Generation and Application of Semantic Networks from Plain Text and Wikipedia



Pia-Ramona Wojtinnek

Somerville College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2012

To my parents

Abstract

Natural Language Processing systems crucially depend on the availability of lexical and conceptual knowledge representations. They need to be able to disambiguate word senses and detect synonyms. In order to draw inferences, they require access to hierarchical relations between concepts (DOG *isAn* ANIMAL) as well as non-hierarchical ones (GASOLINE *fuels* CAR). Knowledge resources such as lexical databases, semantic networks and ontologies explicitly encode such conceptual knowledge. However, traditionally, these have been manually created, which is expensive and time consuming for large resources, and cannot provide adequate coverage in specialised domains.

In order to alleviate this acquisition bottleneck, statistical methods have been created to acquire lexical and conceptual knowledge automatically from text. In particular, unsupervised techniques have the advantage that they can be easily adapted to any domain, given some corpus on the topic. However, due to sparseness issues, they often require very large corpora to achieve high quality results.

The spectrum of resources and statistical methods has a crucial gap in situations when manually created resources do not provide the necessary coverage and only limited corpora are available. This is the case for real-world domain applications such as an NLP system for processing technical information based on a limited amount of company documentation.

We provide a large-scale demonstration that this gap can be filled through the use of automatically generated networks. The corpus is automatically transformed into a network representing the terms or concepts which occur in the text and their relations, based entirely on linguistic tools. The networks structurally lie in between the unstructured corpus and the highly structured manually created resources. We show that they can be useful in situations for which neither existing approach is applicable. In contrast to manually created resources, our networks can be generated quickly and on demand. Conversely, they make it possible to achieve higher quality representations from less text than corpus-based methods, relieving the requirement of very large scale corpora.

We devise scaleable frameworks for building networks from plain text and Wikipedia with varying levels of expressiveness. This work creates concrete networks from the entire British National Corpus covering 1.2m terms and 21m relations and a Wikipedia network covering 2.7m concepts. We develop a network-based semantic space model and evaluate it on the task of measuring semantic relatedness. In addition, noun compound paraphrasing is tackled to demonstrate the quality of the indirect paths in the network for concept relation description. On both evaluations we achieve results competitive to the state of the art. In particular, our network-based methods outperform corpus-based methods, demonstrating the gain created by leveraging the network structure.

Acknowledgements

I would like to thank my supervisor Stephen Pulman for his support throughout my DPhil. I could rely on his vast experience and clear view of the big picture to keep me on track and stir me into the right direction.

Brian Harrington's work was an essential basis for this thesis. I very much enjoyed working with him in this novel area. His advice and calming words in any fictional and real crisis were much appreciated.

Without Johanna Völker, the Wikipedia-based networks would have remained an idea. She encouraged and helped me turn the networks into reality and we could spend hours spinning new ideas for further research.

I also would like to thank Sebastian Rudolph for numerous interesting discussions and his never ending belief in and excitement about my work. He thoroughly reviewed and commented on everything I ever wrote and published, including the thesis.

Without my parents' support and encouragement, none of this work would have been possible. They gave the me strength to manage this endeavour.

And it all would not have been the same without my friends and colleagues Andrei Akhvlediani, Rebekka Ott, Jonathan Kochems, Kerstin Weller, Roman Priebe, Leopold Haller, Rasmus Wissmann, Karl Moritz Hermann and Afifah Waseem. We shared the ups and downs of our exciting student lives, our hopes and despairs and in exceptional situations even some Ben and Jerry's.

This research was funded by the EPSRC and the Oxford University Department of Computer Science.

Contents

1 Introduction **1**

1.1 Lexical and Conceptual Knowledge Representation 2
1.2 Networks from Text 4
1.3 Contribution 5
1.4 Outline 6

I Motivation **9**

2 Conceptual and Lexical Knowledge Modelling **11**

2.1 Lexical and Conceptual Knowledge Resources 11
2.1.1 WordNet and other Lexical Databases 11
2.1.2 Semantic Networks 13
2.1.3 Ontologies 15
2.1.4 Wikipedia 16
2.2 Conceptual Knowledge Acquisition from Text 17
2.2.1 Vector Space Models 18
2.2.2 Patterns and Relations 20
2.3 Summary 22

3 (Semantic) Networks from Text **25**

3.1 Automatically Generated Networks and their Applications 25
3.1.1 Term Networks 26
3.1.2 Concept Networks 29
3.1.3 Overview 30
3.2 Scope of this Work 34
3.2.1 The Problem 34
3.2.2 Aim and Methodology of this Work 35

II Generation **37**

4 ASKNet **39**

4.1 Framework . . . 39
4.1.1 Sentence-level Networks . . . 39
4.1.2 The Update Algorithm . . . 42
4.1.3 Evaluation and Application . . . 42
4.2 Analysis . . . 43
4.2.1 Network Model and Quality . . . 43
4.2.2 Scalability . . . 45

5 Building the Networks 49
5.1 Plain Text: Term Networks . . . 49
5.1.1 Framework . . . 49
5.1.2 Networks from the British National Corpus . . . 50
5.2 Wikipedia Hyperlink Integration . . . 53
5.2.1 Building Framework . . . 53
5.2.2 Networks from Wikipedia's first sentences . . . 56
5.3 Network Structures . . . 60
5.4 Comparison to other Automatically Generated Networks . . . 64
5.5 Summary . . . 65

6 Implementational Aspects and Infrastructure 67
6.1 Memory Usage . . . 67
6.2 Network Model Complexity and Efficient Processing . . . 68
6.2.1 Skeleton Graph . . . 70
6.3 Standardised Outputs and Integration with External Tools . . . 72
6.3.1 Resource Description Framework (RDF) . . . 72
6.3.2 Graph Modelling Language (GML) . . . 74
6.4 Summary . . . 75

III Application 77

7 A Network-based Semantic Space Model for Semantic Relatedness 79
7.1 Semantic Relatedness . . . 80
7.1.1 Approaches . . . 80
7.1.2 Evaluation . . . 82
7.2 Models of Semantic Space . . . 82
7.3 A Network-Based Semantic Space . . . 85
7.3.1 Spreading Activation . . . 85
7.3.2 Network-based Semantic Space Model Parameters . . . 87
7.3.3 Experimental Methodology . . . 90
7.4 1 million sentences BNC Network . . . 91
7.4.1 Raw spreading activation . . . 91
7.4.2 Basis filtering and association function . . . 94

7.4.3 Final results . . . 98
7.5 BNC - Scalability of Results . . . 99
7.6 Wikipedia Networks . . . 99
7.6.1 Concept Mapping . . . 100
7.6.2 Results . . . 102
7.7 Combining BNC and Wikipedia . . . 104
7.8 Comparison with Related Approaches to Semantic Relatedness . . . 105
7.9 Summary . . . 108

8 Network-based Unsupervised Noun Compound Paraphrasing 111
8.1 Noun Compound Paraphrasing . . . 112
8.1.1 Task and Evaluation . . . 113
8.1.2 Approaches . . . 114
8.2 Path-based Noun Compound Paraphrasing . . . 116
8.3 Plain Networks . . . 117
8.3.1 BNC Networks . . . 117
8.3.2 Plain Wikipedia Network . . . 123
8.4 Wikipedia Network . . . 124
8.5 Comparison with Related Unsupervised Systems . . . 126
8.6 Summary . . . 127

9 Conclusion and Future Work 129
9.1 Future Work . . . 130

Appendix 133

Chapter 1

Introduction

Any Natural Language Processing (NLP) application, which aims to draw accurate conclusions from written language, depends on the ability to understand its fundamental building blocks - concepts and their relations. Consider the following news snippet to be processed by an automated NLP-based trading system:

> `Mr Buffett, owner of a popular chain of restaurants, notes that demand in chips has suddenly risen sharply. This has already resulted in considerable delays throughout the supply chain leaving customers unhappy. [...] The two major producers of chips are expected to run unexpectedly high profits in this quarter.`

Without understanding that 'chips' can refer to both tasty deep-fried chunks of potato and essential pieces of computer hardware and the fact that Mr Buffett's area of expertise suggests the snippet is concerned with the former, the system may be inclined to send out an order to buy large amounts of IBM stock. This identification of word senses and their disambiguation in context is often considered the most basic ingredient of dealing with concepts.

Having correctly identified the concepts in the text, a system may then be faced with needing to draw inferences that involve understanding how the concepts relate. Consider the following query posed to a search engine or question answering system and a corresponding text snippet from a website:

> `Q: Where can I dance Salsa in London?`
> `W: Havana runs On2 socials every Friday and Saturday from 8pm til late. Beginners and Intermediate lessons available 9pm-10pm on the upstairs floor. Contact: Bar Havana, 353 Clement Road, SE13 5AE London`

In order to identify the website as a valid match, the system needs to know that 'On2' is a specific style of Salsa, and that the noun compound *On2 social* implies the relation 'On2-*dancedAt*-social'. The first relation is a hierarchical, IS-A relationship, the second one is non-hierarchical.

Many NLP approaches employ lexical and conceptual knowledge resources to retrieve the knowledge needed to draw the illustrated conclusions. However, resources are usually manually cre-

ated and often do not provide the coverage of concepts and relations needed. In order to deal with this drawback, statistical methods have been created for the automatic acquisition of all aspects of conceptual knowledge from text. Those, however, often rely on the availability of very large corpora to produce good results.

This leads to a crucial gap particularly in the case of domain-specific, real-world applications for which resources do not provide enough coverage and the available corpora are limited. This thesis demonstrates that networks, automatically generated from text, can fill this gap. They can be generated quickly and on demand in contrast to resources. At the same time, they make it possible to achieve higher quality representations from less text than corpus-based methods which use the text directly.

1.1 Lexical and Conceptual Knowledge Representation

Lexical and conceptual knowledge resources explicitly encode information such as noun senses, hierarchical and non-hierarchical relations. The most prominent example of such a resource is WordNet [Fellbaum, 1998], which contains senses and synonyms, hierarchical IsA and IsPartOf relations as well as definitions for approximately 120,000 nouns, verbs and adjectives. WordNet was manually built by experts over 10 years and has been widely used for a variety of applications, including Word Sense Disambiguation, Recognising Textual Entailment and Question Answering.

The automatic trading system in our first example could easily employ a WordNet-based sense disambiguation algorithm to detect the correct sense of 'chips' and make a more accurate prediction. The second example, however, reveals the core limitations of manually created resources: coverage and breadth of information. While WordNet does represent hierarchical relationships of the type connecting 'On2' and 'Salsa', it contains neither of the two concepts. This is due to the fact that WordNet is a general coverage resource containing the most common words in English, but does not accommodate specific topic domains such as dance styles, marine biology or physics. If we take the number of articles in Wikipedia - about 3.5 million - as a ballpark figure of the number of concepts or things a large-scale NLP application needs to be able to reason about and compare it to the 120,000 concepts in WordNet it becomes apparent that manual construction by a handful of experts does not scale. A similar problem arises for the relations: WordNet covers hierarchical relationships only and would not help to understand the *dancedAt* relation.

Besides lexical databases such as WordNet, there are other explicit representation formalisms with slightly different foci. Most notably these are semantic networks which connect nodes through a set of relations in a graph structure and ontologies which are logic-based and allow for automated reasoning in the form of theorem provers. The distinction between the representations can be blurry - for example WordNet is often interpreted as a graph in which the

concepts are interconnected by hierarchical relationships. Despite their differences, they all share the inherent problem of the *acquisition bottleneck*.

In order to deal with this acquisition bottleneck, there has been extensive research into acquiring lexical and conceptual knowledge automatically from text, covering all aspects from word senses to non-hierarchical relations and definitions. The most attractive of those are unsupervised or minimally supervised methods, which can learn without any costly human input. Such approaches uncover statistical patterns in the textual data and draw conclusions from these patterns about the meaning of words. A influential idea is that we can "understand the meaning of a word by the company it keeps" [Firth, 1957]. Looking back at our first example, a statistical method could learn that 'chips' has two meanings by looking at sentences which contain the word 'chips' and detect that the other words accompanying it in these sentences tend to fall into two distinct groups - one containing words such as 'eat', 'greasy', 'vinegar' and 'cheese'[1] while the other includes 'computer', 'technology', 'manufacture' and 'iPhone'. In a similar fashion, a statistical approach can derive that 'On2' and 'Salsa' are very similar concepts by finding that they appear in the same contexts such as 'dance ___', '___ music' and '___ ladystyling classes'. Methods of this type often use a vector space representation of contexts and are therefore called semantic space models.

A different common strategy is concerned with learning relations from explicit mentions in a corpus. For example, we can look for explicit statements like 'A is a type of B' or 'As, such as B, C and D' and conclude that 'B IS-A A' [Hearst, 1992]. Automatic systems are concerned with learning reliable patterns for a specific relation, starting off only with a couple of seed concept pairs which exhibit the relation at hand.

The representations generated by statistical methods can often be directly employed by applications, making them not just a means for the automatic acquisition of resources, but an alternative altogether. In the case of word senses, we can keep track of the context clusters we found as a representation of each sense and when faced with a new sentence compare the context words to either cluster. In our first example, the word 'restaurant' is likely to be a prominent part of the 'french fries' cluster, but not of the 'computer chip' cluster.

We can see that statistical methods allow for the automation of conceptual knowledge acquisition and representation. With their dependence on text only, they are flexible and can be applied to any domain. However, they have the drawback of producing imperfect representations that in general cannot match the quality of manually created resources. This issue can often be addressed by simply adding more text since statistical models tend to improve with larger input. However, with semantic space models, we see the phenomenon of diminishing returns - results start off getting better steeply with input size and then flatten, often to the extent that even increasing corpus size by an order of magnitude only has marginal effects. In

[1] We see here that statistical methods can give different representations based on the exact corpus they are derived from. This system has a distinctively British bias.

consequence, very large corpora are often needed to achieve high quality results. The problem of sparseness is even more prominent for relation acquisition methods, which often need to make use of the entire web to gain reliable information

Despite their generally complementary properties, resources and statistical approaches leave a core scenario for which either is problematic - domain specific real-world applications for which resources do not provide the necessary coverage and statistical approaches are limited by the amount of text available. To take an example, we may want to use NLP tools to intelligently query patient data and can only make use of the patient records we have access to. Or a company may want to use NLP tools on their internal technical documentations and thus needs high quality tailored conceptual representations generated from a limited set of company owned texts.

For dealing with this situation, we need to make better use of the underlying text and would ideally achieve as high quality as the manual resources. This is one area where there is great potential for the use of automatically generated networks from text.

1.2 Networks from Text

Some approaches to automatic acquisition of lexical and conceptual knowledge such as word sense induction and measuring semantic similarity and relatedness transform a corpus into a network and then base their technique on this structure The networks that are generated can have very different levels of complexity and carry different amounts of information. In the simplest case, the corpus is represented in a term co-occurrence network which has words as nodes and a link between two words if they occurred close to each other in the text, for instance within the same paragraph [Véronis, 2004, Agirre and Soroa, 2007]. Further refinements of term networks link two words only if they have occurred in specific syntactic relationships [Herdağdelen et al., 2009]. Concept networks add another layer of semantic information by distinguishing different word senses [Szumlanski and Gomez, 2010, Harrington, 2010]. Thus, term networks would have one node for 'chips' while concept networks treat 'chips_(food)' and 'chip_(hardware)' as different nodes.

These networks lie between corpora and resources, in that they are more structured representations of the corpus and in their structure more closely resemble the resources, particularly the more sophisticated the model is. Researchers have taken both views and either argued that they achieved better results by using the network instead of the text directly or that they can solve a task competitively to a resource-based approach.

Automatically generated networks hence have the potential to fill the gap left by resources and statistical methods in domain-specific settings with limited amounts of text. However, despite the encouraging results achieved in previous work, so far their use is only sporadic. We

attribute this fact to both shortcomings of the current approaches and the lack of a unified large-scale understanding of their advantages and limitations. For example, some approaches show convincing results on very small corpora but perform worse than standard corpus-based approaches on larger corpora, severely limiting their applicability. Others use corpora tailored to the task or test set at hand, but do not demonstrate whether results transfer to an untailored corpus, which is the more likely scenario in an real-world application. In addition, due to the fact that each approach uses their own network model and differently sized corpora it is unclear to which extent methods and results are effected by model complexity, corpus size and type.

In order to establish automatically generated networks as a convincing option, we advocate the need to demonstrate that both the construction of the networks and the results that can be achieved scale to realistic corpus sizes such as the widely used 100m words British National Corpus. In addition, their application should be versatile and different types of conceptual knowledge should be derivable without the need to amend the network.

1.3 Contribution

This thesis provides the first large-scale demonstration that automatically generated semantic networks from text present a scalable and versatile basis for lexical and conceptual knowledge acquisition. We transfer established corpus-based statistical methods to the networks and show that we achieve considerable improvements through leveraging the network structure. At the same time, we are able to achieve results that are competitive with WordNet-based methods on the task of measuring semantic relatedness. We conclude that automatically generated networks can contribute to filling the gap between manually created resources and statistical corpus-based methods in the case of domain-specific applications. In contrast to corpus-based methods automatically generated networks can produce higher quality lexical and conceptual representations on less data alleviating the common drawback of the dependence on large corpora. And by being automatically generated, the networks do not incur the problem of the acquisition bottleneck for lexical resources.

In detail, the contribution of this thesis is as follows:

- We devise scalable frameworks for building term networks from plain corpora and concept networks from Wikipedia. Concretely, we build the largest expressive networks automatically acquired from text to date in a matter of a few days. The largest term network covers the entire 100m word British National Corpus in 1.2m terms and 21.1m relations. The Wikipedia network contains 2.7m concepts and 7m relations.
- We develop network-based approaches for measuring semantic relatedness and noun compound paraphrasing. For the former, we construct a network-based semantic space model, for the latter we evaluate relation descriptions from indirect paths. On both evaluations we achieve results competitive to the state of the art and outperform the corpus-based

models. Through our experimental setup we can directly demonstrate the gain achieved from leveraging the network structure.

- We conduct an in-depth analysis of the effect of corpus size and type on the network structure and the performance on the evaluation tasks. The performance on both tasks increases with corpus size and is robust to corpus type. In addition, we contrast term and concept networks to provide an understanding of the exact differences in performance achieved though the additional semantic layer of the latter.

1.4 Outline

This thesis consists of three parts. The first part motivates our work by positioning automatically generated networks within the field of conceptual knowledge representation and acquisition. It finishes by defining the scope and aim of the thesis. The second part describes our network generation frameworks and the concrete networks we built for our investigations. In the third part we devise and evaluate our network-based approaches to Semantic Relatedness and unsupervised Noun Compound Paraphrasing.

PART I: Motivation

Chapter 2 This chapter describes the landscape of conceptual knowledge representation and acquisition. It presents different types of resources including lexical databases, semantic networks and ontologies and outlines statistical corpus-based techniques. We conclude that manual resources have the advantage of high quality but are limited by the acquisition bottleneck. In contrast, statistical methods are automated but often need to rely on very large corpora to achieve good results.

Chapter 3 Scattered over various tasks within conceptual knowledge acquisition, a small number of approaches are based on 'on the fly' generated networks from text. In this chapter, we compile an overview of such approaches and distinguish term and concept networks. We argue that automatically generated networks lie between resources and standard statistical corpus techniques and have the potential to overcome drawbacks of either exactly in those situations when resources do not provide the necessary coverage and web-scale corpora are not available such as in domain-specific applications. We conclude the chapter by defining the aim of the thesis to be the first large-scale demonstration that automatically generated semantic networks present a scalable and versatile basis for conceptual representation and acquisition.

Part II: Generation

Chapter 4 The review of automatically generated networks from text identified that the ASKNet system [Harrington, 2009] leads to the most sophisticated and expressive concept networks. As we base our own frameworks on ASKNet, this chapter describes the system in

more detail and accesses its scalability. We show that the original framework does not scale to large corpora due to its expensive automatic concept disambiguation algorithm.

Chapter 5 In this chapter, we detail our frameworks for generating term networks from plain text and concept networks from Wikipedia articles. The latter makes use of the hyperlinks for concept disambiguation. Concretely, we build term networks of different sizes from the 100m words British National Corpus, the largest of which covers the entire corpus. A concept network of comparable size is generated from the first sentences of all articles in Wikipedia as well as a term network from the plain text version of the same corpus for comparison. The range of networks created allows us to conduct a comprehensive analysis of the effect of corpus size and type and the differences between term and concept networks throughout the thesis. We briefly analyse the structure of the networks and conclude the chapter with putting our networks in perspective to those generated in previous work.

Chapter 6 The generation and application of large-scale expressive semantic networks which consist of tens of millions of nodes and links pose considerable implementational challenges. In this chapter we describe how we addressed the memory usage bottleneck and achieved linear growth, implemented different processing methods and translated the network into standard formalisms including RDF and GML. The translations not only bridge the gap to external tools for querying, visualising and analysing the networks, but also ensures our networks can be accessed easily by other researchers and can be directly integrated with other resources.

Part III: Application

Chapter 7 In this chapter, we construct a network-based semantic space model using spreading activation and evaluate it on the task of measuring semantic relatedness. We conduct an extensive experimental exploration of model settings to establish those that make best use of the network structure. Through direct comparison with an optimised corpus-based semantic space model we demonstrate that our model achieves significantly superior results on all network sizes, confirming our hypothesis that we can make more use of the underlying text. Furthermore, our Wikipedia concept networks prove to achieve competitive results to WordNet thus providing a viable alternative to manual resources. We conclude by showing that a simple combination of the Wikipedia and BNC scores achieves results competitive to the state of the art on the task.

Chapter 8 As a second application, we tackle the task of unsupervised noun compound paraphrase ranking using a simple path-based approach. Again, we investigate the effect of corpus size and type on the quality of the representation and contrast term and concept networks. The results demonstrate that the paths in our automatically generated networks provide a powerful basis for high quality descriptions of relations between concepts as we outperform previous systems and achieve competitive results to web-scale systems.

Chapter 9 This final chapter summarises the findings of the thesis and outlines paths for future work.

Part I

Motivation

Chapter 2

Conceptual and Lexical Knowledge Modelling

In this chapter, we give an overview over lexical and conceptual knowledge representation and acquisition. We cover word senses, synonyms, hierarchical and non-hierarchical relations and contrast traditionally manually created resources with corpus-based statistical methods. The description sets out the landscape and motivates the use of automatically generated networks.

Section 2.1 presents different types of lexical and conceptual knowledge resources including lexical knowledge databases, semantic networks and ontologies. Section 2.2 describes corpus-based lexical knowledge acquisition methods. We focus on semantic space models which can be used to induce word senses, to detect synonyms, form semantic categories and to measure semantic relatedness. In addition, we give an overview of methods for learning non-hierarchical relationships.

2.1 Lexical and Conceptual Knowledge Resources

Most available conceptual knowledge resources are based on one of three core types of structures: *lexical databases*, *semantic networks* and *ontologies*. All three are tightly related and the names sometimes used interchangeably in publications.

2.1.1 WordNet and other Lexical Databases

Lexical databases deal in particular with the concept-lexeme gap. They map words to concepts, thereby dealing with both synonymity and variety of senses. In addition, concepts are ordered hierarchically, primarily on the basis of hyponymy.

