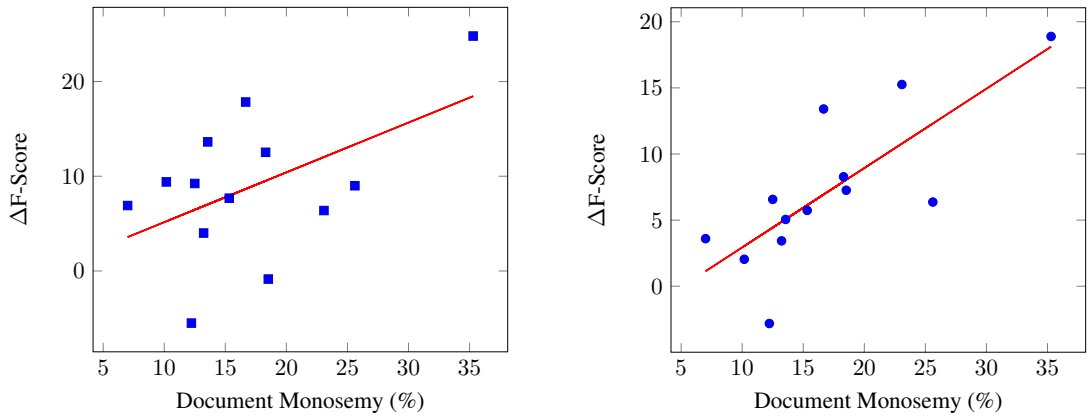
We wanted to investigate why in some cases, the iterative approach can produce poorer results than the conventional approach. We looked at aspects of the subgraphs such as order, size, density, and number of components. Eventually we came to the conclusion that, just like in a Sudoku puzzle, if there are not enough hints to start with, the possibility of finishing the puzzle becomes slim.

Therefore we suspected that if there were not enough monosemous lemmas, to construct the initial $\mathcal{G}_{\mathcal{L}}$, then the effectiveness of the iterative approach could be negated. It turns out, as observed in Figures 7(a) and (b) on the following page that this does effect the outcome. On the horizontal axis, document monosemy represents the percentage of lemmas in a document, not counting duplicates, that are monosemous. The vertical axis on the other hand represents the difference in F-Score between the conventional and iterative approach. Through a simple linear regression of the scatter plot, we observe an increased effectiveness of the iterative approach. This observation is important, because a WSD system may decide on which approach to use based on a document's monosemy.

With m representing document monosemy, and ΔF representing the change in F-Score induced by the iterative approach, the slopes observed in Figures 7(a) and (b) are denoted by Equations (2) and (3) respectively.

$$\Delta F = 0.53m - 0.11 \tag{2}$$

$$\Delta F = 0.60m - 3.07 \tag{3}$$



(a) ϕ = Betweenness Centrality

(b) ϕ = PageRank

Figure 7: Both PageRank (squares) and Betweenness Centrality (circles) are plotted. Each data plot represents the change in F-Score when the iterative approach replaces the conventional approach with respect to the monosemy of the document.

4.5 Experiment 3: A Little Optimisation

Briefly, we made an effort into optimising the iterative approach with subtree subgraphs, and compared these results with systems from SemEval 2013 Task 12 (Navigli et al., 2013) in Table 3.

Team	System	P	R	F
UMCC-DLSI	Run-2$^+$	68.50	68.50	68.50
UMCC-DLSI	Run-3$^+$	68.00	68.00	68.00
UMCC-DLSI	Run-1$^+$	67.70	67.70	67.70
SUDOKU	It-PPR[M]$^+$	67.62	67.51	67.56
MACHINE	**MFS**	66.50	66.50	66.50
SUDOKU	It-PPR[M]	67.20	65.49	66.33
SUDOKU	It-PR[U]	64.07	62.44	63.24
SUDOKU	It-PD	63.58	61.41	62.47
DAEBAK!	PD$^+$	60.47	60.37	60.42
GETALP	BN-1$^+$	58.30	58.30	58.30
SUDOKU	PR[U]	60.09	54.06	56.91
GETALP	BN-2$^+$	56.80	56.80	56.80