WordNet [Fellbaum, 1998] is the most widely used lexical knowledge base available in this category. Its basic unit is the *synset*, which is a group of synonymous words (nouns, verbs,

adjectives or adverbs). Synsets represent concepts and thus for a word, the synsets it belongs to represent its different senses. Each synset comes with a gloss defining the concept. In the latest version, the words in these glosses are themselves disambiguated and the contained content words are linked to the appropriate synset. Synsets are linked by semantic relationships. For nouns, the considered relationships are hypernymy/hyponymy and holonymy/meronymy, giving a taxonomy of concepts. Figure 2.1 shows the entries for "thesis". The term can be found in two synsets, one referring to the "hypothesis" sense, the other one referring to the "dissertation" sense of the word. The inherited hypernym illustrates the WordNet taxonomy of concepts, classifying a dissertation to be a sub concept of a general "communication".

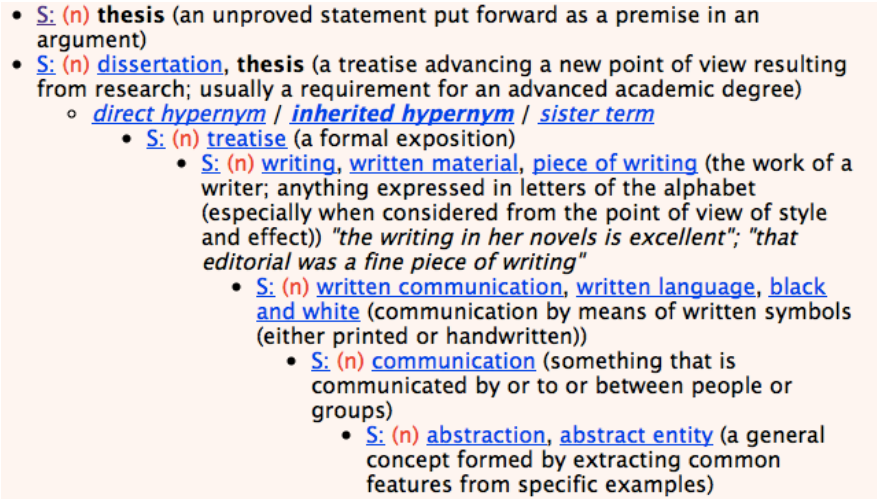


Figure 2.1: WordNet 3.1 entries for "thesis". Screenshot of results on 21st March 2012.

WordNet was built by experts over 10 years and currently covers around 120,000 synsets (80,000 from nouns)[1]. Both the taxonomic structure and the glosses of WordNet have been used for Automatic Word Sense Disambiguation [Navigli and Velardi, 2005, Tsatsaronis et al., 2007] and for measuring Semantic Relatedness [Hughes and Ramage, 2007, Agirre et al., 2009]. It has also been used as part of larger systems for Open-Domain Question Answering [Harabagiu et al., 2001], Recognising Textual Entailment [Tatu and Moldovan, 2005] and Automated Text Summarisation [Chaves, 2001].

The advantages and disadvantages of using WordNet depend on the task at hand. In general, the quality of the encoded information is very high due to manual creation by experts. It provides an important resource for word senses and taxonomic relations and the new addition of disambiguated glosses opens the possibility of wider applications. However, manual construction means that the representation is rigid and despite a good coverage of the most frequent words in English, WordNet based approaches cannot directly be applied to specific domains due to lack of domain coverage. As a comparison, the online encyclopedia Wikipedia currently

[1] As of March 2012, WordNet version 3.0 `http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html`

contains more than 3.5 million entries, giving an idea of the dimension needed for wide coverage. Furthermore, while taxonomic relationships are undoubtably the backbone of conceptual knowledge representations, a far wider range of relations is needed to capture all the ways in which concepts relate to each other.

There are other lexical databases available, with different foci. Jarmasz and Szpankowicz [2001] built an electronic, machine tractable version of *Roget's Thesaurus*. It contains a similar number of concepts, but is structured following slightly different principles. The Brandeis Semantic Ontology (BSO) [Pustejovsky et al., 2006] is a lexical database based on the theory of the Generative Lexicon. Concepts are defined by their qualia roles, including the CONSTITUTIVE (its constitutive parts) and the TELIC (its purpose or function). The BSO hence contains a wider range of explicit semantic relations, but it covered only 3700 concepts in 2007 [Havasi et al., 2007] and its development appears to be on hold. VerbNet[2] [Kipper Schuler, 2005] follows similar principles as WordNet, but constitutes a hierarchy of verb senses. Each verb sense is specified through both a syntactic frame which covers semantic roles and selectional restrictions and a semantic, logic-based description of its meaning. VerbNet currently covers 3,769 verbs in 5,257 senses.

2.1.2 Semantic Networks

The second core structure for conceptual knowledge representation is the concept of a *semantic network*. The term 'semantic network' is not well defined and has been used for a variety of network structures within the Natural Language Processing and Knowledge Representation literature. However, on a most general level, we could say semantic networks represent concepts and their relations in an interconnected way to reflect their meaning. In contrast to taxonomies, concepts are not arranged in a tree or hierarchy based on 'is-a' relationships, but in a network that takes into account all important semantic relationships between the concepts.

Semantic networks have their roots in early works on knowledge representations such as Quillian [1967]. In their core, his semantic networks were supposed to model memory structures in the human brain. It would be processed by spreading activation, imitating the firing of neurones in the brain and the transfer of signals through synapses. In his structure, disambiguated words (such as *cry*, *comfort*, *make*, *sad* in Figure 2.2) are represented as nodes, linked by subtype, modification, conjunction, disjunction, subject or object links. The information is based on definitions. Spreading activation would be used to, for example, find that the shortest connection between *cry* and *comfort* is via *sad*: (1) Cry2 is among other things to make a sad sound. (2) To comfort3 can be to make2 something less sad.

Again, the acquisition bottleneck is a major factor in the realisation of semantic networks. Quillian's networks presuppose a complete sense inventory of all occurring words (nouns, verbs, adjectives and adverbs alike) as well as well annotated definitions of all concepts. Actual real-

[2] `http://verbs.colorado.edu/~mpalmer/projects/verbnet.html`

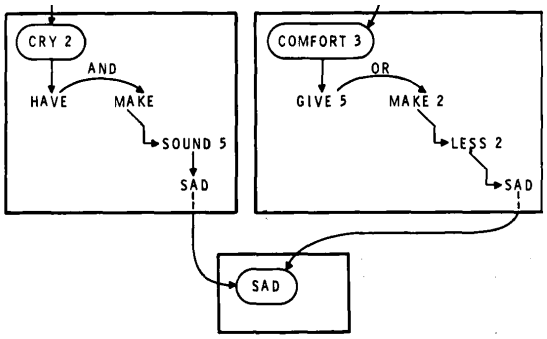


Figure 2.2: Excerpt of a semantic network as envisioned by Quillian [Quillian, 1967]

isations of semantic networks hence tend to reduce their scope.

ConceptNet [Havasi et al., 2007] is the largest available common-sense semantic network and it was generated automatically from a purpose built, crowd-sourced corpus. 14,000 web users were asked to fill in template sentences such as 'THE EFFECT OF EATING FOOD IS ...' or 'A KNIFE IS USED FOR ... '. From these sentences, the semantic network was built using Natural Language Processing tools to extract concepts and semantic relations, facilitated by the template structure. Figure 2.3 shows a small snippet of the network. ConceptNet contains over 150,000 nodes, which can originate from verbs and verb phrases ('drive', 'eat breakfast'), noun phrases ('red car'), prepositional phrases ('at home') or adjectival phrases ('sour'). The links are drawn from a fixed set of 21 relations containing eg. *ConceptuallyRelatedTo* ('bad breath', 'mint'), *PartOf* ('butterfly', 'wing',) *capableOf* ('dentist', 'pull tooth') or *EffectOf* ('eat breakfast', 'full stomach'). Through the use of templates, the restriction to a small set of relations and by taking advantage of collaborative effort, building the resource becomes manageable, but still not viable as an 'on demand' approach. ConceptNet has been used in common sense reliant systems for automatic joke recognition [Tinholt and Nijholt, 2007] and web data annotation [Faaborg and Lieberman, 2006].

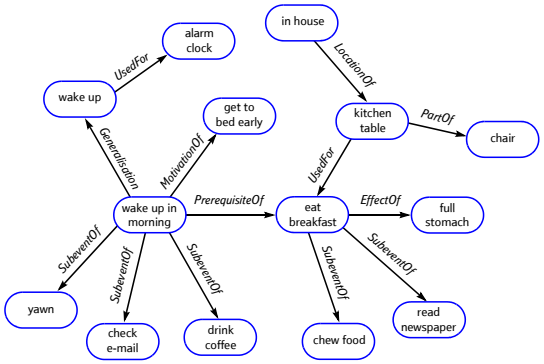


Figure 2.3: Illustration of ConceptNet. Taken from [Liu and Singh, 2004a]

The distinction between semantic networks and WordNet is blurry. WordNet can be viewed from a network perspective and allows for network or graph algorithms such as spreading activation and PageRank to be applied. Sussna [1993] uses the synsets as nodes and the semantic relations from the hierarchical structure as links. Hughes and Ramage [2007] uses both synsets and lexical words as nodes, which are linked by semantic relations or term-to-synset mappings, respectively. Taking into account the recent addition of disambiguated glosses, Navigli and Velardi [2005] introduce an extended model through unnamed links between a synset and the synsets that occur in its disambiguated gloss. This latter framework most closely resembles the networks originally proposed by Quillian. In a similar fashion, researchers have translated machine-readable dictionaries into networks, primarily by connecting word senses through the words they share in their definition [Veronis and Ide, 1990, Kozima and Furugori, 1993].

2.1.3 Ontologies

The third type of conceptual knowledge representation structures originates in the desire for a semantically more strict representation that allows for automatic reasoning. This led to the development of two lines of research. The first looked at the construction of logically precise graphs, which can be directly translated into well-known logic formalisms such as first order logic. Logical reasoning operations are then translated into graph operations. The most well known approach in this category are Conceptual Graphs [Sowa, 2008].

The second line of research looked into the establishment of a non-graph based logic that is both expressive enough to describe concepts and their relationships and tractable for fast reasoning. While there is a long history of proposals for such knowledge representation languages, the most predominant one is based on Description Logics and is called Web Ontology Language (OWL)[3]. It is the accepted standard language for the Semantic Web. The corresponding knowledge base is called an *ontology* and contains a terminological part that defines the concepts and relations and an assertional part that allows for statements on individuals. The framework for the description of concepts contains

- a set of concepts (*dog*, *cat*, *animal*)
- taxonomic relationships between these concepts (*isA(dog, animal)*)
- a set of non-taxonomic relationships (*hasOwner(x,y)*)
- axioms for further specification, i.e. rules ($dog(x) \Rightarrow animal(x) \ \wedge \exists y \ hasOwner(x, y)$)

There is a plethora of OWL ontologies for various domains. They differ in size and in expressiveness. Some contain only concepts and taxonomic relationships. The biomedical ontology GALEN[4] is one of the largest OWL ontologies and covers about 25,000 concepts. While OWL is the most common ontology language, the largest logic-based common sense knowledge base, CYC [Lenat, 1995], is based on its own language CYCL. Started by Douglas Lenat in 1984, it

[3] `http://www.w3.org/TR/owl-features/`
[4] `http://www.co-ode.org/galen/`

has been manually created using handcrafted rules and currently covers about 500,000 concepts with 5 million assertions[5].

2.1.4 Wikipedia

All resources above were primarily manually crafted and aimed towards creating a computationally processable representation of conceptual knowledge. However, recently the Web encyclopedia Wikipedia, which is a resource for human readers, has been shown to be processable in a similar way due to its hyperlinked structure. It is a fast growing resource which currently contains more than 3.5 million articles in the English version and more than 500,000 registered editors. With the advantage of being an online resource, editors structure content for good navigation using hyperlinks, which 'on the fly' produce structures similar to the ones found in the manually created resources.

In Wikipedia, an article can be seen to represent a distinct concept. Disambiguation pages list the different concepts a word can refer to, resembling a WordNet list of synsets or senses of a word. In addition, there is a taxonomy-like category structure. Articles can be linked to one or more of these categories, leading to an implicit hierarchy of concepts. Infoboxes contain standardised information, often in terms of non-taxonomic relationships between concepts. For example, infoboxes in articles for countries usually contain a link to their capital city and their currency. Furthermore, links on article pages to other articles provide a weak semantic network structure in which the links are unnamed. For example, the first sentence of the article about London is "[[London]] is the [[capital]] of [[England]]". As 'capital' and 'England' are hyperlinked in the article, we can attest that there is some relation between them and 'London', but we do not, computationally, know which.

Recent research has looked into adapting approaches used on other semantic resources directly on these resembling Wikipedia structures. For example, measures of semantic relatedness that use the WordNet taxonomy have been adopted to the Wikipedia category structure [Strube and Ponzetto, 2006]. The general conclusion is that, while the quality of the information is less precise than of that in carefully expert-crafted resources, it is the unprecedented coverage that makes Wikipedia a computationally important conceptual knowledge resource.

Apart from using the Wikipedia structures directly, its pre-structured information is a good base for creating rigorous knowledge representations like the ones described above. Most notably, the YAGO ontology[6] is an OWL ontology that incorporates concepts and relations derived from both the Wikipedia category tree and infoboxes [Suchanek et al., 2008].

In conclusion, there is a variety of conceptual knowledge representations and related resources, with different foci, strengths and weaknesses. They follow different rationales as to how exactly

[5] `http://research.cyc.com/`
[6] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

concepts and their relations are presented as well as how to process the resource. WordNet sets an emphasis on the relation between lexemes and concepts and taxonomic relationships. It is hence often used for Word Sense Disambiguation and for generalising by exploiting the taxonomy. ConceptNet is a network that contains a larger number of relations, while being somewhat less rigorous than WordNet. Ontologies describe concepts and relations based on logic and allow for automatic logic reasoning that is particularly useful for agents processing a Semantic Web. However, research into natural language common sense reasoning suggests that human reasoning needs to be fuzzy. For example, Liu and Singh [2004b] note that humans are happy to believe that "birds can fly" despite knowing that "penguins are birds that cannot fly". This sort of reasoning is more easily accommodated for in a semantic network structure like ConceptNet than in an ontology.

Despite differences in the resources, very recent work has seen major efforts in combining and integrating the most used resources. On the one hand, this includes mapping of concepts such as ConceptNet concepts to WordNet synsets and Wikipedia articles. On the other hand, it covers translations from one formalism to the other, for example the translation of WordNet into OWL as part of the YAGO ontology.

However, a drawback that all resources share is the acquisition bottleneck. They either rely on the manual crafting by a few experts, which means construction can take decades to reach a reasonable size, or they are based on the collaborative effort of up to hundreds of thousands of users, which leads to less strict and far noisier structures. In addition, in specific domains such as medicine, the application of collaborative efforts is severely limited. The next section will describe methods of automatic conceptual knowledge acquisition that attempt to close this gap.

2.2 Conceptual Knowledge Acquisition from Text

The bottleneck of manual acquisition of lexical and conceptual knowledge has led to the development of automatic acquisition methods for all its layers: the term-concept gap created by polysemy and synonymy, hierarchical and non-hierarchical relationships between concepts and concept definitions. Work in this area is often summarised under the umbrella of *Ontology Learning* [Biemann, 2005, CIMIANO et al., 2006], which emphasises the goal of such approaches to automatically build a resource. However, the representations generated by statistical methods can often directly be used in an application.

On any level, approaches can differ in the type and extent of data they require. At the one end of the scale are supervised approaches which need hand-annotated or labelled data and methods that focus on extending an existing resource. At the other end of the scale are unsupervised methods that require only plain text as input. While achieving good results, supervised approaches suffer from similar limitations as the manual creation of resources -

labelling (large) amounts of data is expensive. This section reviews unsupervised and minimally supervised methods, which are based on plain text. In particular, we describe two influential principles that underlie a variety of approaches. The first one is the measurement of different types of distributional similarity based on the Distributional Hypothesis. They come into play whenever the *quantification* of similarity or relatedness of terms can be exploited. The other strategy is to use lexico-syntactic patterns to extract both hierarchical and non-hierarchical relations.

2.2.1 Vector Space Models

The proclamation that "You shall know a word by the company it keeps."[Firth, 1957] and the related Distributional Hypothesis that words with similar distributional properties have similar meanings lie at the heart of corpus-based vector space representations of meaning. Concrete semantic space models then differ in the distributional properties they consider. The most widely applied group of models expresses a word by a vector of words it co-occurs with within a certain unit of context. This context unit can be simply a certain word window of a fixed width around the target word [Lund and Burgess, 1996, Bullinaria and Levy, 2007, Agirre et al., 2009] or it can be motivated by syntactic analysis, taking into account the syntactic relationship between the words [Lin, 1998, Padó and Lapata, 2007]. Table 2.1 illustrates the *word-word* matrix a window-based model with width 2 would retrieve on from the following two sentences

S1: The busy student bought new headphones.
S2: The price of headphones has increased.

		Word						
		busy	student	buy	new	headphones	price	increase
Word	busy	0	1	1	0	0	0	0
	student	1	0	1	1	0	0	0
	buy	1	1	0	1	1	0	0
	new	0	1	1	0	1	0	0
	headphones	0	0	1	1	0	1	1
	price	0	0	0	0	1	0	0
	increase	0	0	0	0	1	0	0

Table 2.1: Word-word matrix for a symmetric context window of 2.

The entries of the matrix can be based on co-occurrence frequency, but usually involve more sophisticated weighting. The semantic similarity of two words can be measured by the similarity of their corresponding rows in the matrix. The plethora of work which uses this type of vector space model has produced a variety of options for context choice, weighting, term filtering and similarity measurements the optimal setting of which often depends on the application at hand. We will review these 'dimensions of semantic space' in Chapter 7.

Co-occurrence based word-word matrices are not the only type of vector space representation that has been applied to conceptual knowledge acquisition. Originating in applications for Information Retrieval, word-document matrices have been used, representing a word by the set of documents, paragraphs or text snippets it has occurred in [Deerwester et al., 1990, Landauer and Dumais, 1997]. The resulting matrix is shown in Table 2.2(a). In this representation, words are similar if they appear in the same documents. This gives subtle differences in the semantics of the similarity measured. Word-document based approaches are often combined with further matrix operations such as singular value decomposition which lead to incorporation of the co-occurrences found in word-word matrices [Landauer and Dumais, 1997, Sahlgren, 2006]. A third type of context moves away from the concept of co-occurrence in favour of extracting concept attributes or features from text using templates such as *the * of ** [Almuhareb and Poesio, 2004] (cf. Table 2.2(b)). While this is the semantically most explicit representation, it comes at the cost of greater data sparseness.

		Document	
		S1	S2
Word	busy	1	0
	student	1	0
	buy	1	0
	new	1	0
	headphones	1	1
	price	0	1
	increase	0	1

		Attribute		
		busy	new	price
Concept	student	1	0	0
	headphones	0	1	1

Table 2.2: (a) Word-document matrix as used in Deerwester et al. [1990] and Landauer and Dumais [1997]. (b) Concept-attribute matrix based on Almuhareb and Poesio [2004].

Vector space models have been used for a variety of conceptual knowledge acquisition tasks - both with the goal of building a lexical resource and as a direct input into an NLP application. On the lowest level, they can be used to overcome the gap between lexemes and concepts caused by polysemy and synonymy. Schütze [1992] uses vector-based context clustering for word sense induction and disambiguation and Padó and Lapata [2007] devise a less powerful word sense ranking algorithm. Synonyms have been identified using context window-based vectors [Bullinaria and Levy, 2007], syntax-based vectors [Padó and Lapata, 2007] and document vectors [Landauer and Dumais, 1997]. Together, sense induction and synonymy detection form the first step towards building a lexical or conceptual knowledge resource. Synonyms, however, are only a special case of semantically similar words. Semantic space models have successfully been applied to semantic clustering or categorisation, ranging again from syntactic vectors [Hindle, 1990], context-window vectors [Bullinaria and Levy, 2007] and attribute vectors [Almuhareb and Poesio, 2004]. Theoretically, semantic clusters can be used to build a taxonomy based on the assumption that each cluster represents a more general concept. For example, the cluster {*car, truck, bus*} could be the general concept "motorised vehicle". The difficulty then lies in automatically labelling the clusters and achieving different levels of granularity for corresponding depths in the taxonomy. The latter can be addressed by applying hierarchical clustering

algorithms, as demonstrated in Cimiano et al. [2004] with syntax-based vectors and Cimiano and Staab [2005] with attribute vectors.

In a similar fashion, vector space models are used to measure semantic relatedness [Agirre et al., 2009, Mitchell and Lapata, 2008] and semantic priming [Padó and Lapata, 2007]. Semantic relatedness judges the degree of relatedness not only based on hierarchical relationships (e.g. *car* and *motorbike* are similar because they are both vehicles), but on non-hierarchical relationships (e.g. *car* and *gasoline* are related because the latter fuels the former). Semantic priming is concerned with associations human subjects produce when presented with a target word.

Which type and setting of semantic space model achieves the best result appears to depend on both the task at hand and the notion of context used. Common practice in recent papers evaluates a variety of options and picks the best model [Agirre et al., 2009, Padó and Lapata, 2007]. While the fact that it is not well understood how exactly models and applications relate has motivated comparative experimental [Bullinaria and Levy, 2007, Sahlgren, 2006] and theoretical work [Lowe, 2001], this is still an active research area.

We have seen that semantic space models have been successfully applied to a variety of lexical and conceptual knowledge acquisition tasks, ranging from word sense induction to measuring semantic relatedness, making them a core corpus-based technique. The performance of vector space models, however, crucially depends on the size of the corpus they are based on. Curran [2004] evaluated five models with different context types for measuring semantic similarity and showed that the results were log-linear in the number of words. In other words, while performance generally increases with corpus size, there are diminishing returns of additional data. Curran showed that results can already flatten considerably after 20m words. This effect means that vector space models often need to rely on very large corpora to achieve good results. Agirre et al. [2009] encounter the same effect on the task of measuring semantic relatedness and needed to scale up to a 1.6 trillion word corpus to reach results competitive to WordNet.

2.2.2 Patterns and Relations

In this section, we give an overview of techniques for the automatic acquisition of relations. While we encountered quantitative measures of relatedness in the previous section, the objective here is to acquire explicit qualitative descriptions.

Relations between nominals have traditionally been modelled by an inventory of relations such as PART-WHOLE (e.g. *finger-hand*) and CAUSE-EFFECT (eg. *Amor-love*). Ideally, the set of relations should be both exhaustive and non-overlapping so that each pair of nominals is described by exactly one relation. This approach paves the way for a direct incorporation of the learned relations into a semantic network or lexical database. However, it comes with the difficulty of finding a good relation inventory. Hendrickx et al. [2009] review a variety of inventories and

settle for a set of nine relations in their setup of the SemEval Task of "Classification of Semantic Relations Between Pairs of Nominals". Besides PART-WHOLE and CAUSE-EFFECT, these include for instance PRODUCT-PRODUCER (e.g. *apples-farmer*) and Communication-Topic (e.g. *article-morphology*).

The most common technique for learning specific relations from text is based on Hearst [1992], who acquires hierarchical hypernym-hyponym relations through the use of a handful of patterns. For instance, from an occurrence of the pattern $\texttt{NP}_1$`, `$\texttt{NP}_2$`*, and other NP ...` it can be derived that $\texttt{NP}_1$ and $\texttt{NP}_2$ are hyponyms of `NP`. Hence, from the text snippet *temples, treasuries, and other important civic buildings* she extracts two hierarchical relations *hyponym("temple", "civic building")* and *hyponym("temple", "civic building")*. The method can be used to learn any specific relation and she outlines a bootstrapping algorithm for doing so. It starts off with a couple of seed pairs known to exhibit the desired relation, followed by the retrieval of occurrences of the two words in proximity within the corpus and the discovery of commonalities of their environment as new patterns. These in turn can be used to extract word pairs and so on. Berland and Charniak [1999] use this strategy to extract *part-of* relations and Cimiano and Wenderoth [2005] acquire qualia roles for the definition of concepts.

There are several challenges associated with the extraction of relations in this way, in particular when it comes to the full automation of the bootstrapping algorithm. For concepts, it is simple to map the concept to a syntactic form - they are in general represented by nouns or named entities. However, a single relation can be expressed in a large number of ways: *Tom's shoe*, *the shoe of Tom*, *Tom owns a shoe* and *the shoe that Tom owns* all describe the relation *possess("Tom", "shoe")*. At the same time, generic patterns such as $\texttt{N}_1$ `of` $\texttt{N}_2$ can describe various relations. This complex relationship between relations and surface forms means that the automatic 'discovery of commonalities of the environment' needs to have an accurate way of determining which syntactical or surface structure constitutes a valid pattern. Furthermore, the predictive capacity of a pattern needs to be taken into account so that the bootstrapping algorithm does not spin off into noise. The situation is additionally complicated by the fact that explicit occurrences of non-generic patterns are extremely rare, leading to considerable sparseness issues. Pantel and Pennacchiotti [2006] address these issues in their ESPRESSO system. They use a minimum edit distance algorithm to decide what constitutes a candidate pattern and calculate reliability scores for patterns according to the accuracy of their predictions. In addition, they employ Google and the web as a corpus to counter sparseness issues for both instance retrieval and reliability scoring of patterns.

The view so far assumes the relation between two nominals is constant and that a pair is always described by exactly one relation. However, when we need to determine the relation of two nominals in a sentence, the relation turns out to be dependent on the context. Hendrickx et al. [2009] give the example of the noun compound *wood shed*, which can denote a shed made of wood but also a shed of any material used to store wood. At the same time, more than one

relation can hold simultaneously or 'overlap' [Rozovskaya and Girju, 2009]. The best performing system on this task trains a support vector machine using a plethora of features, which require the use of a variety of sources, including the context words, WordNet hypernyms, the syntactic parse, pre-extracted verb-based relations from a web-scale database and relatedness scores calculated from the Google-n-gram corpus.

These issues with a preset, fixed set of relations have motivated research into softer representations and bottom-up relation generation. Davidov and Rappoport [2008] generate a data-driven set of relations through clustering of patterns. Nakov [2008a] focusses on the subtask of describing the relation between the two parts of a noun compound and rejects the idea of a fixed set of relations altogether. He advocates that a weighted set of possible verb-centred *paraphrases* is a better and more usable description than a specific relation from a fixed set. Abstract relations underspecify the semantics and cannot directly be immediately used in an application. For example, in the case of *malaria mosquito*, the fine-grained list of weighted paraphrases *carry(23), spread(16), cause(12), transmit(9), etc.* gives a more appropriate picture of the relation than the according relation CAUSE from the noun compound relation inventory.

Other researchers have made similar observations and subsequently, a noun compound paraphrasing task was introduced at SemEval2010 [Butnariu et al., 2009]. The best performing system [Li and Lopez-Fernandez, 2010] is a hybrid of a supervised and an unsupervised approach. The latter uses templates to extract <*head, paraphrase, modifier*> triples from the Google n-gram corpus - a web-scale resource based on more than 1 trillion words. Despite its size, sparseness was a major issue meaning that even with wild-carded templates for the head or modifier, the extracted triples needed to be extended with the help of WordNet.

In conclusion, we saw that techniques for learning explicit relations heavily rely on the availability of large-scale corpora, even more so than semantic space models. All approaches we discussed make use of the web or web-scale corpora to counter sparseness. An example of a realisations of web-based, large-scale knowledge extraction is the 'Never-Ending Language Learning' (NELL) system[7] [Carlson et al., 2010]. NELL has been running since 2010 and continuously harvests the web for assertions such as '*wanstead_park* is a *place of worship*'. If the confidence in an assertion is high enough, it is included in the NELL knowledge base.

2.3 Summary

This chapter gave an overview of the spectrum of methods for conceptual knowledge acquisition and representation. On the on end, we find explicit representations in the form of resources including lexical databases, semantic networks and ontologies. While such representations generally provide very high quality of information, the suffer from the acquisition bottleneck as they traditionally rely on expensive manual construction. On the other end of the spectrum,

[7] `http://rtw.ml.cmu.edu/rtw/`

statistical corpus-based methods allow for fast and automated acquisition while often requiring very large amounts of text. This leaves a gap in the case of domain applications, in which manual acquisition is too time and labour intensive and the amount of data available is limited.

Chapter 3

(Semantic) Networks from Text

In this chapter, we review work which automatically generates a network from a corpus, representing the terms or concepts that occur in the text and their relations. The network models differ greatly in what constitutes a node and an edge. The simplest model is a co-occurrence graph in which nodes correspond to words and edges link two words that have co-occurred within some context in the text. The most sophisticated model disambiguates nouns and Named Entities to form a network of concepts (in contrast to lexemes), linked by explicit relations derived from linguistic processing. On the one hand, generated networks differ from WordNet-based semantic networks such as Navigli and Velardi [2005] in that they are automatically generated from unstructured corpora. On the other hand, they differ from large-scale knowledge acquisition systems such as NELL in that they do not extract and select specific conceptual knowledge from a corpus or the web to create a knowledge representation, but rather form a direct network representation of the underlying text.

Section 3.1 reviews previous work on using automatically generated networks from text. Section 3.2 we define the aim and scope of this work.

3.1 Automatically Generated Networks and their Applications

This section looks at work that is based on building graphs or networks generated automatically and directly from a corpus. We group the structures into word or term networks and concept networks. *Term networks* have words or lexemes as their nodes, i.e. they do not distinguish between different senses of the word nor do they account for synonymity. *Concept networks*, in contrast, have concepts as their nodes. This added level of semantics makes them more closely related to the semantic networks we encountered in the previous chapter.

3.1.1 Term Networks

Term networks can be further divided into those that link the nodes based on co-occurrence of the terms within a unit of context using no or only minimal linguistic processing and those that involve full syntactical parsing.

3.1.1.1 Shallow Co-occurrence

A central motivation for building shallow co-occurrence graphs comes from the task of unsupervised word sense induction and disambiguation. It is based on the observation that the neighbourhood of a polysemous or ambiguous word in a co-occurrence graph shows subclusters corresponding to the different senses. For example, the noun *bank* will co-occur with and hence be connected to *river*, *boat*, *water*, *money*, *account* and *equity*. In turn, *river*, *boat* and *water* will be strongly interlinked with each other and so will *money*, *account* and *equity*. However, we would expect no or only occasional co-occurrences between the two word groups, leaving two separate clusters in the graph. Approaches differ in the exact graphs they build and the methods they use to identify the subclusters.

One of the earliest works in this area is Véronis [2004], who builds one co-occurrence graph per polysemous target word. Each graph is based on a specifically retrieved corpus consisting of a few hundred small paragraphs containing the target word. Nodes in the graph correspond to nouns and adjectives, which are linked if they co-occurred in a paragraph. The weighting of the link is based on conditional probability. The sense induction algorithm proceeds in two steps. It first identifies a representative for each cluster by finding highly connected "hubs" and then assigns each node in the graph to one hub using a minimal spanning tree algorithm. Figure 3.1 replicates their illustration that nicely shows the different meanings of the French word *barrage* discovered in this way. The pink nodes are the identified hubs, which can act as labels for the sense. Agirre and Soroa [2007] further developed this approach by improving the original hub search algorithm and extended it to also be applicable to verbs. Their system achieved the best result at the SemEval 2007 Word Sense Induction Task, outperforming the non-graph-based systems.

Dorow et al. [2004] show that similar structures exist in a *global* co-occurrence graph built from the whole British National Corpus of 100m words. Rather than linking words that co-occurred in a paragraph, however, they only link nouns that were found in a coordination, connected by either "and" or "or", or were separated by commas. They give the example of the coordination *a recognised body or an individual solicitor or registered foreign lawyer*, which leads to the undirected edges *body-solicitor*, *body-lawyer* and *solicitor-lawyer*. They provide a slightly broader range of applications, including measuring the degree of ambiguity of a term, sense induction and the formation of semantic categories. One of their core methods is the use of the measure of *curvature* (also known as the *clustering coefficient*) of a node, which measures the degree of interconnectedness of the *neighbours* of a node with each other. They show that curvature is a good predictor of the number of senses of a word using WordNet as a refer-

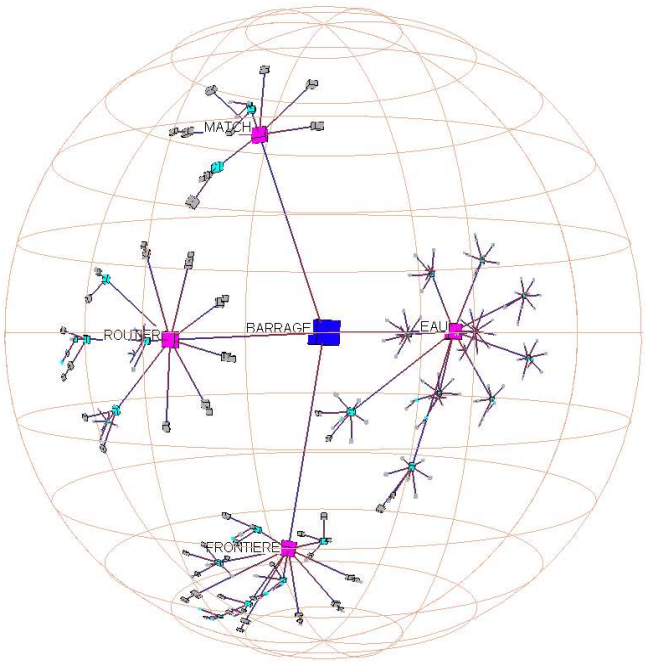


Figure 3.1: Different senses of the French word *barrage* found by Veronis' minimal spanning tree approach. Figure taken from Véronis [2004]

ence. In addition, they demonstrated that removing nodes with low curvature (that is, many senses) leaves the graph split into subgraphs representing semantic categories such as countries.

Not all applications of co-occurrence graphs are based on the exploration of cluster properties, as Lemaire and Denhière [2002] demonstrate. Their graph model is slightly different in that it also incorporates second-order co-occurrences and is built incrementally. Figure 3.2 shows one of their sample graphs. They slide a context window over a text, centering one word at a time. The directed link $A \rightarrow B$ is enforced when B occurs in the window centered at A, and so are - to a lesser degree - all 'second-order co-occurrence links' $B \rightarrow C$. At the same time, the weights of the links $A \rightarrow D$ are decreased for all D which do not occur in the context. On this graph, they measure similarity of terms as the product of the weights on the shortest path connecting the corresponding nodes. They achieve good results on measuring the strength of association between words when evaluated on a dataset for which humans were asked to freely associate words with a target word. Despite the slightly different graph model, their path based association measure is somewhat independent from the exact model at hand and could be applied to the previous graphs, with the loss of the directional asymmetry.

3.1.1.2 Syntactic Co-occurrence

The two models in this category use a syntactically richer notion of co-occurrence. They parse the text using a dependency grammar parser and then link words that were connected by a specific dependency path. Figure 3.3 shows the dependency parse of the sample sentence *The*

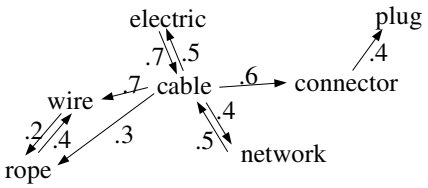


Figure 3.2: Illustration of Lemaire's networks. Taken from [Lemaire and Denhière, 2002]

student gets 15 credits for each module that he takes. The formalism represents syntax in the form of directed head-modifier dependency relations between words. In the sample sentence, the head *get* is modified by *student* over a subject and by *credits* over an object dependency relation. Through this inherent graph structure of the parse, this syntax formalism is particularly suitable to be straightforwardly integrated into a global graph model.

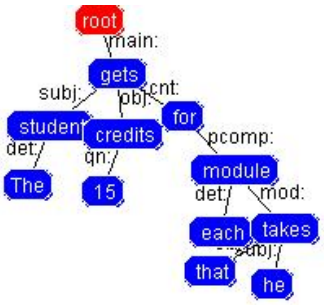


Figure 3.3: Dependency parse of sample sentence, created using the online parser `www.connexor.eu`

Ferrer i Cancho and Solé [2001] build a global graph by simply integrating all of these 'mini sentence graphs' into one global graph. This integrated syntactic network hence consists of a node for *every word* that occurred in the corpus, including determiners and function words, and a link from node i to node j if a head-modifier dependency $i \rightarrow j$ was observed in syntactic dependency tree at least once. The links are unweighted. Figure 3.4 depicts the building process. Strictly speaking, the networks barely classify as a term network due to this inclusion of all function words. As part of their research, Ferrer i Cancho and Solé [2001] build their graph from the British National Corpus and conduct an extensive analysis of the structure of the network, covering, for instance, the degree distribution or the average shortest path length. The application lies in the area of language acquisition and structure analysis rather than in lexical knowledge representation and acquisition. For example, they observe that children's syntactic networks only gradually start showing the adult's network characteristics (eg. hubs are acquired later) and that the properties can differ between languages [Ferrer i Cancho et al., 2004].

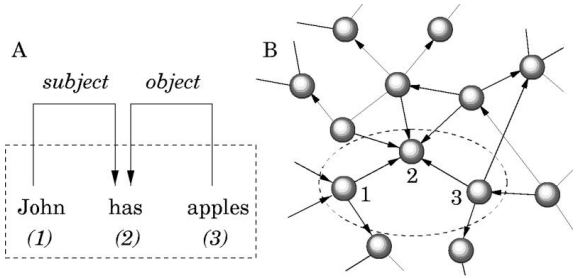


Figure 3.4: Illustration of the integration into a global syntactic dependency network as presented in Ferrer i Cancho et al. [2004]

Herdağdelen et al. [2009] take a more fruitful direction in terms of conceptual knowledge acquisition and representation. They link two target nodes (*noun-noun*, *noun-verb* or *noun-adjective*) in their global network only if they were connected by a specific fixed set of frequent dependency paths, such as 'A-with-B'. Of all approaches presented in this chapter, they use the largest corpus of 2 billion words, from which they extract 25 million tuples. Using a restricted random walk model, they build a vector space representation of the meaning of a word and apply it to measuring semantic similarity and semantic priming and to semantic categorisation. They demonstrate that their model outperforms classical vector space models when the corpus is very small (∼2m words).

3.1.2 Concept Networks

The approaches in this section aim to generate *concept* networks, distinguishing between different senses as well as accounting for synonyms.

Szumlanski and Gomez [2010] automatically acquire a network of WordNet synsets, using the whole of Wikipedia as a corpus. Two synsets are linked in the network if they have been judged sufficiently related. The building process consists of two steps. First, *noun-noun* pairs are extracted for which both nouns are sufficiently frequent in the corpus and for which their co-occurrence based relatedness score exceeds a threshold. Using a combination of three different WordNet-based disambiguation strategies, these pairs are then mapped to pairs of synsets. Figure 3.5 shows a sample network around the synset Tennis. In Szumlanski and Gomez [2011] they show that their networks can achieve competitive results to networks derived from WordNet itself on the Word Sense Disambiguation Task.

While Szumlanski relies on WordNet for moving from terms to concepts, Harrington [2009] uses no external sources. In his work, the semantic network *itself* disambiguates and integrates instances as it is being built by employing a novel spreading activation technique. His core work is aimed at integrating Named Entities, leading to an entity-relation network, a small snippet of which is shown in Figure 3.6. The corpus is syntactically and semantically parsed and the relations found between the named entities are translated into explicit relational paths

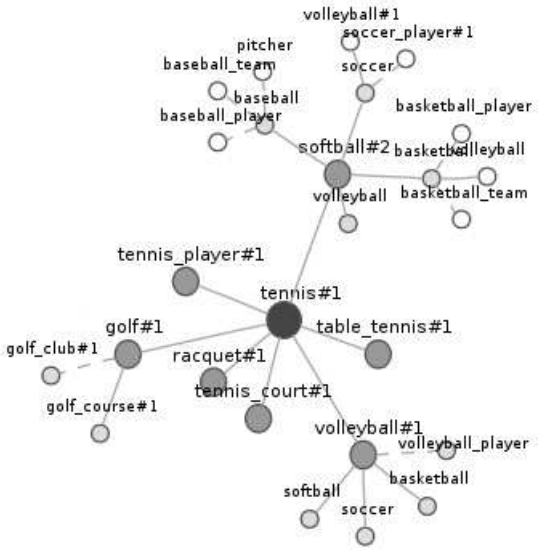


Figure 3.5: Sample local network in Szumlanski and Gomez [2010]

in the network. This leads to a dual role of nodes in the network, representing both named entities and elements on the relational path. In the sample network, we see the two black named entity nodes *Elian Gonzales* and *Cuba* connected through the relational path *return-to*. The disambiguation and integration algorithm can be transferred to induce noun senses, turning the entity-relation network into a concept-relation network. He shows in later work that these concept networks can achieve competitive results to WordNet on the task of measuring Semantic Relatedness [Harrington, 2010]. The combination of concept nodes and explicit relations makes the ASKNet system the most sophisticated method for automatically generating semantic networks from text. A system similar to ASKNet is MindNet [Richardson et al., 1998], which also builds semantic networks based on linguistic parsing. However, MindNet is more restrictive in that it uses a rule-based method to extract a small set of relations from definitions in a dictionary. Definitions display a specific syntactic and semantic structures which make a rule-based extraction viable. As the MindNet approach has not been extended to general corpora, we will exclude the system here.

3.1.3 Overview

The previous section described a variety of automatically generated, corpus-based networks within the broad area of lexical and conceptual knowledge representation and acquisition. As we could see, they all differ in model, size and application. Table 3.1 gives a comparative overview of the reviewed work.

Network Models The models are determined by the information represented by nodes and edges, and we have seen different levels of 'sophistication' or 'semanticness' in both. The core distinction we made was between term versus concept networks, depending on whether

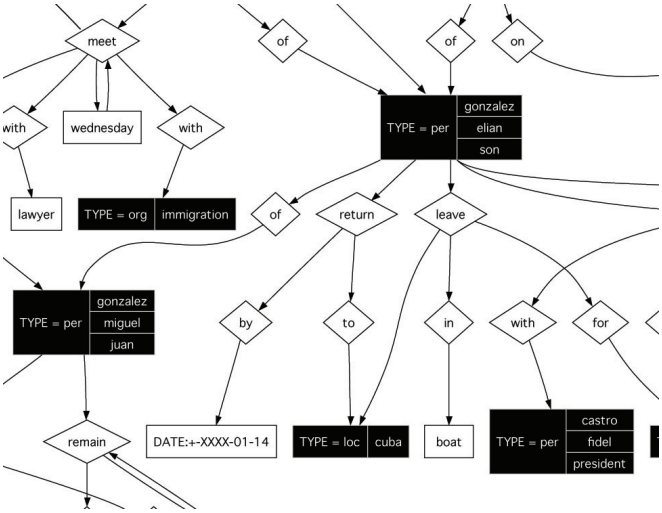


Figure 3.6: ASKNet snippet as presented in Harrington [2009]

the nodes were based on lexemes or distinguished between senses. Similarly, we saw different meanings of edges, representing either superficial co-occurrence within some unit of text or syntactic connections in a dependency parse. The most expressive model encountered was the concept network in Harrington [2009], which is the only model that explicitly keeps track of named relations between the concepts, represented by relational paths. The work reviewed illustrates the whole spectrum from mere co-occurrence graphs to automatically constructed *semantic* networks.

Size Due to the very different models, the sizes of the networks are difficult to compare. In Table 3.1, we list both the underlying corpus size and resulting network size in terms of nodes and edges, where available. Corpus size can be seen as an indicator of scalability of the network model. However, it can be misleading as the different models integrate different amounts of information into the network, which may lead to very different network sizes (e.g. compare Dorow et al. [2004] with Ferrer i Cancho and Solé [2001]). Using deep linguistic processing in contrast to superficial co-occurrence counts means that a scaling to a large corpus is associated with potentially expensive building. Similarly, keeping track of a lot of information such as explicit semantic relations results in large network sizes from smaller corpora (cf. Harrington [2009]). Hence, all aspects of the network model need to be taken into account when comparing network sizes.

Applications The two main applications covered are Word Sense Induction and Disambiguation and measuring semantic similarity, relatedness or association strength between terms or concepts represented by the nodes. The respective methods are based on the detection of clusters [Agirre and Soroa, 2007, Dorow et al., 2004] or on the use of graph exploration algorithms such as random walks [Herdağdelen et al., 2009] and spreading activation [Harrington, 2009].

Results Agirre and Soroa [2007] demonstrated that their graph-based approach was able to outperform all other systems on the SemEval 2007 Word Sense Induction task. Herdağdelen et al. [2009] show that their random walk based distributional similarity model can outperform standard vector space models on semantic similarity tasks when the corpus is very small. Harrington [2010] achieved competitive results to WordNet-based measures of semantic relatedness.

Automatically Generated Networks						
	Nodes	Edges	Corpus Size	Network size		Applications
				nodes	edges	
Term Networks						
Agirre and Soroa [2007]	Nouns, verbs, adjectives	Co-occurrence in paragraph	A couple of hundred paragraphs	n/a	n/a	Word Sense Induction and Disambiguation
Dorow et al. [2004]	Nouns	Co-occurrence within a coordination	100m words (BNC)	48,727	505,412	Ambiguity Detection, Word Sense Induction, Semantic Category Clustering
Lemaire and Denhière [2002]	Non-function words	Significant cooccurrence	3.6m words	n/a	n/a	Measuring association word pairs
Ferrer i Cancho and Solé [2001]	All words	Head-modifier relation in dependency parse	100m words (BNC)	478,773	17m	Language Acquisition, Language Structure
Herdağdelen et al. [2009]	Nouns, verbs, adjectives	Connection through specific set of dependency paths	2bn words	n/a	25m	Semantic Similarity, Semantic Categorisation, Semantic Priming
Concept Networks						
Szumlanski and Gomez [2010]	WordNet synsets	Significant co-occurrence	n/a (Wikipedia)	27,312	84,086	Word Sense Disambiguation
Harrington [2009]	Named Entities / concepts (merged). Relations and attributes (not merged)	Semantic relations (eg. agent, patient)	4.5m words	800,000	1.9m	Novel Fact Discovery, Semantic Relatedness, Semantic Priming

Table 3.1: Summary of related work on automatically generated networks.