Table 3: Comparison to SemEval 2013 Task 12

Firstly, we were able to marginally improve our original result as team DAEBAK! (Manion and Sainudiin, 2013), by applying the iterative approach to our Peripheral Diversity centrality measure (It-PD). Next we tried Personalised PageRank (It-PPR[M]) with a surfing vector biased towards only *M*onosemous senses. We also included regular PageRank (It-/PR[U]) with a *U*niform surfing vector as a reference point. It-PPR[M] almost defeated the MFS baseline of 66.50, but lacked recall. To rectify this, the MFS baseline was used as a back-off strategy (It-PPR[M]$^+$)[4], which then led to us beating the MFS baseline. As for the other teams, GETALP (Schwab et al., 2013) made use of an Ant Colony algorithm, while UMCC-DLSI (Gutiérrez et al., 2013) also made use of PPR, except they based the surfing vector on SemCor (Miller et al., 1993) sense frequencies, set $L = 5$ for shortest paths subgraphs, and disambiguated using resources external to BabelNet. Since their implementation of PPR beats ours, it would be interesting to see how effective the iterative approach could be on their results.

[4]Note that plus$^+$ implies the use of a back-off strategy.

5 Conclusion & Future Work

In this paper we have shown that the iterative approach can substantially improve the results of regular subgraph-based WSD, even to the point of defeating the MFS baseline without doing anything complicated. This is regardless of the subgraph, graph centrality measure, or level of disambiguation. This research can still be extended further, and we encourage other researchers to rethink their own approaches to unsupervised knowledge-based WSD, particularly in regards to the interaction of subgraphs and centrality measures.

Resources

Codebase and resources are at first author's homepage: `http://www.stevemanion.com`.

Acknowledgments

This research was completed with the help of the Korean Foundation Graduate Studies Fellowship: `http://en.kf.or.kr/`

References

Eneko Agirre and Aitor Soroa. 2008. Using the Multilingual Central Repository for Graph-Based Word Sense Disambiguation. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC'08)*, pages 1388–1392, Marrakech, Morocco. ELRA.

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL'09)*, pages 33–41, Athens, Greece. ACL.

Eneko Agirre, Oier Lopez De Lacalle, Christiane Fellbaum, Shu-Kai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers. 2010. SemEval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 75–80, Uppsala, Sweden. ACL.

Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30:107–117.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Linton C. Freeman. 1979. Centrality in Social Networks Conceptual Clarification. *Social Networks*, 1(3):215–239.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, 26:415–439.

Yoan Gutiérrez, Antonio Fernández Orquín, Franc Camara, Yenier Castañeda, Andy González, Andrés Montoyo, Rafael Muñoz, Rainel Estrada, Dennys D. Piug, Jose I. Abreu, and Roger Pérez. 2013. UMCC_DLSI: Reinforcing a Ranking Algorithm with Sense Frequencies and Multidimensional Semantic Resources to solve Multilingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 241–249, Atlanta, Georgia. ACL.

Taher H. Haveliwala. 2003. Topic-Sensitive Pagerank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796.

Adam Kilgarriff. 1998. SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. In *Proceedings of the 1st Language Resources and Evaluation Conference (LREC'98)*, pages 581–585, Granada, Spain.

Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632.

Els Lefever and Veronique Hoste. 2010. SemEval-2010 Task 3: Cross-Lingual Word Sense Disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 15–20, Uppsala, Sweden. ACL.

Els Lefever and Veronique Hoste. 2013. SemEval-2013 Task 10: Cross-lingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 158–166, Atlanta, Georgia. ACL.

Steve L. Manion and Raazesh Sainudiin. 2013. DAEBAK!: Peripheral Diversity for Multilingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 250–254, Atlanta, Georgia. ACL.

Rada Mihalcea. 2005. Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling. In *Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*, pages 411–418, Vancouver, British Columbia. ACL.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A Semantic Concordance. In *Proceedings of the Workshop on Human Language Technology (HLT'93)*, pages 303–308, Princeton, New Jersey. Morgan Kaufmann Publishers.

Roberto Navigli and Mirella Lapata. 2007. Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1683–1688, Hyderabad, India.

Roberto Navigli and Mirella Lapata. 2010. An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.

Roberto Navigli and Simone Paolo Ponzetto. 2012a. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli and Simone Paolo Ponzetto. 2012b. Joining Forces Pays Off: Multilingual Joint Word Sense Disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL'12)*, pages 1399–1410, Jeju Island, South Korea. ACL.

Roberto Navigli and Paola Velardi. 2005. Structural Semantic Interconnections: A Knowledge-based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1086.

Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. 2007. SemEval-2007 Task 07: Coarse-Grained English All-Words Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, Prague, Czech Republic. ACL.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 222–231, Atlanta, Georgia. ACL.

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2):10:1–10:69.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English Tasks: All-Words and Verb Lexical Sample. In *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 21–24, Toulouse, France. ACL.

Ted Pedersen. 2007. Unsupervised Corpus-Based Methods for WSD. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, chapter 6, pages 133–166. Springer, New York.

Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich Word Sense Disambiguation Rivaling Supervised Systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 1522–1531, Uppsala, Sweden. ACL.

Didier Schwab, Andon Tchechmedjiev, Jérôme Goulian, Mohammad Nasiruddin, Gilles Sérasset, and Hervé Blanchon. 2013. GETALP: Propagation of a Lesk Measure through an Ant Colony Algorithm. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval -2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 232–240, Atlanta, Georgia. ACL.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In *Proceedings of the International Conference on Semantic Computing (ICSC'07)*, pages 363–369, Irvine, California. IEEE.

Benjamin Snyder and Martha Palmer. 2004. The English All-Words Task. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*, pages 41–43, Barcelona, Spain. ACL.

David Yarowsky. 1993. One Sense Per Collocation. In *Proceedings of the ARPA Workshop on Human Language Technology (HLT'93)*, pages 266–271, Morristown, New Jersey. ACL.

David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95)*, pages 189–196, Cambridge, Massachusetts. ACL.

REFERENCES

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). *SEM 2013 Shared Task: Semantic Textual Similarity. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 32–43, Atlanta, Georgia. ACL.

Agirre, E. and Edmonds, P. (2007). Introduction. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation Algorithms and Applications*, chapter 1, pages 1–28. Springer, New York.

Agirre, E., Lopez De Lacalle, O., Fellbaum, C., Hsieh, S.-K., Tesconi, M., Monachini, M., Vossen, P., and Segers, R. (2010). SemEval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 75–80, Uppsala, Sweden. ACL.

Agirre, E. and Soroa, A. (2008). Using the Multilingual Central Repository for Graph-Based Word Sense Disambiguation. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC'08)*, pages 1388–1392, Marrakech, Morocco. ELRA.

Agirre, E. and Soroa, A. (2009). Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL'09)*, pages 33–41, Athens, Greece. ACL.

Amsler, R. A. (1984). Lexical Knowledge Bases. In *Proceedings of the 10th International Conference on Computational Linguistics (COLING'84)*, pages 458–459, Stanford, California.

Atkins, S. (1992). Tools for Computer-aided Lexicography: The HECTOR Project. In Kiefer, F., Kiss, G., and Pajsz, J., editors, *Papers in Compu-*

tational Lexicography (Complex'92), pages 1–60, Hungarian Academy of Sciences, Budapest.

Bar-Hillel, Y. (1960). The Present Status of Automatic Translation of Languages. *Advances in Computers*, 1:91–163.