3.2 Scope of this Work

3.2.1 The Problem

The review in the previous section showed that there is promising but sporadic work on using automatically generated networks (AGNs) as lexical representations. Two lines of motivation appear, based on the position of the networks between the two poles of (manually) created lexical resources such as WordNet and statistical corpus-based representations of meaning. On the one hand, they can be seen as automatically generated 'approximations' of resource-based semantic networks. On the other hand, we can understand them as more structured representations of the underlying corpus, potentially allowing statistical methods to make better use of the corpus.

AGNs as an automatically generated lexical resources The argument from this point of view is that the networks are similar in structure to the semantic networks commonly derived from lexical resources. This is particularly true for concept networks, which make the distinction into senses - a crucial component of lexical knowledge resources. The potential hence lies in being able to use the automatically created networks when manual resources are not available or lacking coverage, an issue often faced for domain specific terminology and knowledge.

AGNs as structured corpus representations for statistical methods The idea behind this view is that through turning the corpus into a network, a deeper, more structured representation of the underlying information is achieved. In particular, indirect relations between terms or concepts are uncovered and can be capitalised on. The core potential of AGNs from this point of view lies in being able to leverage the network structure and address the issue of sparsity often faced by corpus-based methods. In short, we could have a method at hand to "make more out of less text".

Overall, AGNs have the potential to provide a powerful way of obtaining lexical representations both quickly and from limited data. In particular for domain specific applications, which face a lack of coverage by general lexical resources and may not be able to rely on vast amounts of data, being able to use such networks as a basis could be very beneficial.

However, in order to qualify as a viable alternative, a network approach needs to be scalable and allow for a variety of lexical information to be derived. Scalability crucially involves the ability to build networks from satisfyingly large corpora. But a large representation by itself is not sufficient - it also needs to be demonstrated that the necessary results are achieved consistently. To give an example, Herdağdelen et al. [2009] build the largest network both in terms of corpus size (2bn words) and number of edges in the network (∼25m). Still, their network-based semantic similarity measure only outperforms standard corpus-based vector space models when based on a small 2m words subcorpus - 0.1% of their original corpus. This severely limits the scope on which the networks provide a benefit. In addition, the simplicity of their model using

untyped links between words means that all *descriptions* of the relations are lost, potentially restricting the information that can be generated. In contrast, the networks by Harrington [2009] are built with almost no loss of information from the corpus. However, their networks are generated from a corpus of only 4.5m words, which is very small compared to standard corpora such as the 100m words British National Corpus. This corpus is tailored to the test data on the semantic relatedness task, additionally raising the question to which extent the good results generalise to a general, non-optimised corpus.

The sporadic work on AGNs shows promising results, but a large-scale, convincing case for their use is still to be made. We make this case in our work by generating the largest, most expressive networks to date, showing their wide applicability and demonstrating that they can consistently outperform other approaches.

3.2.2 Aim and Methodology of this Work

The aim of this thesis is to provide the first large-scale demonstration that automatically generated semantic networks present a scalable and versatile basis for lexical representation. While doing so, we want to understand the extent to which the use of concept instead of term nodes adds value to the representation and how the quality of the representations change with corpus size and quality. The work is split in two parts - network generation and application.

Generation We pair the ASKNet model [Harrington, 2009] with large-scale plain text and Wikipedia corpora in a new framework to build the largest, most expressive term and concept networks to date. Our largest plain text term network covers the entire 100m words British National Corpus and includes every single relation extracted from the semantic parse of every sentence. Smaller networks of between 20m and 80m words then provide the basis of investigations of the effect of network growth. In order to create a concept network of comparable scale, we extract a Wikipedia corpus consisting of all first sentences and use the hyperlinks as supervised disambiguation. This way we can generate a high quality concept network to understand the limits of how much can be achieved through this additional level of semantics. To get a direct comparison, we build a simple term network from the same Wikipedia corpus, which at the same time also serves to compare term networks from different corpora by contrasting it to the BNC networks.

Application We picked two very different tasks for our evaluation to illustrate the breadth of possibilities, both of which are specific examples of the two big areas of lexical knowledge. The first one, semantic relatedness, falls into the area of *quantitative* measures of relations between concepts. We approach it by constructing a network-based semantic space model in parallel to the widely applied vector space model in statistical NLP (cf. Section 2.2.1). This allows us to directly demonstrate the gain achieved through using the network structure instead of the text directly. In addition, previous work on the same data set includes a variety of WordNet approaches, which lets us draw conclusions about the quality of the networks as approximations

of manual resources. The second application, noun compound paraphrasing, then requires us to produce *qualitative* descriptions of the relations between two concepts. As statistical approaches to any relation description and extraction often heavily rely on the availability of web-scale data to reduce sparsity, this is an area of application where the networks can be particularly beneficial. It also shows their flexibility - none of the resources we reviewed covers noun compound relations.

Part II

Generation

Chapter 4

ASKNet

Our review of automatically generated networks from text in Chapter 3 clearly identified the ASKNet system by Harrington [Harrington, 2009] to be the most sophisticated and expressive network model. It was initially built to generate entity-relationship networks and uses a spreading activation algorithm to disambiguate and integrate entities during the generation process. In a similar fashion, it can be used to construct concept networks. We base our framework for generating large-scale term networks from plain text and concept networks from Wikipedia on ASKNet, which we therefore review in more detail in this chapter. In Section 4.1 we give a brief description the system. In Section 4.2 we discuss the complexity of the network model and access the scalability of the system.

4.1 Framework

ASKNet builds semantic entity-relationship networks automatically from text in an incremental fashion. It first parses each sentence into a logical form and in turn translates each logical form into a network fragment. A global entity-relationship network is built by integrating those network fragments into one unified network. The update algorithm is based on spreading activation and is used to decide which nodes refer to the same real-world entity. It aims at both distinguishing different entities with the same label (eg. *Paul Newman*, the actor or the computer scientist) and combining nodes with different labels referring to the same entity (eg. *Will Smith* and *Mr. Smith* or *George Bush* and *Mr. President*).

4.1.1 Sentence-level Networks

The translation from sentence to network fragment makes use of two NLP tools, the Clark and Curran parser [Clark and Curran, 2004] and the semantic analysis tool Boxer [Bos et al., 2004], both of which are part of the C&C Toolkit[1]. The parser uses Combinatory Categorial Grammar (CCG) and has been trained on 40,000 manually annotated sentences of the Wall Street Journal. It is both efficient and robust. Boxer is designed to convert the CCG-parsed

[1] http://svn.ask.it.usyd.edu.au/trac/candc

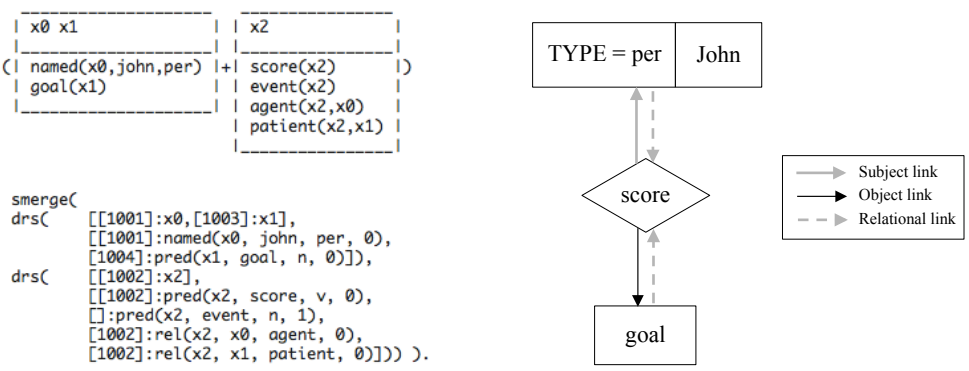


Figure 4.1: Boxer outputs and the corresponding network fragment for the sentence *"John scored the goal."*. Adapted from Harrington [2009]

text into a logical representation based on Discourse Representation Theory (DRT). Figure 4.1 shows the Boxer outputs for the sentence *John scored the goal.* It is given in two versions, the 'pretty print' output for human readability and the Prolog output, which is the one processed in ASKNet. The 'pretty print' output reads as follows: the variables on top of a box are existentially quantified and called 'discourse referents', the predicates in the lower part of the box are further restrictions. Thus the first box translates to

$$\exists x_0, x_1.\ (John = x_0) \wedge goal(x_1)$$

The representation of verbs follows event semantics. In this case, there is an event x_2 with the name *score*, that has x_0, *John*, as an agent and x_2, *goal*, as a patient. The semantic roles are based on VerbNet [Kipper Schuler, 2005].

The ASKNet translation from a Boxer processed sentence into a network fragment is largely straightforward. Figure 4.1 presents the network fragment corresponding to the previous sentence, while Figure 4.3 shows a more elaborate example. The discourse referents from nouns and pronouns (eg. *John*, *chairman*) correspond to object nodes (rectangles), events (eg. *score*) and binary relations (eg. *of*) are translated into relational nodes (diamonds) and attributes into attribute nodes (round rectangles). The direction of relations are preserved using "subject links" and "object links", attributes are attached by "attribute links". Each of these links has a

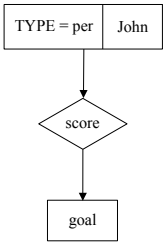


Figure 4.2: Stylised representation of the network fragment in Figure 4.1.

corresponding link in the opposite direction. Subject and object links have a reverse "relational link" and attribute links have a reverse attribute link. This allows for traversal of a relational path in any direction as well as for activation to spread either way if desired. For details on the translation from the Boxer Prolog output to the network, refer to Harrington [2009]. Note that Figure 4.3 is based on an earlier release of Boxer used in the initial publications on ASKNet. The release in use now allows for an easy separation of *nonexecutive director* into an object *director* and an attribute *nonexecutive*, which would be depicted accordingly in the network. Furthermore, publications on the ASKNet system often use a stylised version of the network for easy readability purposes only, depicted in Figure 4.2. It shows the direction in an intuitive way and leaves out the details of link types. We adopt this strategy in our work. However, one should keep in mind that the underlying network structure is always the rich one presented here.

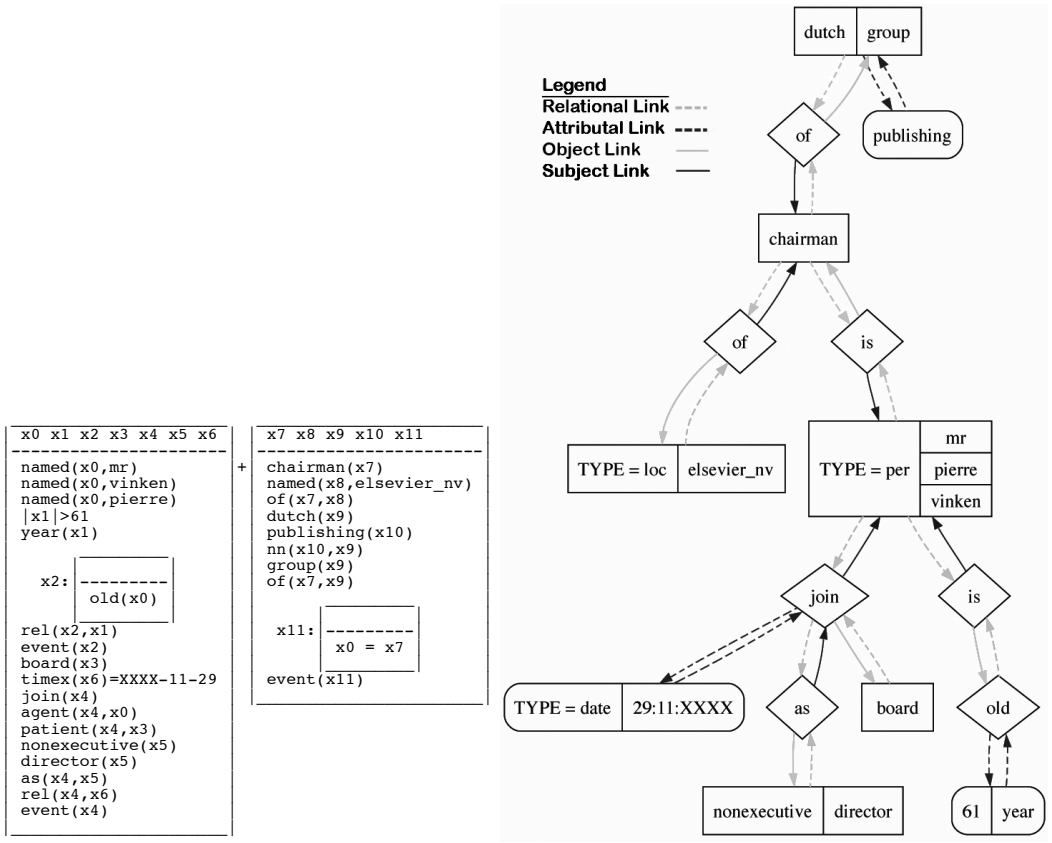


Figure 4.3: Elaborate example of the ASKNet translation for the sentence *"Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group."*. Taken from Harrington [2009]

4.1.2 The Update Algorithm

The network fragments are integrated into a global network using a spreading activation based 'update algorithm' illustrated in Figure 4.4. The network fragments are incrementally added to the large network containing all previous sentences. In this example, the Bush node *bu* could be mapped to *georgebush* or *johnbush* in the large network. In order to disambiguate, the mapping candidates other two entities, *Gore* and *Whitehouse* are retrieved and get activated in the large network. There, the activation spreads over the links, eventually activating the *greorgebush* node, but not the *johnbush* node, indicating that this node is the better mapping. The algorithm then repeats the procedure for the other named entities in the network fragment and induces a self-reinforcing loop. The entities of the fragment are then merged to their maps if a certain level of confidence is reached or they are added as new entities to the big network. For details of the update algorithm refer to Harrington and Clark [2007]. In Harrington [2009, 2010], this update algorithm is adjusted to integrate general concepts instead of named entities.

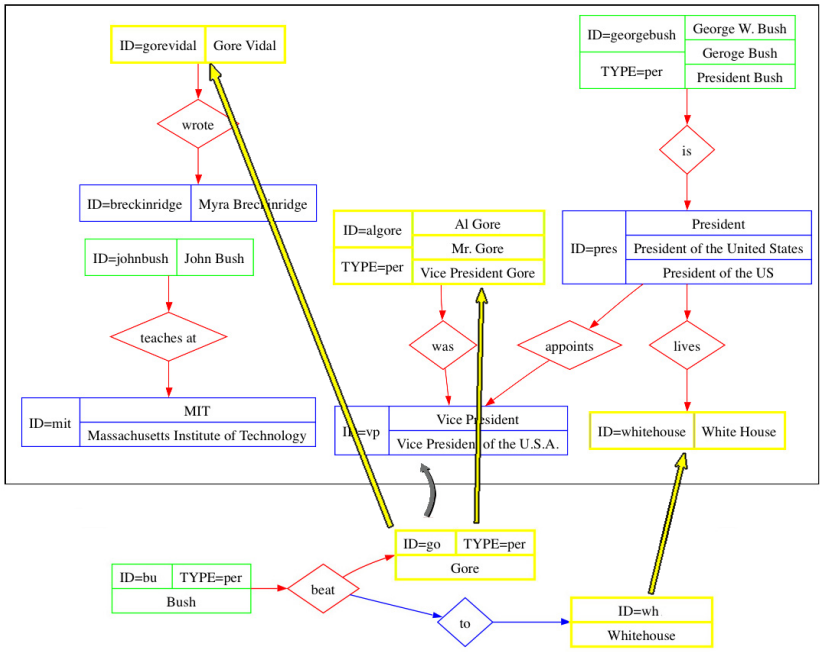


Figure 4.4: Illustration of the update algorithm in ASKNet. Taken from Harrington [2009]

4.1.3 Evaluation and Application

In Harrington [2009], an entity-relation network of 1.5 million nodes (including object, relational and attribute nodes) and 3.8 million links, was built in just under 3 days. The quality of the network was evaluated manually. Human evaluators were given automatically retrieved "cores" of the network. They were asked to evaluate the combined correctness of paths, NE tags and established mappings. The network scored a average precision of 79.1% with a Kappa

Coefficient of 0.69. In a second step, they evaluate the quality of the network indirectly through tackling the Semantic Relatedness task. For that purpose, a corpus of about 4.47m words is retrieved from the web aimed at providing good coverage of all words in the reference dataset. The update algorithm is set to integrate concepts, instead of named entities and the resulting network contains 800,000 nodes and 1.9 million links[2]. Relatedness of two concepts a and b is determined by the amount of activation one receives when the other one fires within a spreading activation algorithm. In later work, they achieve results comparable to WordNet-based approaches [Harrington, 2010].

4.2 Analysis

4.2.1 Network Model and Quality

ASKNet constitutes a *semantic* network in two ways, through the use of a semantic parse as a base for the network and through disambiguation and integration of entities and concepts while building the network. Both determine the quality of the network as well as its scalability in terms of time, memory and processability.

Using the Boxer semantic parse as a basis for the network takes an important step towards considering semantic relationships independently from their varied syntactic surface forms. For example, the syntactical forms *progress of student* and *student's progress* have the same Boxer representation and so do *the student who attends the lecture* and *the student attending the lecture*. This greatly reduces the complexity of the encountered structures and hence allows for a neat translation into a network without the need of many cumbersome rules usually required to process syntactic parses. The ASKNet translation into a semantic network fragment does not make use of all the semantic information given by Boxer. For example, the semantic roles of verb arguments derived from VerbNet are reduced to directions of relations (ie."subject" and "object" links). However, it would be a simple change to allow for a different link type for each semantic role. This opens the possibility to integrate the resulting networks with this standardised resource and any other approach that makes use of it.

Nevertheless, the translation preserves a great deal of the semantic information. A more complex example in Figure 4.5 reveals how ASKNet deals with nesting boxing structures produced for propositional complements (*lets [all lecturers sign his list of modules]*) and implications (*all lecturers sign his list of modules*) as well as for negations, conjunctions and disjunctions. ASKNet translates these by using a parent node structure. For example the implication *all lecturers sign list of modules* is reified by an *imp*-node, which has two dummy children representing the antecedent and the conclusion. This preserved complexity has the advantage of little loss of information and offers great flexibility. We can build the network once and then choose to look only at certain types of relationships or take advantage of all relationships depending

[2]In a second run they use a slightly larger corpus of about 5m words, but the resulting network size is not given. We hence work with the first network as a reference.

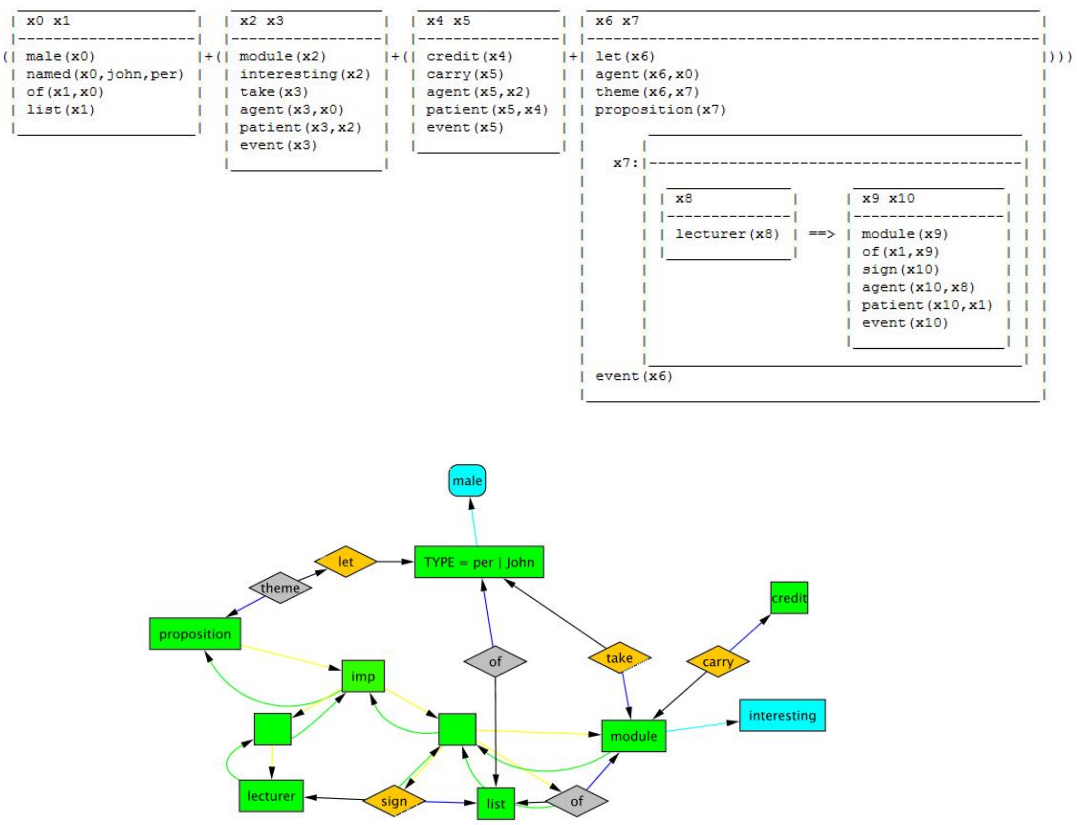


Figure 4.5: Boxer output and network fragment for *"John takes interesting modules. The modules carry credits. He lets all lecturers sign his list of modules."*

on the application at hand. However, it poses great challenges on processing, navigating and mining the network efficiently, several of which we tackle in this work.

Ultimately, the quality of the relations in the network depend on the quality of the semantic analysis tool. The evaluation of the semantic parses is difficult as there is no gold standard available and different types of errors need to be considered separately. Bos [2008] evaluates outputs of seven texts and concludes that in particular the predicate argument structures and resolution of coordinations and conditionals are largely correct. While a good quality of the semantic analysis is desirable, we would expect small errors to be negligible in networks with millions of nodes.

The update algorithm provides both disambiguation and integration for Named Entities. For nouns, it disambiguates only as candidates for mapping are based on lexical overlap (eg, *car* would not be merged with *automobile*). The key selling point of the algorithm is that it takes into account indirect paths between entities that were not stated in the text. For example, in Figure 4.4, the *whitehouse* node in the large network can contribute to disambiguating the *bu* node due to its path over *pres* to *georgebush*. However, this path can have resulted from

previous merging, and not necessarily from a sentence that contained both *georgebush* and *whitehouse*. While the manual evaluation of the named entity merging shows good results, a comparison of the noun word sense disambiguation with previous work on unsupervised word sense induction and disambiguation is missing. This makes it difficult to fully judge the quality of this part of the semantics of the network.