Basile, P., Caputo, A., and Semeraro, G. (2014). An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING'14)*, pages 1591–1600, Dublin, Ireland. ACL.

Biber, D., Conrad, S., and Leech, G. (2002). *Longman Student Grammar of Spoken and Written English*. Pearson/Longman, Harlow, UK.

Brants, T. and Franz, A. (2006). Web 1T 5-gram Version 1. *Linguistic Data Consortium*.

Brin, S. and Page, L. (1998). The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30:107–117.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

Bruce, R. and Wiebe, J. (1994). Word-Sense Disambiguation Using Decomposable Models. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (ACL'94)*, pages 139–146, Las Cruces, New Mexico. ACL.

Chapman, R. (1977). *Roget's International Thesaurus, 4th edition*. Harper & Row, New York.

Chklovski, T. and Mihalcea, R. (2002). Building a Sense Tagged Corpus with Open Mind Word Expert. In *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 116–122, Philadelphia, Pennsylvania. ACL.

Crowther, J., Dignen, S., Lea, D., Deuter, M., Greenan, J., Noble, J., and Phillips, J., editors (2002). *Oxford Collocations Dictionary for Students of English*. Oxford University Press, New York.

Deverson, T. and Kennedy, G., editors (2005). *The New Zealand Oxford Dictionary*. Oxford University Press, New York.

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Freeman, L. C. (1979). Centrality in Social Networks Conceptual Clarification. *Social Networks*, 1(3):215–239.

Gale, W., Church, K. W., and Yarowsky, D. (1992a). Estimating Upper and Lower Bounds on the Performance of Word Sense Disambiguation Programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL'92)*, pages 249–256, Newark, Delaware. ACL.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992b). A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, 26:415–439.

Giles, J. (2005). Internet Encyclopaedias Go Head to Head. *NATURE*, 438(15):900–901.

Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C. M., and Wirth, C. (2012). UBY – A Large-Scale Unified Lexical-Semantic Resource Based on LMF. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL'12)*, pages 580–590, Avignon, France. ACL.

Gutiérrez, Y., Orquín, A. F., Camara, F., Castañeda, Y., González, A., Montoyo, A., Muñoz, R., Estrada, R., Piug, D. D., Abreu, J. I., and Pérez, R.

(2013). UMCC_DLSI: Reinforcing a Ranking Algorithm with Sense Frequencies and Multidimensional Semantic Resources to solve Multilingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 241–249, Atlanta, Georgia. ACL.

Halliday, M. and Hasan, R. (1976). *Cohesion in English*. Longman, London, UK.

Hanks, P. (2000). Do Word Meanings Exist? *Computers and the Humanities*, 34(1):205–215.

Haveliwala, T. H. (2003). Topic-Sensitive Pagerank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796.

Hirst, G. (1987). Introduction. In Joshi, A. K., editor, *Semantic Interpretation and the Resolution of Ambiguity*, chapter 1, pages 5,9. Cambridge University Press, Cambridge, UK.

Hutchins, W. J. (1995). Machine Translation: A Brief History. In Koerner, E. F. K. and Asher, R. E., editors, *Concise History of the Language Sciences: From the Sumerians to the Cognitivists*, pages 431–445. Pergamon Press, Oxford, UK.

Ide, N. and Veronis, J. (1998). Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1):1–40.

Ide, N. and Wilks, Y. (2007). Making Sense About Sense. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation Algorithms and Applications*, chapter 3, pages 47–73. Springer, New York.

Kaplan, A. (1950). An Experimental Study of Ambiguity and Context. *Mechanical Translation*, 2(2):39–46.

Kilgarriff, A. (1997). I Don't Believe in Word Senses. *Computers and the Humanities*, 31(2):91–113.

Kilgarriff, A. (1998). SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. In *Proceedings of the 1st Language Resources and Evaluation Conference (LREC'98)*, pages 581–585, Granada, Spain.

Kilgarriff, A. (2001). SENSEVAL-2: English Lexical Sample Task Description. In *Proceedings of the 2nd Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 17–20, Toulouse, France. ACL.

Kilgarriff, A. (2007). Word Senses. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation Algorithms and Applications*, chapter 2, pages 29–46. Springer, New York.

Kilgarriff, A. and Palmer, M. (2000). Introduction to the Special Issue on SENSEVAL. *Computers and the Humanities*, 34:1–13.

Kilgarriff, A. and Rosenzweig, J. (2000). Framework and Results for English SENSEVAL. *Computers and the Humanities*, 34(1-2):15–48.

Kleinberg, J. M. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632.

Kondrak, G. (2005). N -Gram Similarity and Distance. In *Proceedings of the 12th International Conference on String Processing and Information Retrieval (SPIRE'05)*, pages 115–126, Buenos Aires, Argentina. Springer Berlin Heidelberg.

Kucera, H. and Francis, W. N. (1967). *Computational Analysis of Present-Day American English*. Brown University Press, Providence, Rhode Island.

Leacock, C., Towell, G., and Voorhees, E. (1993). Corpus-based Statistical Sense Resolution. In *Proceedings of the ARPA Workshop on Human Language Technology (HLT'93)*, pages 260–265, Princeton, New Jersey. Morgan Kaufmann Publishers.

Lefever, E. and Hoste, V. (2010). SemEval-2010 Task 3: Cross-Lingual Word Sense Disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 15–20, Uppsala, Sweden. ACL.

Lefever, E. and Hoste, V. (2013). SemEval-2013 Task 10: Cross-lingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 158–166, Atlanta, Georgia. ACL.

Lesk, M. (1986). 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 System Documentation (SIGDOC'86)*, pages 24–26, Toronto, Ontario. ACM.

Levenshtein, V. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics - Doklady*, 10(8):707–710.

Litkowski, K., Hargraves, O., Hall, B., and Road, H. (2007). SemEval-2007 Task 06: Word-Sense Disambiguation of Prepositions. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 24–29, Prague. ACL.

Lizorkin, D., Medelyan, O., and Grineva, M. (2009). Analysis of Community Structure in Wikipedia. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*, pages 1221–1222, Madrid. ACM.

Mallery, J. C. (1988). *Thinking about Foreign Policy: Finding an Appropriate Role for Artificially Intelligent Computers*. MIT Political Science Department, Cambridge, Massachusetts.

Manion, S. L. and Punchihewa, A. (2008a). Fluency Enhancement of Machine Translation. In *Proceedings of the 2nd International Conference on Signal Processing and Communication Systems (ICSPCS'08)*, pages 591–596, Gold Coast. IEEE.

Manion, S. L. and Punchihewa, A. (2008b). Self Learning Live Translation System. In *Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT'08)*, pages 631–639, Busan, South Korea. IEEE.

Manion, S. L. and Sainudiin, R. (2013). DAEBAK!: Peripheral Diversity for Multilingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 250–254, Atlanta, Georgia. ACL.

Manion, S. L. and Sainudiin, R. (2014). An Iterative 'Sudoku Style' Approach to Subgraph-based Word Sense Disambiguation. In *Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (*SEM'14)*, pages 40–50, Dublin, Ireland. ACL.

Manning, C. D. and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts.

Màrquez, L., Escudero, G., Martínez, D., and Rigau, G. (2007). Supervised Corpus-Based Methods for WSD. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation Algorithms and Applications*, chapter 7, pages 167–216. Springer, New York.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). Finding Predominant Word Senses in Untagged Text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, pages 280–287, Barcelona, Spain. ACL.

McGuire, G., Tugemann, B., and Civario, G. (2012). There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem. *School of Mathematical Sciences, University College Dublin, Ireland*.

Medelyan, A., Manion, S. L., Broekstra, J., Divoli, A., Huang, A.-l., and Witten, I. H. (2013). Constructing a Focused Taxonomy from a Document

Collection. In *Proceedings of the 10th Extended Semantic Web Conference (ESWC'13)*, pages 367–381, Montpellier, France. Springer, Heidelberg.