4.2.2 Scalability

In Harrington [2009], a task-specific corpus of 4.47m words was processed into a concept network of 800,000 nodes within 3 days. As our aim is to build a network from a the general-coverage 100m words British National Corpus (BNC), the time complexity of the building algorithm is crucial. The following pseudo code sketches the building algorithm described in Section 4.1.2.

```
1   buildNetworkAskNet(text) {
2      for s : text
3         smallNet = constructSentenceNetwork(s);
4         objNodes = smallNet.getAllObjectNodes;
5         mappingCand = globalNet.getCandidates(objNodes);
6         simMatrix = computeLabelSimilarity(objNodes,mappingCand);
7         j = 0;
8         while (j <= stop && simMatrix.scoreChanges > thr)
9           for (o : objNodes)
10             activations = globalNet.spreadActivation(mappingCand(objNodes\{o}));
11             simMatrix.refineScores(o,activations);
12          end
13          j++;
14        end
15        merge(smallNet,globalNet,simMatrix);
16     end
17  }
```

The global network is built incrementally by creating sentence networks and integrating these into to the main network. Hence, we are interested in the time complexity in terms of number of sentences in the text. `constructSentenceNetworks` contains the linguistic preprocessing and translation into a network fragment and `smallNet.getAllObjectNodes` retrieves all object nodes in the network fragment. For each sentence, both steps are $O(1)$ and bound by the processing time for the most complex/longest sentence in the corpus. Line 5 retrieves all potential mapping candidates for the small network object nodes in the global network. Using a hash table that maps a label to a linked list of all nodes containing the label, this step is again $O(1)$ for each sentence.

Lines 6-14 cover the spreading activation based integration algorithm, which establishes for each object node in the sentence network whether it should be merged with any of its mapping candidates in the global network or added as a new node. The time needed for this integration depends both on the number and type of object nodes / concepts in the small network that are to be added and on the current size and structure of the global network. The former is not easy to estimate in the general case because different types of object nodes will have different num-

bers of potential mapping candidates. In particular, named entities will have a considerably larger number of candidates than object nodes derived from proper nouns. Thus the type of text being processed will have an unusually large effect on the actual complexity. Furthermore, the number of mapping candidates naturally grows as the global network is constructed, starting at none for all nodes, and increasing, but the rate of increase will diminish in a non-linear way, until all senses or unique entities have been added to the network. Again this quantity will be highly dependent on the type of text being analysed, making it very difficult to come up with a reliable general-purpose estimate of complexity. We can reasonably assume that after some time of construction, the number of mapping candidates is bound by a constant for almost all nodes, turning the computation of the initial similarity matrix in line 6 eventually into an $O(1)$ step.

The spreading activation algorithm is then run for each of the object nodes in turn initialised by putting activation into all mapping candidates but those that belong to the object node in question. The similarity matrix is then refined using the spreading activation scores. The whole process is repeated until either the scores changes in the similarity matrix fall below a certain threshold or a fixed number of iterations has been executed, so we can take the 'stop' as an upper bound. The time complexity of the spreading activation algorithm itself is complex to analyse and depends on the node in question, the level of initial activation, the thresholds set, the number of nodes and edges in the network and its structural properties such as average clustering coefficients. For a fixed graph $G = (N, E)$ with |N| nodes and |E| edges, a rough estimate is that a one-directional spreading activation algorithm will visit $\sum_{i=1}^{k}(\frac{|E|}{|N|})^i$ edges, while k depends on the proportion of threshold values and initial activation. In this estimation, $avg = \frac{|E|}{|N|}$ is the average number of edges per node. When the initial node is activated, activation spreads over avg edges, reaching avg nodes. In the next step, activation spreads over avg^2 edges and so on. This gives an average time in $O((\frac{|E|}{|N|})^k)$. In our case of the network building, the average node degree increases with the addition of new sentences. The number of new object nodes in the global network increases as new sentences are added, but flattens as more and more objects have been seen before and are merged. However, the number of relations in the network grows linearly in the number of sentences. Hence, spreading activation is an expensive operation that gets more expensive as the building process moves on. Once the mappings are established, the merging process is in line 15 can be implemented in $O(1)$. Overall, we can see that the bottleneck of the AskNet building process lies in the disambiguation part, in particular the spreading activation operation.

In order to get an idea of the absolute time on a large general coverage corpus, we tried to run AskNet on the British National Corpus (BNC) based on the code Brian Harrington kindly released to us. At about 70,000 sentences, the average time for the integration of a sentence network was 5.14s. Even at a conservative estimation that assumes a constant integration time from then onwards, this would mean a building time of about 300 days for the entire BNC. Our variations replaces the update algorithm with either simple lexical merging or a given

sense annotation provided by Wikipedia hyperlinks. In both cases, there is an unambiguous mapping for every object node. In other words, we replace line 5 in the pseudo code with a O(1) lookup of the whether the global network already contains a node with the same label, skip lines 6-14 and perform the O(1) integration of each object node into the global network. Hence, our network building algorithm is linear in the number of added sentences. Figure 4.6 shows the comparative experimental plot of the AskNet algorithm against ours on the first 100,000 sentences of the BNC.

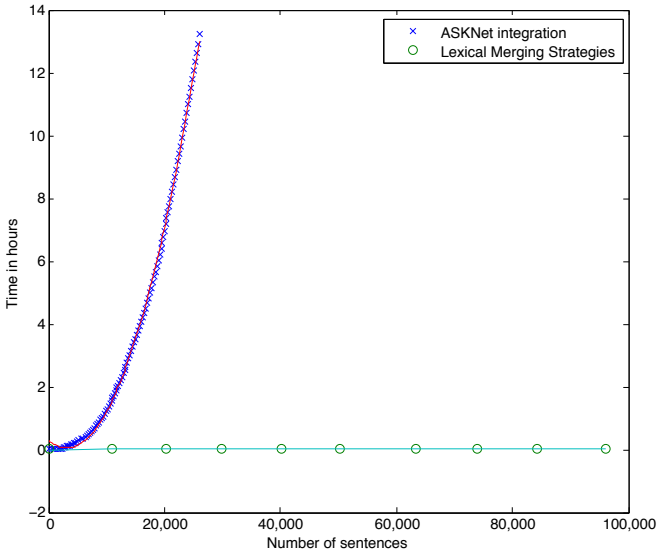


Figure 4.6: Comparison of ASKNet update algorithm with simple lexical merging, based sentences from the BNC.

Chapter 5

Building the Networks

In this chapter, we detail our frameworks for generating term networks from plain text and concept networks from Wikipedia, based on ASKNet. Section 5.1 describes the term network framework, the concrete networks we built from the British National Corpus and analyses their size and building time. Section 5.2 then presents the Wikipedia framework and the network from all first sentences. Section 5.4 compares our networks to the networks we reviewed in Chapter 3. Section 5.5 concludes.

We previously published descriptions of earlier versions of the frameworks and networks in Wojtinnek et al. [2010b], Wojtinnek and Pulman [2011] and Wojtinnek et al. [2012]. This chapter contains revised and improved frameworks and in consequence completely rebuilt networks.

5.1 Plain Text: Term Networks

In order to build expressive term networks, we adapt the ASKNet building framework as far as the translation from text to network fragments is concerned, but replace the integration algorithm with lexical merging. This allows us to judge the advantages of using a network structure over plain text independently of concept disambiguation and to draw a direct comparison to distributional models. As we will show in following sections, using lexical merging strategies enables us to scale the network building to fully cover our 5 million sentence BNC corpus.

5.1.1 Framework

Just as in ASKNet, the network is built incrementally. Object nodes, i.e. those that are derived from nouns and named entities, are integrated to form one unified network from the individual sentence networks. The merging is simply lexical in order to form a term network. For clarity, we refer to object nodes as *term nodes* from now on.

We have three merging strategies that depend on the type of term node:

1. Merge two non-NE term nodes if they have the exact same label.

2. Merge two NE-tagged term nodes if they have the exact same labels and the same NE tag.

3. Split noun compounds into a term node derived from the head of the compound and an attribute node derived from the modifier. Merge the term nodes as in 1.

The first rule basically means that we merge all tokens of the same noun into one node. All occurrences of the noun *student* end up in one node, leading to a generalised representation of the concept *student* by accumulation. This pure lexical mapping does not deal with the imperfect mapping between lexemes and concepts from polysemy and synonymy. We risk accumulating information on different concepts such as *bank*, the financial institution or the river bank, and miss possible generalisations such as those from recognising *car* and *automobile*. The named entity tagger included in the C&C tools we use for the linguistic analysis allows for a splitting of named entities with the same label into different types. We hence use the tags to make sure that two distinct entities such as *W.H. Smith*, the person or the company, are not merged. For noun compounds we found that generalisation is more beneficial to the information in our network than fine-grained distinction.

5.1.2 Networks from the British National Corpus

The British National Corpus (BNC)[1] is a widely used, 100 million word corpus of a variety of British English language samples. It consists of 90% written and 10% transcribed spoken language. It was specifically designed to contain a wide cross-section of British English and includes a variety of texts including newspapers, academic articles, popular fiction as well as letters and school essays. We make use of the written language part only in this work, because the spoken parts contain a high number of incomplete sentences, which parsers cannot deal with. After filtering, we were left with about 4.9 million sentences. We built networks of different sizes in order to investigate the evolution of their structure and to test how the results of our algorithms change with network size.

Table 5.1 lists the number of sentences for each sub corpus from which we built the networks. As we can see, the coverage of the C&C parser and Boxer is high, with 94.7% for the full corpus. Variations in coverage are due to the different sources of text. While the excerpts in the 1 million sentence sub corpus contain many news articles, which are close to the Wall Street Journal text the C&C parser was trained on, the other parts include popular fiction or school essays. Hence the performance drops when more of the latter texts are included. Table 5.2 shows the sizes of the resulting networks, the largest containing 62 million nodes. This overall number contains all node types - term nodes, relational and attribute nodes. As relational nodes are only part of relational paths connecting the term nodes, such as STUDENT-*prepare-for*-TEST with two relational nodes, we gain a better understanding of the network size by listing the

[1] `http://www.natcorp.ox.ac.uk/`

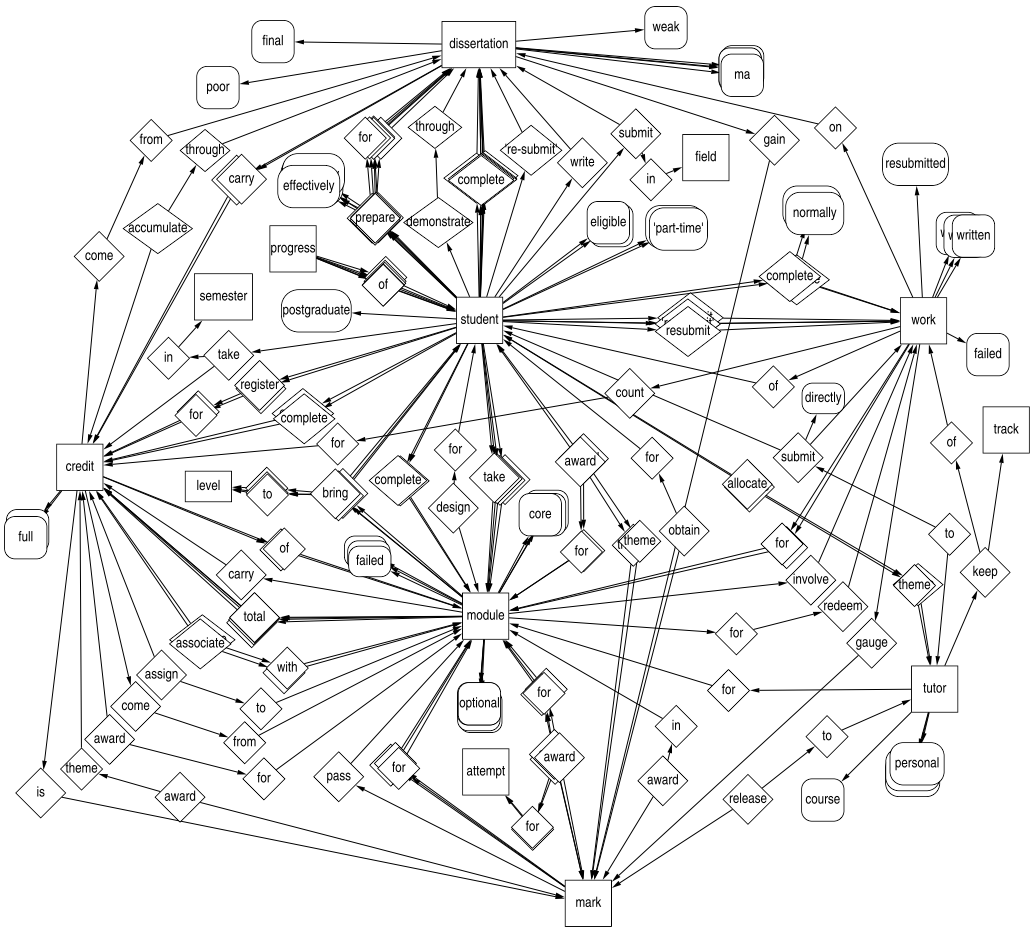


Figure 5.1: Subgraph displaying selected concepts (rectangular nodes), relations (diamond nodes) and attributes (rounded rectangular nodes) from sample network. Figure has been previously published in Wojtinnek et al. [2010b], Wojtinnek and Pulman [2011], Wojtinnek et al. [2012].

number of term nodes and relational nodes separately. The number of effective relational paths then is the number of relations between two term nodes such as *prepare-for*. We will look into this 'dual role' of relational nodes in more detail in Chapter 6.

Naturally, term nodes only make up for a small fraction all nodes of each network. Most of the nodes are part of relational paths or attribute nodes. As to be expected, we can see that the fraction of term nodes gets smaller as the network gets bigger. More concepts are repeated in the text and their occurrences merged. Relations between concepts are collected. We will analyse the network structure in more detail in the next chapter.

The complete construction time including the linguistic preprocessing for the networks was between 2.3 days for the 1 million sentences network and just under 12 days for the full 4.6m

Corpus and Network Sizes					
	1m sent.	**2m sent.**	**3m sent.**	**4m sent.**	**all sent.**
Number of sentences	1,040,665	2,082,714	3,118,332	4,196,430	4,850,885
Successfully parsed and boxed	1,000,000	2,000,000	3,000,000	4,000,000	4,592,967
	(96.1%)	(96.0%)	(96.2%)	(95.3%)	(94.7%)

Table 5.1: Corpus sizes for BNC

Node Statistics					
	1m sent.	**2m sent.**	**3m sent.**	**4m sent.**	**all sent.**
Total number of nodes	14,218,539	28,404,661	42,199,157	54,572,708	62,001,403
Term nodes	359,140	650,821	855,906	1,061,893	1,201.097
	(2.5%)	(2,3%)	(2,0%)	(1,9%)	(1.9 %)
Relation and attribute nodes	13,859,399	27,753,840	41,343,251	53,510,815	60,800,306
Effective relational paths	4,894,151	9,833,170	14,504,550	18,853,638	21,132,280

Table 5.2: Node statistics of all networks.

sentences corpus network (cf. Table 5.3). The largest chunk of construction time is taken up by the linguistic preprocessing including the syntactic and semantic parsing, accounting for 68.2%-91.9% of construction time. As the parsers take one sentence at a time, the linguistic processing time is linear in the number of sentences. However, this can easily be sped up by parallelisation. The actual network building time lies between 4.5 hours for the 1m network and 90 hours for the full corpus network. However, our adapted implementation is not an optimal realisation of the underlying network building algorithm. We saw in Chapter 3 that the building can be implemented in linear time since adding a new sentence network fragment to the global network requires only a constant time lookup of node mappings and a constant time node merging. The constant time mapping lookup requires node labels with more than one part ("John Smith") to be encoded as one label ("John_Smith"). As a legacy of the original AskNet code, this is not the case and a lookup of "John Smith" is performed by searching through a linked list of all nodes containing "John" and checking for "Smith" as a second label, resulting in an inefficient $O(n^2)$ implementation.

Network Construction Times					
	1m sent.	**2m sent.**	**3m sent.**	**4m sent.**	**all 4.6m sent.**
Linguistic Parsing					
Total	$49h\ 39min$	$86h\ 37min$	$133h\ 21min$	$168h\ 28min$	$196h\ 15min$
Av. per sentence	$0.17s$	$0.15s$	$0.15s$	$0.14\ s$	$0.15s$
Network Building					
Total	$4h\ 29min$	$16h\ 43min$	$37h\ 10min$	$64h\ 32min$	$\mathbf{89h\ 58min}$
Total					
Parsing & Building	$\mathbf{54h\ 08min}$	$\mathbf{103h\ 20min}$	$\mathbf{170h\ 31min}$	$\mathbf{233h\ 00min}$	$\mathbf{287h\ 13min}$

Table 5.3: Complete construction times of the networks including linguistic processing and network building.

5.2 Wikipedia Hyperlink Integration

In this part, we build a concept network of similar scale using Wikipedia. Instead of merging "bank", the financial institution, and "bank", the river bank, as we would do in lexical merging, we only want to merge occurrences that refer to the same sense. Intuitively, we would expect the quality of the network to be higher and hence the results on the tasks to be better.

In order to build a high quality disambiguated network, we would ideally need a fully manually sense annotated corpus. The largest such corpus is SemCor[2], a sub corpus of the Brown Corpus with about 360,000 words or 20,000 sentences. Still, the corpus is several magnitudes smaller than the BNC (0.034% on word count basis) and it covers only about 16,000 unique noun senses[3] or concepts, 10% of WordNet. For any real world application, this is too small and an extension would require further manual annotation of corpora.

A different emerging sense annotated corpus comes from the collaboratively edited online encyclopedia Wikipedia. In Wikipedia, each article describes a concept or entity and can be considered a 'sense'. For example, the disambiguation page for 'king' lists some 60 pages including Monarch, King (chess) and King (playing card). Now consider the following first sentence of the article Checkmate:

> **Checkmate** is a situation in **chess** in which one player's **king** is threatened with capture and there is no way to meet that threat.

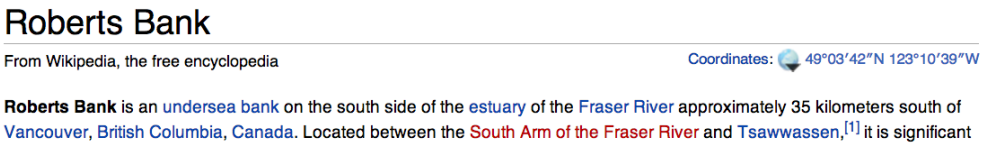
The word 'king' is hyperlinked to the article King (chess), effectively sense annotating this occurrence of the noun. Despite the fact that the annotations are crowd sourced, they have proven to be of very high quality. Mihalcea [2007] trained a standard supervised word sense disambiguation algorithm on Wikipedia annotations and showed that it can even outperform good systems based on manually annotated data. In addition, mappings between Wikipedia articles and WordNet synsets as part of DBPedia mean that we can easily connect any resource built from Wikipedia 'senses' with the most widely used lexical resource. With more than 3.6 million articles in the English Wikipedia alone, this leads to a good quality corpus with an unbeatable coverage of concepts, meaning it provides a source for real world applications and a variety of domains.

5.2.1 Building Framework

The Wikipedia-based networks are built similarly to the plain text networks, but integrate the hyperlinks for concept disambiguation. It evolves around the idea that we can think of an article as referring to a concept and a hyperlinked token in a text as an instance or mention of the concept. Consider the first sentence of the article about "Robert's bank": The standard

[2] `http://www.cse.unt.edu/~rada/downloads.html`

[3] This number is not directly given in the statistics that come with the corpus. It can however easily be computed by extraction of the noun senses from the sense occurrence lists of the two subcorpora brown1 and brown2, concatenation of the resulting files and removal of duplicate lines.



Roberts Bank

From Wikipedia, the free encyclopedia

Coordinates: 49°03′42″N 123°10′39″W

Roberts Bank is an undersea bank on the south side of the estuary of the Fraser River approximately 35 kilometers south of Vancouver, British Columbia, Canada. Located between the South Arm of the Fraser River and Tsawwassen,[1] it is significant

Wikipedia markup syntax is then as follows.

> 'Roberts Bank' is an [[bank_(geography)|undersea bank]] on the south side of the [[estuary]] of the [[Fraser River]]...

The text token "undersea bank" is linked to the article BANK_(GEOGRAPHY), which disambiguates the word "bank" and establishes a relation between the two concepts ROBERTS BANK and BANK_(GEOGRAPHY). For "Fraser River", the name of the token is identical to the name of the hyperlinked article. Our corresponding network fragment would contain three concept nodes ROBERTS BANK, BANK_(GEOGRAPHY) and FRASER RIVER, and relations extracted using the linguistic tools as before. The concept nodes in the Wikipedia network are exactly those that have been identified with an article and we will sometimes refer to them as *article concept nodes* for clarity.

The building framework consists of four steps and is depicted in Figure 5.2. The Corpus Extractor and Preprocessor component extracts the relevant text and produces a plain text version as well as a version preserving the hyperlinks to other articles. The Corpus Extractor part was implemented and hosted by Johanna Völker[4]. It uses a database to represent information from an official Wikipedia snapshot and extracts text from Wikipedia that is cleared of all mark-up apart from hyperlinks. Depending on the objective, the extracted text can theoretically be first sentences, first paragraphs or whole articles and it can either take into account all Wikipedia articles or only a subset for a specific domain. Within this thesis, we consider first sentences only, as we will explain in the following section. In addition to markup removal, the extractor performs elementary *primary entity tagging*, which refers to the marking of occurrences of the article name within the article's text. In the above sample sentence, this means that we annotate the token 'Roberts Bank' as if it was hyperlinked to its article, giving the following final representation.

> [[Roberts Bank]] is an [[bank_(geography)|undersea bank]] on the south side of the estuary of the [[Fraser River]].

This enhances the disambiguations given in the text while ensuring high precision, following the one sense per document assumption. The assumption is in particular valid when we look at first sentences and first paragraphs only. We will refer to the disambiguations as *tags*, covering both original hyperlinks and added primary entity annotations.