Medelyan, O., Milne, D., Legg, C., and Witten, I. H. (2009). Mining meaning from Wikipedia. *Journal of Human Computer Studies*, 67(9):716–754.

Mendes, P. N., Jakob, M., García-silva, A., and Bizer, C. (2011). DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS'11)*, pages 1–8, Graz, Austria. ACM.

Mihalcea, R. (2005). Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling. In *Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*, pages 411–418, Vancouver, British Columbia. ACL.

Mihalcea, R. (2007). Knowledge-Based Methods for WSD. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation Algorithms and Applications*, chapter 5, pages 107–131. Springer, New York.

Mihalcea, R., Chklovski, T., and Kilgarriff, A. (2004). The SENSEVAL-3 English Lexical Sample Task. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*, pages 25–28, Barcelona, Spain. ACL.

Mihalcea, R., Sinha, R., and McCarthy, D. (2010). SemEval-2010 Task 2: Cross-Lingual Lexical Substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 9–14, Uppsala, Sweden. ACL.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A Semantic Concordance. In *Proceedings of the Workshop on Human Language Technology (HLT'93)*, pages 303–308, Princeton, New Jersey. Morgan Kaufmann Publishers.

Milne, D. and Witten, I. H. (2008). An Effective , Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence*, pages 25–30, Chicago, Illinois. AAAI Press.

Nakayama, K., Hara, T., and Nishio, S. (2007). Wikipedia Mining for an Association Web Thesaurus Construction. In *Proceedings of the 8th International Conference on Web Information Systems Engineering (WISE'07)*, pages 322–334, Nancy, France. Springer, Heidelberg.

Navigli, R. (2009). Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2):10:1–10:69.

Navigli, R., Jurgens, D., and Vannella, D. (2013). SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 222–231, Atlanta, Georgia. ACL.

Navigli, R. and Lapata, M. (2007). Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1683–1688, Hyderabad, India.

Navigli, R. and Lapata, M. (2010). An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.

Navigli, R., Litkowski, K. C., and Hargraves, O. (2007). SemEval-2007 Task 07: Coarse-Grained English All-Words Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, Prague, Czech Republic. ACL.

Navigli, R. and Ponzetto, S. P. (2012a). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250.

Navigli, R. and Ponzetto, S. P. (2012b). Joining Forces Pays Off: Multilingual Joint Word Sense Disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL'12)*, pages 1399–1410, Jeju Island, South Korea. ACL.

Navigli, R. and Ponzetto, S. P. (2012c). Multilingual WSD with Just a Few Lines of Code: the BabelNet API. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, pages 67–72, Jeju, South Korea. ACL.

Navigli, R. and Velardi, P. (2005). Structural Semantic Interconnections: A Knowledge-based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1086.

Nelson, F. (1976). Homographs. *American Speech*, 51(3):296 – 297.

Ng, H. T. and Lee, B. L. (1996). Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics (ACL'96)*, pages 40–47, Santa Cruz, California. Morgan Kaufmann Publishers.

Norvig, P. (2007). Google Developers Day US: Peter Norvig Seminar.

Nunan, D. (1993). Cohesion. In *Introducing Discourse Analysis*, chapter 2, pages 21–33. Penguin English, London, UK.

Olinsky, C. and Black, A. W. (2000). Non-Standard Word and Homograph Resolution for Asian Language Text Analysis. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP / INTERSPEECH'00)*, pages 733–736, Beijing, China. ISCA.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., and Dang, H. T. (2001). English Tasks: All-Words and Verb Lexical Sample. In *Proceedings of the*

2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2), pages 21–24, Toulouse, France. ACL.

Pedersen, T. (2007). Unsupervised Corpus-Based Methods for WSD. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation: Algorithms and Applications*, chapter 6, pages 133–166. Springer, New York.

Ponzetto, S. P. and Navigli, R. (2010). Knowledge-rich Word Sense Disambiguation Rivaling Supervised Systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 1522–1531, Uppsala, Sweden. ACL.

Procter, P. (1978). *Longman Dictionary of Contemporary English*. Longman Group, Harlow, UK.

Punchihewa, A., Manion, S. L., and De Silva, L. (2006). Interactive Translation of Japanese to Korean. In *Proceedings of the 2nd International Conference on Information Automation (ICIA'06)*, pages 313–318, Colombo, Sri Lanka. IEEE.

Ravin, Y. and Leacock, C., editors (2000). *Polysemy*. Oxford University Press, Oxford, UK.

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw Hill Book Company, New York.

Schwab, D., Tchechmedjiev, A., Goulian, J., Nasiruddin, M., Sérasset, G., and Blanchon, H. (2013). GETALP: Propagation of a Lesk Measure through an Ant Colony Algorithm. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval -2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 232–240, Atlanta, Georgia. ACL.

Sérasset, G. (2012). Dbnary: Wiktionary as a LMF based Multilingual RDF network. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, pages 2466–2472, Istanbul, Turkey. ELDA.

Sharoff, S. (2006). Open-source Corpora: Using the Net to Fish for Linguistic Data. *International Journal of Corpus Linguistics (IJCL'06)*, 11(4):435–462.

Sinha, R. and Mihalcea, R. (2007). Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In *Proceedings of the International Conference on Semantic Computing (ICSC'07)*, pages 363–369, Irvine, California. IEEE.

Snyder, B. and Palmer, M. (2004). The English All-Words Task. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*, pages 41–43, Barcelona, Spain. ACL.

Sorensen, T. (1948). A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons. *Videnski Selskab Biologiske Skrifter*, 5(4):1–34.

Stevenson, M. and Wilks, Y. (2001). The Interaction of Knowledge Sources in Word Sense Disambiguation. *Computational Linguistics*, 27(3):321–349.

Templeton, S. (2003). Comprehending Homophones, Homographs, and Homonyms. *Voices from the Middle*, 11(1):62–63.

Turdakov, D. Y. (2010). Word Sense Disambiguation Methods. *Programming and Computer Software*, 36(6):309–326.

Van Rijsbergen, C. (1979). Evaluation. In *Information Retrieval*, chapter 7, pages 112–140. Butterworths, London, 2nd edition.

Voss, J. (2005). Measuring Wikipedia. In *Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics (ISSI'05)*, pages 1–12, Stockholm, Sweden.

Weaver, W. (1949). Translation. In Locke, W. N. and Booth, A. D., editors, *Machine Translation of Languages: Fourteen Essays*, pages 15–23. MIT Press and John Wiley & Sons (later published in 1955), Cambridge, Massachusetts and New York.

Wilks, Y. and Stevenson, M. (1996). The Grammar of Sense: Is Word-Sense Tagging Much More Than Part-of-Speech Tagging? Technical report, University of Sheffield, Sheffield, UK.

Wilks, Y. and Stevenson, M. (1998). The Grammar of Sense: Using Part-of-Speech Tags as a First Step in Semantic Disambiguation. *Natural Language Engineering*, 4(2):135–143.

Wilson, R. J. and Watkins, J. J. (1990). *Graphs: An Introductory Approach*. John Wiley & Sons.

Yarowsky, D. (1993). One Sense Per Collocation. In *Proceedings of the ARPA Workshop on Human Language Technology (HLT'93)*, pages 266–271, Morristown, New Jersey. ACL.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95)*, pages 189–196, Cambridge, Massachusetts. ACL.

Zipf, G. K. (1945). The Meaning-Frequency Relationship of Words. *The Journal of General Psychology*, 33(2):251–256.

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley Press, Inc., Cambridge, Massachusetts.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Final Version as of August, 2014 (classicthesis version 1.0).