[4]Dr Johanna Völker, Data and Web Science Group, Universität Mannheim, Germany. `http://dws.informatik.uni-mannheim.de/en/people/researchers/dr-johanna-voelker/`

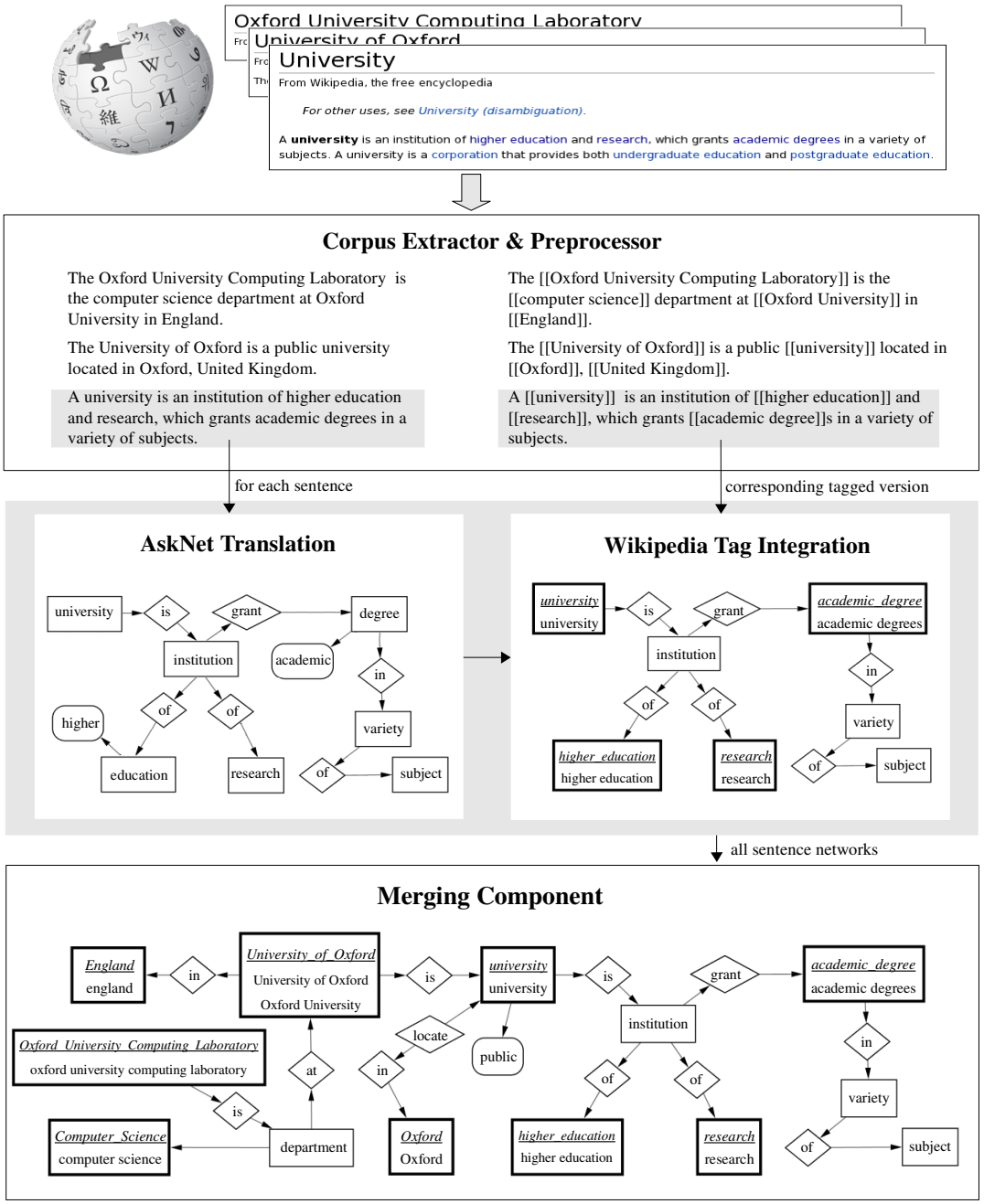


Figure 5.2: Illustration of framework for building Semantic Wikipedia Networks. The figure was previously published in Wojtinnek et al. [2012].

The received corpus runs through our Preprocessor which tailors it to the needs of the building process. The first set of actions clean and filter the corpus. A normalised formatting of the corpus is enforced and all articles for which the sentence extraction failed are removed. In addition, all disambiguation and list pages as well as hyperlinks to them are removed as neither represent concepts in our sense. We chose to also exclude incomplete sentences and sentences with residue markup such as phonetic descriptions. Both would not be parsed correctly by our linguistic tools and lead only to noise. The second set of actions revises the hyperlink tags. In Wikipedia, redirects are resolved only 'after clicking', meaning that the name of the article a token is linked to that we see in the corpus is not necessarily the final target. We hence resolve redirects and normalise capitalisation differences in the preprocessor. We also added an additional rule-based primary entity tagger to complement the tagging produced by the Extractor. Finally, the Preprocessor outputs the revised hyperlinked corpus and a corresponding plain text version.

In the next step, each plain text sentence is then translated into a network fragment in the same way as for the plain text network. In the following Wikipedia Tag component, the hyperlink annotations are integrated into the network fragment. In simple cases, where the token has is exactly one word, as for 'university', we simply turn the corresponding original node into an article concept node and add the article name. However, tokens often consist of more than one word and are represented by more than one node in the plain network, such as for 'academic degree'. Then we collapse these nodes into one newly created article node. Article concept nodes are identified by their article name, but the original plain text tokens are also kept. For example, the article concept node ACADEMIC DEGREE will keep track of the token 'academic degrees' and in the sample sentence above, the node BANK_(GEOGRAPHY) will have the token 'undersea bank'. This is in particular important for word-to-article disambiguation.

Finally, the merging component incrementally integrates all sentence network fragments to form one unified semantic network by merging all article concept nodes with the same article name tag into one. In this way, the OXFORD UNIVERSITY article node from the first sentence is merged with the OXFORD UNIVERSITY article node in the second sentence, while both tokens "Oxford University" and "University of Oxford" are kept.

5.2.2 Networks from Wikipedia's first sentences

For the investigations in this thesis, we chose to extract the first sentences of all articles in Wikipedia as our corpus. With about 3.6 million articles in the Wikipedia dump used here, this not only leads to a corpus of comparable size to the BNC, but also ensures broad coverage of all available concepts. In addition, the first sentences are highly informative as they are usually definitions of the concept and at the same time contain a large number of hyperlinks. In order to isolate the effect of disambiguation from the underlying quality of the corpus, we compare any results directly to the corresponding term network built from the same corpus without hyperlinks following the 'plain text' framework. We will refer to this network as 'plain

Wikipedia Corpus Statistics		
Total number of Wikipedia pages	3,578,709	
- without disambiguation or list pages	**3,379,165**	
Retrieved sentences	2,813,131	83.2%
After filtering noisy sentences	2,750,620	81.4%
Successfully parsed sentences (C&C + Boxer)	**2,706,193**	80.0%

Table 5.4: Corpus statistics and performance of the extractor, preprocessor and linguistic analysis tools.

(Wikipedia) network' to indicate that it is the term network version of the same corpus.
The Wikipedia snapshot of 16 Jan 2010 used for our experiments contains about 3.6 million pages, of which just under 200,000 are disambiguation and list pages. From the remaining 3.4 million pages, the external extractor retrieved about 2,8 million first sentence snippets, achieving a recall of 83.2%. In personal discussions with Johanna Völker, we learned that losses on the recall are primarily due to inherently imperfect rule-based sentence boundary detection. Through complementation with resources such as DBPedia, which explicitly registers the first sentences of Wikipedia articles, the recall could be pushed up considerably. However, we decided that for the purpose of this thesis, the recall was sufficient and leave improvements for future work. The number of final sentences is only lowered slightly to 2.75 million by our additional filter of noisy sentence and parser coverage. Indeed, the C&C Parser and Boxer achieve a combined coverage of 98.4% on the plain Wikipedia corpus, compared to 94.7-96.2% on the BNC. This is difference can probably be attributed to the slightly shorter average sentence length in Wikipedia (19.44 versus 20.74 words per sentence) as well as a lower complexity of syntactical constructs. A summary of the corpus statistics can be found in Table 5.4[5].

Looking at the number of hyperlinks or disambiguations per sentence, we find that the first sentences are indeed well disambiguated. With an average of 3.38 tags per sentence and a tag spanning on average 1.78 words, an average of 6.0 words out of 19.44 words are disambiguated (see '1st generation' in Table 5.5). In comparison, the fully annotated SemCor corpus contains about 4.38 tags to noun-concepts per sentence[6], showing that the first sentences in Wikipedia do not fall short of a fully tagged corpus by much. In particular, more than 2.43 million sentences (89.9%) contain two or more tags and hence contribute at least one relation between two article nodes to the network. Under 49,000 sentences (1.8%) do not contribute much to the network as they do not contain any tagged article.

This corpus leads to a network with 2.17 million article concept nodes and just under 27 million relational and attribute nodes. While this is a very broad coverage, it fell behind our expectations from feeding in 2.7m sentences. As each sentences describes a different article

[5]Small differences to the numbers reported in Wojtinnek et al. [2012] are due to a refined clean up of the retrieved corpus as well as the full removal of disambiguation and list pages for the thesis

[6]Again, this number was calculated from the statistics provided in the downloaded corpus: number of noun sense tags over the two subcorpora brown1 and brown2 (88,312), divided by the number of sentences (20,138).

concept and is usually in form of a definition, we would in general expect them to at least contain a primary entity tag. Hence all article concepts should be represented in the network. However, we find more than 500,000 less article concept nodes in the network than we fed in sentences. Looking into this issue, we noticed that about 924,000 sentences (34.1%) did not contain a primary entity tag, due to the basic tagger considering exact matches of the article name only. This led to misses of concept mentions such as in the following two examples.

> Economy of Antarctica - "There is no economic activity in [[Antarctica]] at present, except for fishing off the coast [...]"
>
> Algorithms for calculating variance - "[[Algorithm]]s for calculating [[variance]] play a major role in [[statistics|statistical]] computing [...]"

In the first example, there is no explicit mention of the entity. In the second example, substrings are already individually disambiguated and further tagging would lead to nested tags. It is difficult to estimate how many sentences indeed do not have a primary entity match. However, the above mentioned cases are most common for long article names (3 or more words), which covers 74.9% of the remaining sentences. In addition, 48.4% contain a tagged substring.

We added a rule-based tagger to the Preprocessor to complement the one in the Extractor. The most effective rules are listed in Table 5.6. They added about 619,000 additional primary entity tags to the 2nd generation of the corpus. A sample based analysis of the remaining 300,000 instances revealed that not all sentences contain an explicit mentioning of the primary entity that could be tagged unambiguously. The combined primary entity taggers appear to cover almost all possible tags and the corresponding 2nd generation network reaches the desired 2.7m article nodes (cf. Table 5.7). The number of article nodes is even higher than the number of sentences, meaning that the network even covers some concepts for which the first sentence was not retrieved through hyperlinks to it in other sentences. As mentioned above, we also build a term network from the same Wikipedia first sentences corpus following the plain text framework. This allows us to isolate the effect of the disambiguation from the quality of the underlying corpus. Table 5.7 shows that the resulting network size is similar to the Wikipedia concept networks, but the term network contains slightly more nodes overall and a higher number of relations. We will look at the qualitative differences between the networks in more detail in the next section.

The total construction time for the Wikipedia network was 90 hours, or just over 3.5 days. As for the networks from the British National Corpus in Part II, most of the construction time is spent on the linguistic parsing, i.e. C&C and Boxer, with 75.5 hours (see Table 5.8). However, it takes more than 40% less time to build the Wikipedia network than the plain BNC network of comparable size. In one part, that is due to a faster processing of the Wikipedia corpus than the BNC corpus with 75.5h against 123.3h, which is likely to be due the shorter average sentence length in Wikipedia and its higher parsing success rate. But also the network building process itself is considerably faster. This can partially be attributed to the different type of

Tags		
	1st generation	**2nd generation**
Hyperlinks		
total	9,272,213	9,890,953
av. per sentence	3.38	3.60
Sentences		
without hyperlinks	48,769	22,705
with ≥ 2 hyperlinks	2,434,115	2,527,852
without PE tag	924,065	305,325

Table 5.5: Statistics on the number of tags (hyperlinks and primary entity annotations) in both versions of the Wikipedia first sentence corpus.

Primary Entity Tagging Rules			
Rule	**Example Article name**	**Sentence**	**Count**
Exact match of non-disambiguation part, ignoring additional white spaces	Aries (constellation)	[[*Aries (constellation)*|**Aries**]] is one of the [[constellation]]s of the [[zodiac]] [...]	270,190
	Edward Waring	[[*Edward Waring*|**Edward Waring** was an [[England|English]] [[mathematician]] [...]	
Exact match of pre-comma part	Austin, Texas	[[*Austin, Texas*|**Austin** is the capital of the [[U.S. state]] of [[Texas]] [...]	140,952
Exact match of one word (two word tags only)	Approximant consonant	[[*Approximant consonant*||**Approximant**s are [[phone (phonetics)|speech sounds]] that [...]	101,788
Discontinuous match with one intermediate word (two word tags only)	Aage Bohr	[[*Aage Bohr*|**Aage** Niels **Bohr**was a [[Denmark|Danish]] [[nuclear physics||nuclear physicist]] [...]	98,399

Table 5.6: Selection of rules used in the primary entity tagger of the Preprocessor component.

Wikipedia Network Sizes			
	Concept Network		**Term network**
	1st generation	**2nd generation**	
Total number of nodes	26,962,361	27,214,232	31,635,544
Concept/term nodes	2,174,147	2,728,387	2,800,321
Relations	n/a	7,122,255	13,440,925

Table 5.7: Network sizes of the two Wikipedia concept networks and the plain term network on the same corpus.

corpus and concept distribution as we can see that the plain version runs faster than the same framework on the BNC. In addition, due to the hyperlink annotation, the building process for the concept network is less effected by the implementation inefficiency in node retrieval described in Section 5.1.2, leading to a shorter building time.

Network Construction Times			
	Wikipedia	**Plain Version**	**BNC (2.75m)**
Linguistic Parsing			
Total		$75h\ 25min$	$123h\ 23min$
Av. per sentence		$0.10s$	$0.15s$
Network Building			
	$14h\ 32min$	$25h\ 29min$	$31h\ 56min$
Total			
Parsing & Building	***89h 57min***	$100h\ 54min$	$155h\ 19min$

Table 5.8: Complete construction times of the networks.

5.3 Network Structures

The three networks we constructed show differences in their size and structure as well as in the concepts they represent. Table 5.9 lists the node and relation statistics of the term and concept Wikipedia networks and the BNC network from a comparable number of sentences. A comparison between the BNC term network and Wikipedia term network allows us to understand the qualitative differences between the networks due to the underlying corpora while differences between the Wikipedia term and concept network illustrate the effects of using hyperlinks to for concept disambiguation.

Firstly, we note that the Wikipedia term network contains less total nodes (including term, relation and attribute nodes) than the BNC network. While this can partially be attributed simply to the higher average number of words per sentence in the BNC corpus (20.74 versus 19.44), the main reason lies in the different number of named entities in the two corpora. The BNC network contains about 800,000 term nodes, 500,000 of which are represent named entities. The Wikipedia term network, however, 2.4 million out of 2.8m terms stem from named

Network Sizes			
	BNC	**Wikipedia**	
	(2.75m)	**Plain Term**	**Concept**
Total number of nodes	38,887,214	31,635,544	27,214,232
Concept/term nodes	817,879	2,800,321	2,728,387
Other nodes	38,069,335	28,835,233	24,485,845
Effective Relational Paths	13,472,719	13,440,925	7,122,255

Table 5.9: Network sizes of the Wikipedia networks, the plain network from the Wikipedia corpus and a BNC network of comparable underlying corpus size.

entities. As named entities often span more than one word, this reduces the number of nodes per sentence. The Wikipedia concept network in turn has less overall nodes than its plain text term equivalent for a similar reason - the hyperlinks often span multiple words. Consider our previous example of the following first sentence after primary entity tagging.

> Algorithms for calculating variance - "[[Algorithms for calculating variance]] play a major role in [[statistics|statistical]] computing [...]"

The term network would contain four nodes from the NP 'algorithms for calculating variance', while the concept network contains only one concept node.

More crucial than the number of overall nodes are the quantitative and qualitative differences between the terms/concept nodes of each network. We already noted that the Wikipedia term network contains a considerably larger number of named entity term nodes than the BNC network. However, for both networks named entity nodes have a small average number of neighbours, i.e. term nodes they are directly connected to with an average of 2.9 for Wikipedia and 6.0 for the BNC. In contrast, nodes stemming from proper nouns are connected to an average of 40.1 and 74.1 term nodes in the two networks, respectively. In consequence, named entities primarily effect the peripheral parts of the networks. Looking at the connectedness of the term/concept nodes in more detail, we find that their node degrees, or number of neighbours, follow a power-law distribution. While most nodes have very few neighbours, there are a few "hubs" with a high number of neighbours. Networks with this property are called *scale-free*[**?**]. Formally,

$$C(k) \sim k^{-\beta}$$

where C(k) is the number of nodes with degree k. Scale-freeness is commonly demonstrated by plotting k vs $C(k)$ on log-log axes, on which a power law distribution looks linear (see Figure 5.3).

A central question then is whether the hubs, or most interconnected terms/concepts differ between the networks. Table 5.10 lists the top ten term/concept nodes with highest degree for the three networks. In the BNC network, the top most linked concepts are general, frequent words such as *time*, *year*, *people* or *part*. With Wikipedia being an encyclopedia which in particular covers named entities, the hubs in the plain term Wikipedia network reflect this difference in corpus. Nevertheless, all but three of the top ten terms nodes in the BNC are also in the top 50 terms in the Wikipedia network and we find a 49.3% overlap between the two networks in the top 1000 terms nodes. However, we find a larger difference between the Wikipedia term network and its concept equivalent, in which the most linked concepts tend to be frequent named entities, in particular countries, leading to an overlap of 25.1% in the top 1000 nodes.

This difference introduced by the hyperlink disambiguation is not surprising. In the term networks, hubs roughly correspond to frequent nouns, while the top concepts in the Wikipedia concept network are those that occurred frequently and at the same time were often considered to be worth linking to an article. The Wikipedia guidelines advise contributors to avoid "over-

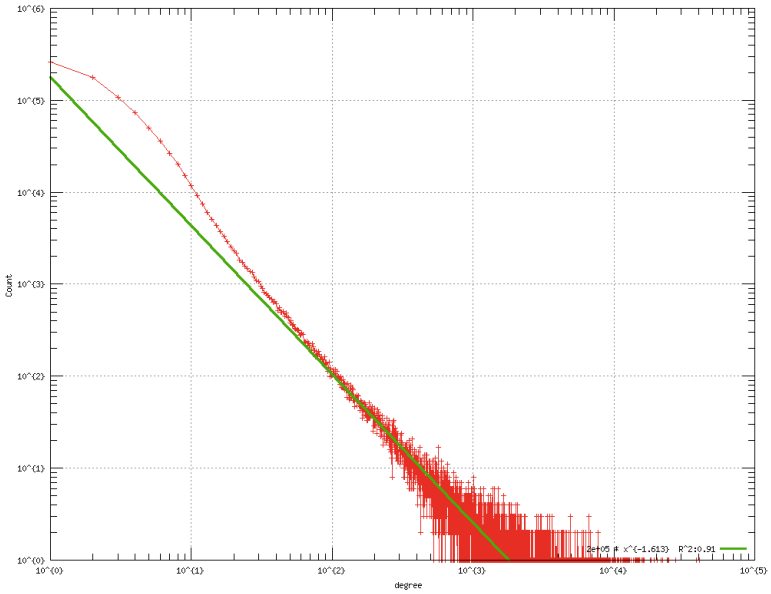


Figure 5.3: Log-log plot of degree distribution of the 2.75m sentences BNC network. The plot was produced using the SNAP library (Stanford Network Analysis Project (SNAP)`http://snap.stanford.edu/`).

Most Connected Terms/Concepts						
	2.75m BNC		**Wikipedia (Term)**		**Wikipedia (Concept)**	
Rank	**degree**	**concept**	**degree**	**concept**	**degree**	**concept**
1	40,295	year	84,402	village	57,599	United_States
2	38,571	time	59,238	album	44,940	Village
3	27,590	way	58,378	player	34,255	Association_football
4	24,632	man	54,089	town	32,844	Communes_of_France
5	24,325	day	51,166	species	28,487	Football_(soccer)
6	24,020	people	47,923	genus	22,646	Genus
7	22,196	part	43,187	footballer	22,472	United_Kingdom
8	21,525	work	42,151	politician	19,674	Canada
9	20,249	group	40,045	name	19,041	England
10	19,260	place	39,912	member	15,532	Australia

Table 5.10: Top 10 terms/concepts with highest number of neighbours.

linking", i.e. to avoid linking plain English words unless particularly relevant to the topic or the article and to add links that are of value to the reader.[7] Therefore, in the sentence "New York is a city in the United States", *United States* is likely to be linked to the state's article, but linking *year* in "Anne Hathaway received an Oscar this year" to the elaborate description of the word *year* and its definition would be against the guidelines. The hubs are in some way complementary in the two types of networks, with the concept network accumulating more

[7]see http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking)#Overlinking

information on concrete entities and concepts and while providing less information on concepts known well to every reader.

Looking at relations, we first of all note that while the number of relations is similar in the two term networks, the Wikipedia concept network contains considerably less relations. Remember that we count as one relation each direct relational path between two term nodes. Hence, in the sentence "The student signed up for a course" we would count one relation *signed-up-for* between the two term nodes *student* and *course*. In the case of the Wikipedia network, we chose to count relational paths between concept nodes only as they form the backbone of the network. This leads to a lower relational count for two reasons. The first comes from non-hyperlinked nouns such as in the case of the sentence ''[[London]] is the capital of [[England]]." where there term 'capital' is considered part of the path between LONDON and ENGLAND. The second reason lies in multi-word spans of the hyperlinks. Consider our previous example "[[Algorithms for calculating variance]] play a major role [...]". In the concept network, the four words correspond to one concept node. In the term network, we would have two term nodes *algorithms* and *variance*, connected by a relation.

We can then look at the token versus type frequencies of the relationships and find that relationship counts follow a power law distribution as demonstrated in Figure 5.4 for the BNC network. On other words, while most pairs of term/concept nodes are connected by very few relations, a few pairs of nodes have a lot of direct relations between them. As Table 5.11 illustrates, the most frequently stated relationships tend to be those between hubs.

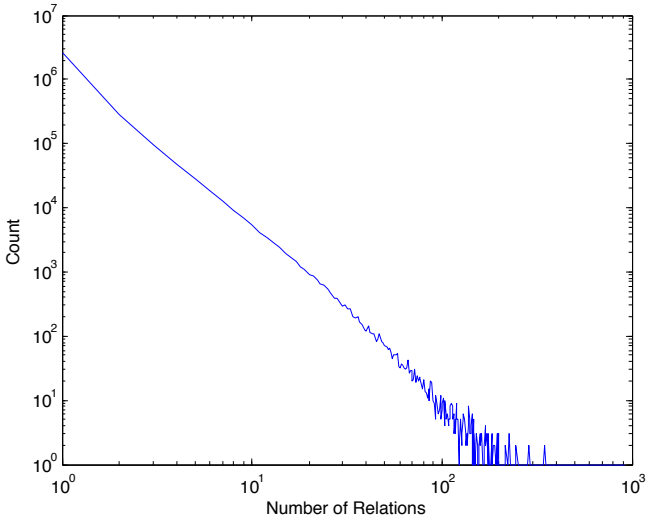


Figure 5.4

Pairs with most direct relations	
BNC	**Wikipedia concept network**
time - year	Departments_of_France - Communes_of_France
man - woman	France - Departments_of_France
day - time	Moth - Noctuidae
view - point	Protein - Gene
people - group	Ontario - Canada

Table 5.11: Top 5 pairs of nodes with highest number of direct relational paths between them.

5.4 Comparison to other Automatically Generated Networks

In this section, we put the size of our networks in perspective by comparison with other term and concept networks. Table 5.12 repeats the network sizes of the related work we reviewed in Chapter 3 (cf. Table 3.1), if they were available. In order to achieve a fair comparison, we give the number of term or concept nodes and the number of relational paths as edges. This follows the rationale that a relational path, consisting of a sequence of relational nodes such as *prepare-for*, semantically act as one relation. As mentioned in the review of the related networks, Harrington [2009] does not make this distinction and only gives the total number of all nodes. We hence provide a separate direct comparison in Table 5.13, in which we list the equivalent reference numbers.

The most appropriate direct comparison for the term networks is Dorow et al. [2004], both because the nodes of their networks represent nouns and their corpus is the BNC. It is clearly visible how the incorporation of every relation from every sentence yields a much larger network with 1.2m term nodes and 21.1m relational paths, in contrast to about 49,000 nodes and half a million edges in their representation. Both Ferrer i Cancho and Solé [2001] and Herdağdelen et al. [2009] have a more comparable number of edges, but their nodes represent not only nouns (and named entities) as in our case, but all words or all nouns, verbs and adjectives, respectively. A direct comparison is hence difficult. The Wikipedia concept network we are dealing with contains about 100 times more nodes and edges than Szumlanski and Gomez [2010], who also build concept networks.

The comparison with the network size in Harrington [2009] (cf. Table 5.13) illustrates the dimension of network growth through consequent scaling to large corpora. From 800,000 nodes and 1.9m links, we are now dealing with up to 62 million nodes and 176 million links. Even from a purely computational perspective, such network sizes pose huge challenges for efficient building and processing. This chapter showed that we can build the networks within a few days. The next chapter discusses several issues we faced in terms of memory consumption as well as when actually *working* with such huge networks.

Automatically Generated Networks					
	Nodes	Edges	Corpus Size	Network size	
				nodes	edges
Term Networks					
Dorow et al. [2004]	Nouns	Co-occurrence within a coordination	100m words (BNC)	48,727	505,412
Ferrer i Cancho and Solé [2001]	All words	Head-modifier relation in dependency parse	100m words (BNC)	478,773	17m
Herdağdelen et al. [2009]	Nouns, verbs, adjectives	Connection through specific set of dependency paths	2bn words	n/a	25m
Full BNC network	Nouns and named entities	Explicit relational path	88m words (written BNC)	**1.2m***	**21.1m†**
Plain Wikipedia Network	”	”	45m words	**2.8m***	**13.4m†**
Concept Networks					
Szumlanski and Gomez [2010]	WordNet synsets	Significant co-occurrence	n/a (Wikipedia)	27,312	84,086
Wikipedia Network	Wikipedia concepts	Explicit relational path	45m words	**2.7m***	**7.1m†**

Table 5.12: Comparison of network sizes with other term and concept networks. (*) term/concept nodes only (†) direct relational paths such as *prepare-for*

Comparison to ASKNet			
	Corpus size	Nodes	Links
Harrington [2009]	4.45m words	800,000	1.9m
Full BNC Network	88m words	62.0m	176.0m
Plain Wikipedia	45m words	31.6m	86.6m
Wikipedia	45m words	27.2m	78.8m

Table 5.13: Comparison with ASKNet network in Harrington [2009]. Nodes include all types (concept, relational, attribute).

5.5 Summary

In this chapter we described our scalable frameworks for building term networks from plain text and concept networks from Wikipedia. We built the largest expressive automatically generated networks to date. Our largest term networks integrates the terms and relations from every single sentences in the 88m words written language part of the British National Corpus and contains 1.2m term nodes and 21.1m explicit relations. The Wikipedia concept networks covers 2.7m concepts and 7.1m relations. Either also considerably exceeds the coverage of the manual resources reviewed in Chapter 2 such as WordNet (120,000 concepts), ConceptNet (150,000 concepts) and CYC (500,000 concepts).

All networks were entirely automatically created with the sole use of linguistic tools and hyper-

links in the case of Wikipedia. The longest total building time including linguistic processing was 12 days for the network from the entire BNC, most of which was spent on linguistic preprocessing (8 days), which could straightforwardly be reduced through parallelisation. In particular, the total construction time for the Wikipedia network was a mere 4 days (14h without linguistic preprocessing). This shows that despite their complexity, the networks scale to very large corpora and can be built on a regular machine on a single core within a span of a few days.

Chapter 6

Implementational Aspects and Infrastructure

In previous chapters we saw that scalability is one of the core challenges of using automatically generated semantic networks. We consider scalability to include two aspects: the time and memory resources needed to *create* the networks as well as the efficiency in *processing* and *working with* the created networks. Both are crucially influenced by two factors: the complexity of the network model with several types of nodes and links and the network size of up to hundreds of millions of nodes. We already addressed building time in Chapter 4 and Chapter 5 and showed that we could build our networks from corpora of up to 100m nodes within a few days. This chapter briefly addresses memory usage but is primarily concerned with the infrastructure needed to work with the networks.

Section 6.1 looks at improvements of the original ASKNet implementation with regard to memory usage. Section 6.2 deals with the complexity of the network model and different conceptual views of the network. In Section 6.3 we describe how we use several standardised outputs in order to take advantage of existing tools for network visualisation, navigation, mining, filtering and analysis. Section 6.4 summarises the chapter.

6.1 Memory Usage

We found that memory usage was as important a bottleneck for scalability as time due to the size, complexity and wealth of information contained in the network representation. Figure 6.1 shows the quadratic time usage of the ASKNet base code, when changed to building term networks.[1] It extrapolates to about 160g RAM for a network from 1 million sentences and more than 3,500g for 5 million sentences. We found that even with the possibility of using the

[1]The memory usage of generating term networks can be seen as a lower bound for the memory usage of the ASKNet system.

Oxford Supercomputer[2], this was not sustainable. In particular, we experienced a considerable blow up in building time when virtual memory had to be used. It was therefore vital to the feasibility of our work to reduce the memory to an extent that we could run all of our experiments on our two servers with 48g and 72g RAM respectively.

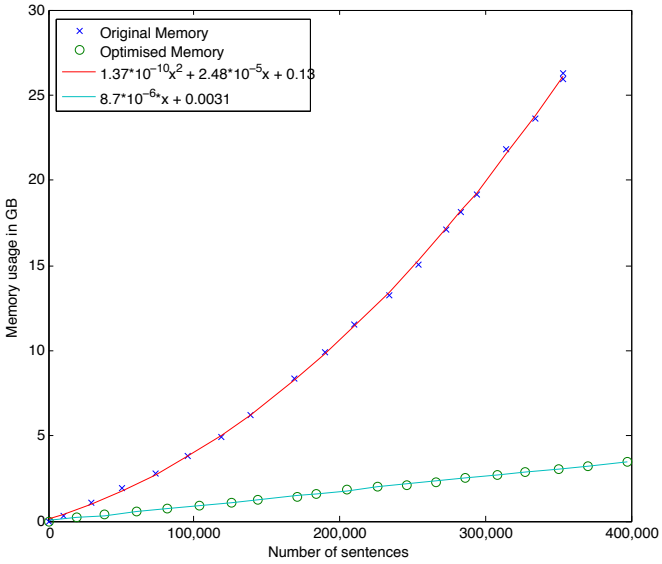


Figure 6.1: Plots of the original and the optimised memory usage.

The major cause for high memory usage turned out to primarily lie in inefficient dealing with the large number of strings as well as unnecessarily created data structures. Linear memory growth with a use of about 44g for the full BNC network was achieved by a few changes such as the following.

- Integer node IDs instead of String IDs. The String IDs originally kept track of the sentence IDs of the nodes that were merged into this node to allow for back tracing. This information is now written to a file.
- String pooling for node labels. That way multiple occurrences of the same label reference to one String object, instead of having a separate String object for each of them.
- Setting initial capacities of extensively used data structures (eg. vectors for links and labels) to minimal size.

6.2 Network Model Complexity and Efficient Processing

The network model we are working with is very complex. It contains several types of nodes (object, relational, attribute, parent) and links (subject, object, relational, attribute, parent,

[2]http://www.oerc.ox.ac.uk/computing-resources/osc

child). In addition, both nodes and links can have several attributes, such as labels and IDs for nodes and weights for links. On the one hand, this means fine-grained semantic information can be distinguished and either made use of or filtered as needed. On the other hand, this complexity comes with challenges for processing the network, in particular when the network has tens of millions of nodes and links. As there is no standardised format for expressing semantic networks such as ours, there are no external tools for processing that meet our needs immediately and comprehensively. The ASKNet code that we based our model on was written in Java and uses self created classes for all parts of the network. It was specifically designed to be processed using spreading activation. In order to establish the flexible use of semantic networks for conceptual knowledge acquisition envisioned here, we had to turn this implementation into a comprehensive processing suite. In our work alone, we were faced with the need to develop solutions for the following

- **Visualisation and Navigation** Being able to 'look' at the network is a key ingredient for analysing and illustrating the network. We cannot visualise a network of millions of nodes, so we need to restrict ourselves to certain areas such as the neighbouring network of a node of interest. While automatic graph layouting algorithms are a crucial component, we ideally want to have an environment that allows us to interactively adjust the layout, by dragging and dropping as well as extending or "exploring" as needed.

- **Path Extraction** The extraction of paths tells us in which way concepts are connected in the network. In our work, we use path extraction to determine paraphrases of noun compounds, which falls in the wider area of *naming* the relations between concepts. Ideally, we want to be able to put restrictions on the paths or return only paths with certain lengths or structures, such as containing certain types of nodes or links. With very large networks, efficiency is crucial.

- **Context Analysis** We are often interested in the surroundings or "context" of a specific node in the network, such as the number of nodes that are one, two or three hops away. When using spreading activation, we would like to be able to monitor how much and exactly which part of the surrounding network gets activated. Moreover, if we have two contexts, we want to be able to compare them and analyse their overlap.

- **Structural Analysis** This part of processing is concerned with the global structure of the network. It includes simple measures such as number of nodes and links, but also density of the network, average shortest path lengths or level of clustering. These measures are important for understanding the behaviour of the network as well as to transfer results to new corpora.

- **Querying and Advanced Graph Algorithms** While not contained within the scope of this thesis, we made sure our implementation allowed for querying. We can, for example, retrieve all object nodes that are subject to a specific verb while also having a specific attribute. We also make it possible to use advanced graph algorithms such as PageRank.

There are two core factors that severely limit the applicability of tools for graphs or networks from other fields achieving these processing tasks.
Firstly, we are dealing with *networks* (different types of nodes and links), and not *graphs* (one type of node and link), meaning that most tools are not able to represent our rich semantic model. Secondly, the model has an inherent 'dual view' rooted in the representation of relations as nodes themselves. On the one hand, using relational nodes allows for a clean representation of tertiary predicates such as *discuss with* in Figure 6.2 as well as for complex structures such as propositions in Figure 4.5. In addition, if we use spreading activation, it allows for fine-grained settings and taking into account the length of a relational path. On the other hand, we also have the inherent view that a relational path acts as a 'link' between two object nodes. In Figure 6.2, we want to be able to say that *dissertation* has neighbours *credit*, *student* and *topic*, to which it it is 'linked' by a relation. However, in any standard graph or network model, its neighbours would be those nodes connected to it by a link, which are the relational nodes *for*, *write*, *submit* and *of* and the attribute node *final*. The issue is complicated by the fact that the relational paths can have very different lengths. In Figure 4.5, the path between *John* and *lecturer* goes over five intermediate nodes. An internal implementation of *getNeighbours* directly on this network model must hence recursively follow all surrounding paths until an object node is found.

Students attend modules and write a final dissertation. Modules provide credits, which the student collects throughout the degree. [..]
Credits are also awarded for the dissertation. The student must discuss the topic of the dissertation with the tutor. Students can pick topics from several fields. [...]
Students must submit the dissertation themselves.

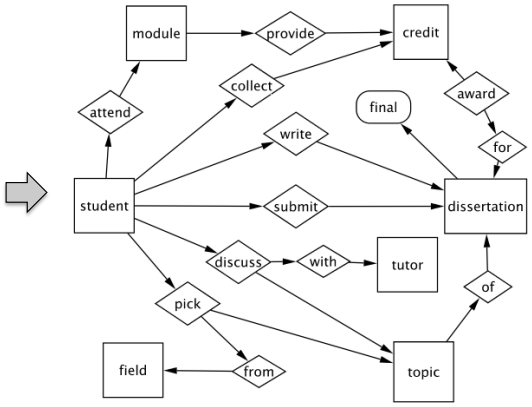


Figure 6.2: Sample text and corresponding term network

6.2.1 Skeleton Graph

This dual view, which already makes a simple method like retrieving neighbouring nodes difficult, is the core issue of most of the processing tasks listed above. We approach this problem by implementing a skeleton graph version of the network, which only contains object nodes and links between them. It abstracts away from relation names and only keeps track of the number of relations. In this graph, object nodes are direct neighbours. The skeleton graph representation is used for both supporting the internal processing tasks of path extraction and context analysis as well as being used as a stand alone version whenever we are primarily interested in the backbone structure, but do not need the names of the relation such as for the structural

analysis of the global network.

Figure 6.3 repeats the sample graph used in Section 6.2 and shows its corresponding skeleton graph. The skeleton graph contains only the term or concept nodes and weighted, undirected edges between them. Two term/concept nodes are connected by an edge if a relational path exists between them in the full network. The weight of the edge is the number of unique relational paths, ie. paths that do not have a node in common. In the sample graph, the term nodes *module* and *credit* are connected by an edge with weight 1 as they are connected by exactly one relational path *provide*. Similarly, *student* and *dissertation* are linked by two relational paths *write* and *submit* and hence have an edge weight of 2. Note that we ignore edge direction. *Tutor* and *topic* are connected by an edge as they are connected through a relational path over *discuss* despite the edge direction changing on this path.

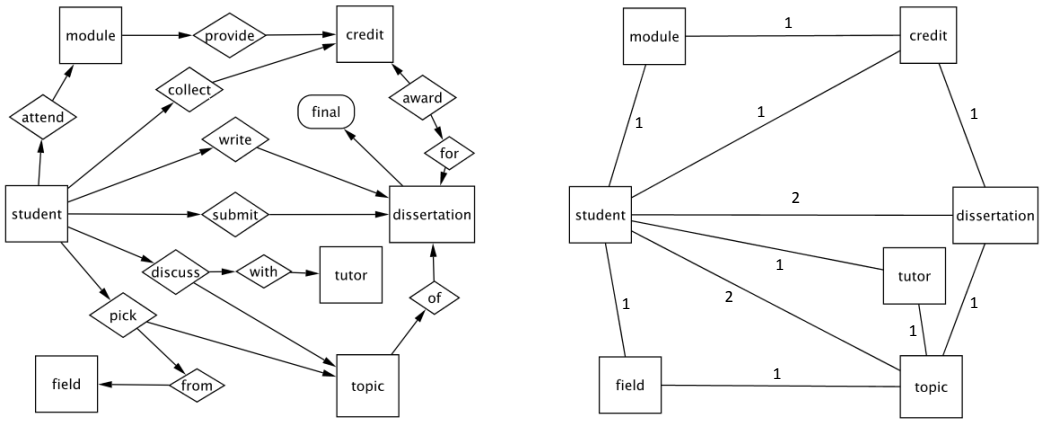


Figure 6.3: Full sample network and corresponding skeleton graph.

In order to create the graph efficiently, it is built incrementally alongside the full network. When a sentence network fragment is created and merged with the big network, the corresponding skeleton graph fragment is derived and merged with the big skeleton graph. With the help of this skeleton graph, we can implement path extraction in the full network and context analysis more efficiently. For example, in order to extract paths between object nodes with intermediate object nodes, we first extract paths in the skeleton graph and then only retrieve direct relational paths in the full network. In order to find paths of maximally one intermediate object node between *module* and *topic*, we would retrieve the object node path *module-student-topic* in the skeleton graph and then retrieve *attend* for the first hop and *pick* and *discuss* for the second hop. This approach allows for a considerably more efficient path extraction than attempting direct retrieval from the full network.

For the implementation, we make use of the Java JGraphT library[3]. It can describe weighted

[3]http://www.jgrapht.org/

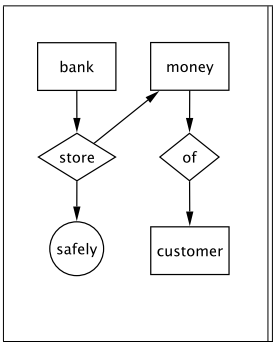
and unweighted, directed and undirected graphs and takes any Java object as nodes. Some basic graph traversal and graph algorithms such as Dijkstra's shortest path algorithm are also included in the library. In order to use external tools for the graph, we implemented graph outputs in several standard formats, which we will look at in the next section.

6.3 Standardised Outputs and Integration with External Tools

Standardised formats of representation for semantic networks are important for both establishing standardised processing tools as well as to ensure that different resources and research results can be integrated. We advocate the establishment of such standards in Harrington and Wojtinnek [2011], as they do no currently exist. In this work, we opted to using standards from related fields. They have different levels of expressiveness and hence capture the rich semantics of our networks to varying extents. We can take advantage of tools available for these standards and ensure interoperability with other resources at the same time.

6.3.1 Resource Description Framework (RDF)

Our core serialisation into the Semantic Web Resource Description Framework (RDF)[4] is the most expressive and semantics-preserving output format. RDF is the basis for the Web Ontology Language (OWL), the standard for expressing ontologies[5]. In a collaboration with Sebastian Rudolph from the University of Karlsruhe, Germany, we found RDF to be well suitable to express most of our semantic network. Figure 6.1 shows our serialisation as presented in more detail in Wojtinnek et al. [2010a].




```
@prefix  rdf:       <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix  x:         <http://example.org/> .
@prefix  link:      <http://example.org/link/> .
@prefix  nodeType:  <http://example.org/nodeType> .
x:id1    rdf:type       nodeType:Object ;     rdf:label  "bank" .
x:id2    rdf:type       nodeType:Event ;      rdf:label  "store" .
x:id3    rdf:type       nodeType:Attribute ;  rdf:label  "safely" .
x:id4    rdf:type       nodeType:Object ;     rdf:label  "money" .
x:id5    rdf:type       nodeType:Prep ;       rdf:label  "of" .
x:id6    rdf:type       nodeType:Object ;     rdf:label  "customer" .
x:id1    link:subLink   x:id2 .
x:id2    link:attLink   x:id3 .
x:id2    link:objLink   x:id4 .
x:id4    link:subLink   x:id5 .
x:id5    link:objLink   x:id6 .
```

Table 6.1: Graph and its RDF serialisation.

With this RDF serialisation, we can make use of various tools for visualisation, efficient storage and querying that were developed by the Semantic Web community to efficiently process large

[4] http://www.w3.org/TR/rdf-primer/
[5] http://www.w3.org/TR/owl-ref/

ontologies. For visualisation, we use the RDF-plugin for Cytoscape[6]. It contains a variety of standard graph drawing algorithms, but, in contrast to other libraries, it also allows for manual drag-and-drop adjustments. In addition, we can navigate or 'explore' the network manually by expanding paths as we wish. Figure 6.4 shows an automatic layout generated by Cytoscape that nicely illustrates hubs in the graph. Figure 5.1 in Section 5.1 was created from this layout by using some filtering in Cytoscape and further manual interactive layouting.

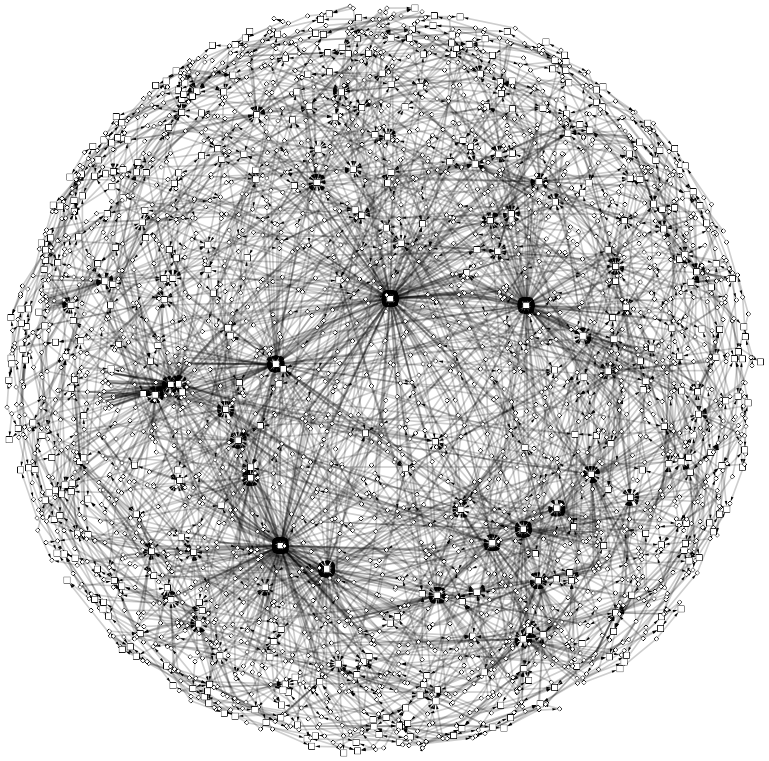


Figure 6.4: Larger sample network showing hubs.

For storage and querying, we can use the Jena framework[7]. The data is stored in RDF databases called *triple stores* and can be queried using the SPARQL Protocol and RDF Query Language [Prud'hommeaux and Seaborne, 15 Jan. 2008]. It allows us to find certain patterns in the networks and even to generate subnetworks. With the following query, we can create a new network that contains all object nodes that are linked to nodes with the label "student" over a prepositional node:

```
CONSTRUCT * WHERE { ?x ?p ?y . ?p rdf:type nodeType:prep .
                    ?x rdf:type nodeType:Object ; rdf:label ?xn .
                    ?y rdf:type nodeType:Object ; rdf:label ?yn .
                    FILTER (?xn=‘‘student’’ || ‘‘?yn="student’’) }
```

[6] http://www.cytoscape.org/
[7] http://incubator.apache.org/jena/

This query would create the union of graph fragments corresponding to phrases like "progress of student" or "student in department" [Wojtinnek et al., 2010a]. In collaboration with Johanna Völker from the University of Mannheim, Germany, we further translated our RDF serialisation into the N-Triples format used in triple stores and set up an online accessible SPARQL endpoint. With this endpoint, we can make our resources publically available. They can line up with the growing number of available data resources published in RDF as part of the Linked Data initiative[8], which includes the well-known structured Wikipedia resource DBpedia[9]. Resources within Linked Data can be integrated and can exchange information due to the common language and the option of identifying common concepts by the same Unified Resource Identifier (URI).

While the RDF representation enables us to use a variety of processing tools and ensures interoperability, we need to note that it cannot fulfil all our processing needs. Although RDF data can naturally be represented in a network, its processing is based on database querying and supports reasoning as used for ontologies. We would not most naturally use the RDF presentation for typical graph algorithms or graph structure analysis.

6.3.2 Graph Modelling Language (GML)

Hence, we implemented a second serialisation in the Graph Modelling Language (GML)[10]. It is a common exchange format for graphs and used by many graph drawing and structure analysis tools as well as graph algorithm libraries. As we can see in the following description of a node and an attached edge, the graph is described structurally ("node 15 is connected to node 7 via an edge with label "4"") and graphically. In contrast to the RDF representation, we cannot semantically encode different types of nodes and links, but need to opt for distinguishing them by their graphics such as shape and colour.

```
graph [
     node [                              edge [
         id 15                               label "4"
         label "student"                     source 15
         graphics [                          target 7
             type "rectangle"                graphics [
             fill "#FF0000"                      fill "#0000FF"
         ]                                   ]
     ]                                   ]
    ...
]
```

We implemented GML outputs for both the full network and the skeleton graph. As set out in the previous chapter, the dual view of the full network is a hindrance to using graph tools exactly in the way we want to, but we can at least use graph drawing tools using the GML output of the network. For the skeleton graph, we use the GML input for the analysis and

[8] http://linkeddata.org

[9] http://wiki.dbpedia.org

[10] http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-technical-report.pdf

processing libraries NetworkX[11] written in Python and the Stanford Network Analysis Project (SNAP)[12] written in C++.

While working with NetworkX and SNAP, we discovered that reading in of a GML graph can be inefficient for very large networks. Hence, we implemented a version using the edge list format. It contains a list of edges in the format

```
source_id       target_id       weight
```

It is the least expressive output and can only capture the structure of the skeleton graph. This level of information can be sufficient for global structure analysis.

6.4 Summary

This chapter outlined the implementational issues we faced while building and working with the huge networks in this work which were addressed by building an extensive infrastructure. We showed that through improvements of the original implementation of ASKNet, we were able to reduce memory usage from quadratic to linear, allowing us to keep the full expressive model in memory. We pointed out the extent to which the complexity of the model and the dual role of relational nodes hinders the immediate usability of external graph querying, processing and visualisation tools. We discussed several serialisations into other graph and network languages, each of which enabled us to take advantage of existing tools and at the same time would allow other researchers from other areas such as Semantic Web Research and Ontology Learning to directly use our networks and integrate them with other existing resources.

[11] `http://networkx.lanl.gov/`
[12] `http://snap.stanford.edu/`

Part III

Application

Chapter 7

A Network-based Semantic Space Model for Semantic Relatedness

In this chapter, we present our network-based approach to measuring semantic relatedness. The quantification of the relatedness of two concepts has proven to be a crucial pre-processing task for many NLP applications including Word Sense Disambiguation, automatic essay marking and co-reference resolution. It has seen a large amount of research and a wide spectrum of approaches, including both WordNet or Wikipedia-based as well as corpus-based methods. However, due to the complex background knowledge and intuition used by humans to judge relatedness scores, it has been a challenge for automatic systems.

In our approach, we construct a network-based semantic space model. In a nutshell, we understand the 'context' of a concept to be the nodes surrounding its corresponding node in the network. Analogous to the distributional hypothesis of semantic space models, the assumption then is that semantically related nodes are connected to a similar set of nodes. Using a spreading activation algorithm, we select the local context around a target node and derive raw values for the nodes in the context. We then translate this activation pattern into a vector representing the target word. We refine the model by conducting a systematic translation of the dimensions of semantic space models to our network scenario. Our model is able to leverage the network structure and incorporate indirectly related concepts into the representation. By comparing our results directly to an optimal text-based model on the same corpus, we demonstrate the gain achieved through using the network instead of the text directly. At the same time, we can compare our results to related work based on WordNet and Wikipedia that has been evaluated on the same standard dataset. This allows us to show the extent to which our networks can compete with manual resources.

Section 7.1 describes the semantic relatedness task and gives an overview of related work. Section 7.2 reviews the dimensions of corpus-based semantic space models and Section 7.3 details our transfer of the model to the network. Section 7.4 contains the experimental parameter

exploration and results on the 1 million sentences BNC network. Section 7.5 looks at the effect of network growth using larger parts of the BNC and Section 7.6 investigates the difference between term and concept networks based on the Wikipedia networks. Section 7.7 conducts an unsupervised combination of the BNC and the Wikipedia results. Finally, Section 7.8 compares our results to related work and Section 7.9 sums up our findings.

7.1 Semantic Relatedness

Semantic Relatedness refers to the measurement of the strength of semantic association of a pair of concepts. It is often confused with the related, but more narrow notion of semantic *similarity*. Similar concepts such as *car* and *motorbike* are related, while *car* and *gasoline* are related but not similar. Judgement of similarity primarily relies on taxonomic relations such as hypernymy, synonymy and meronymy. Relatedness in addition involves non-taxonomical relationships and often subtle differences in their importance to the pair at hand. For example, we may judge *professor* and *student* to be more related through the relation *teaches* than *professor* and *blackboard* through *often uses in lectures*. This poses a problem in particular for approaches based on WordNet which contains only taxonomic relations.

Quantification of relatedness has been a crucial part of a variety of systems for complex NLP applications. For example, Mohler and Mihalcea [2009] automatically grade short student essays by measuring similarity of the student's answer with a reference solution. The text similarity measure combines relatedness scores of word pairs found in the texts. Strube and Ponzetto [2006] add relatedness scores to the features of a co-reference resolution system and show they improve the system. The core idea is that two phrases are more likely to refer to the same entity if their terms are related such as 'Barclays' and 'the bank'. A more extensive review of applications can be found in Zhang et al. [2012].

7.1.1 Approaches

Approaches to Semantic Relatedness can be grouped into knowledge resource-based, corpus-based and hybrid approaches that combine one or more methods. The primary resources used are WordNet and Wikipedia while Roget's Thesaurus and Wiktionary are used less frequently.

WordNet-based approaches can be distinguished by the type and amount of information that is used. Early work on the subtask of semantic similarity focussed on WordNet as a taxonomy and exploited primarily the hypernym-hyponym links between synsets. Similarity of concepts was measured by the length of the shortest connecting path in the hierarchy [Wu and Palmer, 1994, Hirst and St-Onge, 1998] or the information content of their least common subsumer [Resnik, 1995]. When applied to the semantic relatedness dataset, however, these measures give considerably less accurate results [Zesch and Gurevych, 2010]. This is commonly attributed to the fact that relatedness relies on more than taxonomic relationships. Early similarity measures based on WordNet's *glosses*, such as the gloss overlap measure [Lesk, 1986] proved to transfer

to relatedness slightly better, which can probably be attributed to the implicit non-taxonomic information contained in these definitions.

These observations gave rise to methods that translate WordNet into a graph and leverage the whole structure instead of just a single piece of information such as the least common subsumer. Hughes and Ramage [2007] build a straightforward graph in which synsets are connected through their taxonomic relationship and over their shared tokens. A random walk model on the graph, which starts at the target concept, is used to compute an individual probability distribution over all nodes. The relatedness of two concepts is then measured by the similarity of their individual distributions. Agirre et al. [2009] use a similar approach but augment the graph with additional associative links between synsets derived from the manually disambiguated glosses available in later versions of WordNet. Their approach is the currently best performing WordNet-based model and one of the best approaches overall.

The first use of Wikipedia for semantic relatedness was based on the observation that the collaboratively engineered category structure is similar to the WordNet taxonomy. Strube and Ponzetto [2006] applied the path-based, information content and gloss overlap measures described above to the Wikipedia category structure. While their observation that Wikipedia outperforms WordNet on these measures stands contested [Zesch and Gurevych, 2010], recent research has established Wikipedia-based measures as leading. Following the WordNet graph models, Yeh et al. [2009], Gouws et al. [2010], Yazdani and Popescu-Belis [2010] build Wikipedia graphs based on hyperlinks, with varying complexity. They are processed using either variants of random walk models [Yeh et al., 2009, Yazdani and Popescu-Belis, 2010] or spreading activation [Gouws et al., 2010] as we are. Zhang et al. [2011] use a random walk model on an integrated WordNet and Wikipedia network. However, the best Wikipedia-based models are, in essence, distributional models that make use of the fact that the corpus is an encyclopaedia where each article describes a concept. Gabrilovich and Markovitch [2007] build an inverted index for each word that lists all articles in which it occurred. This leads to a representation of the word in terms of Wikipedia concepts. The weight of the concept for the word is the *tf-idf* value of the word in the article of the concept - its frequency in the article divided by the number of articles it occurred in. Radinsky et al. [2011] improve on the results of this Explicit Semantic Analysis by adding temporal information in form of a distribution history for each concept based on 130 years of the New York Times archive, leading to the best Wikipedia-based results.

Apart from the Wikipedia-based Gabrilovich and Markovitch [2007] model, other semantic space models have been evaluated for semantic relatedness that cover the whole spectrum of context choices. Mitchell and Lapata [2008] present an optimised context window approach on the BNC, a reimplemented version of which we use for our direct comparison with a corpus-based semantic space model. Agirre et al. [2009] looks at both window and syntax-based models using a 1.6 Teraword web corpus. Finkelstein et al. [2002] report results based on documents as context using Latent Semantic Analysis on an encyclopaedia. The evaluations have demon-

strated that word-based semantic space models on unstructured corpora can give competitive results to those using structured resources as a basis, at the expense of needing very large corpora.

The work most related to ours is Harrington [2009], who uses concept networks automatically generated by his ASKNet system and a spreading activation algorithm. His relatedness measure for a word pair (a, b) takes the simple average of the activation a receives when b is fired and vice versa. It can be seen as a path-based measure in contrast to our context-based model. The corpus he uses is specifically retrieved for the WordSim353 dataset. We apply this measure to our networks as well in order to indicate the difference between the two approaches.

The best performing system, Haralambous and Klyuev [2011], trains a Support Vector Machine classifier to combine the output of several systems, namely the Wikipedia-based Explicit Semantic Analysis, a WordNet-based path measure and a web corpus-based distributional measure.

7.1.2 Evaluation

The standard evaluation set for Semantic Relatedness is the WordSimilarity-353 data set [Finkelstein et al., 2002], which provides average human judgement scores of the degree of relatedness of 352 word pairs.[1] The collection contains classically similar word pairs such as *tiger* and *jaguar* as well as topically related pairs such as *movie* and *popcorn*. However, no distinction was made while judging and the instruction was to specifically rate the general degree of semantic *relatedness*. This distinguishes it from the two other datasets that were commonly used in earlier works, the Rubenstein and Goodenough [1965] and Miller and Charles [1991] datasets, which are based on judgements of *similarity*. They are also considerably smaller with 65 and 30 pairs, respectively. For the WordSim-353 set, judges were asked to consider the two words related if they were related in at least one of their senses in the case of ambiguity. For example, *minister-party* would be judged on the basis of *party* referring to a political party. The performance of systems on this dataset is measured by the extent to which they produce the same ranking of pairs by the means of the Spearman Correlation Coefficient ρ.

7.2 Models of Semantic Space

We introduced the concept of a semantic space in Section 2.2.1. Following the distributional hypothesis that we can understand a word by the company it keeps, a word is represented in a vector space of co-occurring words. A similarity in meaning between words can then be judged by the similarity of their respective vectors. Historically, the exact structure and parametrisation of the semantic space such as the choice of context and similarity measure were set individually by each researcher. More recent research has identified the need for a general

[1] The dataset provided under http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353 contains a duplicate word pair *money-cash*, giving 353 pairs in total

theoretical framework as well as comparative experimental evaluations of the different settings. In this section, we describe the formal parameters of semantic space models following Lowe [2001] and review some of the most common setting options in literature.

Lowe [2001] formally defines a semantic space model to be an assignment of each word in a language to a point in a real finite dimensional space. The assignment is specified by a quadruple $\langle B, A, S, M \rangle$ referring to the basis, the association function, the similarity measure and a dimensionality reduction model, respectively.

Basis The basis elements $b_1, ..., b_D$ determine the dimensions of the vector space. Most commonly, these elements are words or lemmas [Schütze, 1992, Lund and Burgess, 1996], but they can also be paragraphs or full documents [Landauer and Dumais, 1997]. The vector for a specific word is based on the co-occurrence frequency of the word with these basis elements. When the basis elements are words, co-occurrence is usually defined in terms of a fixed context-window size N, with N being a crucial parameter of the model. However, co-occurrence can also take grammatical information into account. Hindle [1990] considers words that occur in a predicate-argument relationship, Padó and Lapata [2007] generalise this idea to certain paths in a dependency parse. Apart from the context type, the dimensionality D needs to be chosen. While it is possible to include all of the words which occurred in at least one context for the basis, most researchers perform *basis filtering*, considering either only the M most frequent words [Padó and Lapata, 2007, Bullinaria and Levy, 2007] or those that show the highest variance [Lund and Burgess, 1996] to form the basis $b_1, ..., b_M$. The rationale is that higher frequency words provide reliable statistics for all target words. However, infrequent words may provide better distinction between the meaning of words as they tend to have more specific co-occurrences. The right balance is again a matter of optimisation.

Association Function The Association Function `A` defines the mapping from raw co-occurrence frequencies between the basis elements b_i and the target word t to the final vector entries. The vector for target word t then has the form $v = [A(b_1, t), A(b_2, t), ..., A(b_D, t)]$. A large variety of association functions have been proposed. When A is the identity function, the vector contains the raw co-occurrence frequency counts. These can be turned into an estimation of the conditional probability distribution $P(b_i|t)$ by dividing each entry by the sum of the co-occurrence counts for the target word with $v_i = f(b_i, t) / \sum_{j=1}^{D} f(b_j, t)$. This allows for vectors to be compared using probability based similarity functions. Both functions have the drawback of not accounting for different frequencies of the basis elements which means entries may be high simply due to high frequency of the basis word and not due to a strong semantic relation between the basis and the target word. Hence most approaches use association functions that explicitly factor out chance co-occurrence, with Pointwise Mutual Information (PMI) being the most popular choice. The entries are then $v_i = log \frac{P(b_i,t)}{P(t)P(b_i)}$, which gives a positive value when the observed co-occurrence $P(b_i, t)$ is larger than expected purely by chance. Variants of the PMI are Positive PMI which discards all negative entries [Lin, 1998] and the simple probability ratio which does not use the logarithm [Bullinaria and Levy, 2007].

Similarity Measure The Similarity Measure S is defined as a function that maps a pair of vectors into $\mathbb{R}$ and represents the degree of contextual similarity. The plethora of measures used can be grouped into geometric measures in Euclidian space, (weighted) set-based measures and probability distribution based measures. The most popular geometric measures are the cosine, which measures the angle between vectors in the Euclidian space, and the L_1 or L_2-norm, which measure distance. Set-based measures include unweighted measures such as the Jaccard-coefficient and weighted measures such as Lin's measure [Lin, 1998] that is specifically designed to measure semantic similarity with Positive Mutual Information as an association function. Divergence measures model the divergence between probability distributions, the most prominent of which is the Kullback-Leibler divergence.

Dimensionality Reduction Model A semantic space model is fully specified with the previous three parameters. Additionally, a mapping into a smaller linear subspace can be performed by the model M, reducing the dimensionality of the space. This step has proven valuable in the context of Latent Semantic Analysis (LSA) [Landauer and Dumais, 1997], which uses documents as basis elements. There, Singular Value Decomposition reduces sparseness of the matrix and uncovers 'latent' semantic structure such as topics. For word-based vectors, which are inherently not sparse, dimensionality reduction can still serve to enhance computational efficiency purposes while preserving the quality of the representation [Schütze, 1992, Sahlgren, 2006].

The brief summary shows that there is a large set of possible parameter combinations leading to a variety of semantic space models. A thorough understanding of the interactions and effects of the parameter settings are still an open area of research. Lowe [2001] investigates the exposure of certain settings to variance of word frequency, known to follow a Zipf's power law, from a theoretical point of view. He in particular advocates the use of chance-corrected association functions and the use of cosine similarity over Euclidian distance. In a related experimental survey, Weeds et al. [2004] compare a variety of similarity measurements and detect that they exhibit different bias towards frequency such as judging target words to be similar if they are similarly frequent. Bullinaria and Levy [2007] conduct an extensive experimental exploration of the context-window based parameter space on several tasks including semantic similarity, semantic and syntactic categorisation. Although they observe certain general tendencies such as that the combination of PMI and cosine similarity perform well, the optimal setting is dependent on the task at hand. The parameters turn out to be highly interdependent. In particular the optimal similarity measurement differs depending on the choice of association function. These points are underlined by works such as Padó and Lapata [2007], who also explore optimal parameters for their dependency parse based semantic space model, for which the ideal settings turn out to be different from the ones established by Bullinaria and Levy [2007].

In recent work, experimental parameter exploration is the common practice. Mitchell and

Lapata [2008] established the optimal parameters for a context-window for the WordSim353 on the British National Corpus, which will serve as an important reference model for our work.

7.3 A Network-Based Semantic Space

This section describes the transfer of the semantic space model to the network. In a nutshell, the context of a concept is its local surroundings in the network. The basis elements of the vector space are the other term nodes. We use spreading activation to both determine the exact size and shape of the context and to derive basic weights of the basis elements. The model is further refined by basis filtering, the use of an association function and different similarity measurements. In this section, we describe the alternative settings and parameter choices we subsequently evaluate in our experiments.

Related work exists which derives vectors from spreading activation patterns, such as Guadarrama and Garrido [2008] on ConceptNet and Gouws et al. [2010] on an untyped graph based on hyperlinks in Wikipedia articles. However, to our knowledge, we are the first to directly transfer the dimensions of corpus-based semantic space models to a network structure. This takes the technique further from raw spreading activation patterns to include basis filtering and association functions. In addition, as we use networks directly built from text, it allows us to draw a direct comparison between a network-based and a corpus-based model.

7.3.1 Spreading Activation

Spreading activation is a method for processing networks that is based on neural networks in the human brain. If a neuron gets activated, it "fires" and activates other surrounding neurons. Quillian [1967] introduced one of the first simple spreading activation models for the computational modelling of memory and semantic priming. It merely consists of markers that start in two nodes and are passed on through their neighbours. Once the two markers meet, a path between concepts is established and can be interpreted.

Collins and Loftus [1975] extended the original model to better reflect psychological reality and experimental findings. Rather than viewing activation as a binary feature, they consider it a signal that attenuates with distance. The amount of activation a node receives is the sum of the activation that is passed on to it from its neighbours. The node in turn fires if its activation level is above a certain threshold. Both the attenuation and the threshold ensure termination of algorithm.

Variants of this algorithm exist with similar functionality such as Crestani [1997]. We make use of the implementation in Harrington [2009], which is closely based on Collins and Loftus' model. A certain amount of initial activation is given to the start node and when it is fired, its attenuated activation level is split evenly between all its links. The activation then spreads over its (incoming and outgoing) links to the neighbouring nodes. These receive the activation

and in turn fire if their activation level exceeds a certain threshold. The spreading activation process stops when no node can fire anymore, i.e. each node's activation level is below its threshold. The signal attenuation and firing threshold can be set individually for each node type. This is particularly important as we want to distinguish between relational nodes and object nodes in our work.

Spreading activation is a parallel process in which a node fires as soon as its threshold is met. Implementationally, this process needs to be approximated by a sequential process. In Harrington [2009], the process is split into "pulses". Each pulse p consists of the firing of all nodes in the current firing queue FQ_p and the creation of FQ_{p+1}, which contains all nodes whose activation level exceed their threshold once all nodes in FQ_p have fired. The algorithm is kicked off with $FQ_1 = \{\text{startnode}\}$ and FQ_2 will contain all neighbours of *startnode* for which the activation they received from *startnode* exceeds their threshold.

In the algorithm underlying Harrington's implementation, a node distributes its activation evenly through all of its links when it fires. As this spreading is independent of link direction, this means parts of the node's activation is immediately sent back to those neighbours it just received activation from. Consider a simple graph $A \rightarrow B \rightarrow C$, with start node A, initial activation of 1, a signal attenuation of 80% and a threshold of 0.1 for all nodes. Then B receives an activation of $1 * 0.8 = 0.8$ from A in the first pulse and sends an activation of $\frac{1}{2} * 0.8 * 0.8 = 0.32$ to A and C respectively in the second pulse. In the third pulse, both A and C send back activation to B. We found that this led to both undesirable reinforcement of the same relationships and high computational costs. Hence we introduced for each node i a list of blocked nodes, which keeps track of all nodes i received activation from since it last fired. When i fires, it distributes activation only to those neighbours that are not on its blocked nodes list. This forces activation to spread 'forwards' only. The list is reset once i fires. In addition, as we are dealing with very large graphs, we relaxed the termination requirement of sub-threshold activation levels for all nodes to allow for a hard stop criterion of a maximum number of pulses T.

More formally, we can describe the algorithm as follows. Each node i has five time-independent properties:

- Node type $type(i) \in$\{concept, relation, attribute\}
- Number of links nl(i)
- Signal attenuation $\beta_i \in [0, 1]$
- Activation threshold $\tau_i \in [0, \infty]$

At time p, $\mathrm{O}^p(i)$ is the current amount of activation node i contains. Node i fires when $\mathrm{O}^p(i) > \tau_i$ and its activation level is set back to zero. At any time p, $\mathrm{O}^p(i)$ consists of two parts:

$$\mathrm{O}^p(i) = \mathrm{res}^{p-1}(i) + \sum_{j \in N(i)} \mathrm{act}_j^{p-1}(i)$$

where

- $\text{act}_j^{p-1}(i)$ is the amount of activation node i receives when node j fires at point $p-1$
- $\text{res}^{p-1}(i)$ is the *residual* activation and consists of any activation i accumulated up to and including iteration $p-1$ but it has not been able to spread because the threshold was not met.

When a node j fires, it splits all its activation $\text{O}^{p-1}(j)$ evenly among its links, ignoring any links to blocked nodes. Thus $\text{act}_j^{p-1}(i)$ can be described as follows:

$$act_j^{p-1}(i) = \begin{cases} 0, & \text{if } \text{act}_i^{p-2}(j) > 0; \\ \beta_i \cdot \frac{\text{O}^p(i)}{\text{nl}(i) - \text{nbl}^p(i)}, & \text{else} \end{cases}$$

with $\text{nbl}^p(i)$ being the number of links to blocked nodes at time p. The algorithm terminates when no node has sufficient activation to fire or a set threshold for maximum pulses $p = T$ is met.

In our semantic space model, we use the spreading activation algorithm to retrieve the context and raw values for each node in the context. The parameters of the algorithm - the initial activation, attenuation and thresholds - influence the size of the context used as well as the relative weighting of the nodes. We will look at different strategies for setting the parameters and experimentally optimise their settings.

7.3.2 Network-based Semantic Space Model Parameters

In this section, we introduce our network based model covering the three core dimensions of a semantic space - the basis, association function and similarity measurement. We also describe our strategies for concept disambiguation in the case of the Wikipedia networks and our experimental methodology used to limit overfitting.

7.3.2.1 Basis - Context and Basic Weights

The basis elements of our semantic space model are the concept nodes in the network, or article concept nodes in the case of the Wikipedia networks:[2]

$$B_{all} = \{\text{node} \in \text{Network} \mid \text{type(node)} = \text{concept}\}$$

Work on corpus-based semantic space models has shown that results can be improved by considering only the most frequent words as basis elements. In our experiments we hence also work with different levels of basis filtering and construct bases containing only the top 100,000, 50,000, 10,000, 5,000, 2,000 and 1,000 concept nodes in terms of their number of links. We denote these bases by B_{100k}, B_{50k}, B_{10k}, B_{5k}, B_{2k} and B_{1k}, respectively.

[2]Initial experiments that included all types of nodes in the context showed negligible changes in result, at high computational cost. We hence decided to work with concept nodes only

We use the spreading activation algorithm to decide which basis elements occur in the context of the target node. The levels of activation they received are used as raw values, which can be directly used in the vector or be further transformed using an association function. To retrieve the context of target node t, we give it an initial activation of 1 and let it fire. The raw value rv of basis element b_i for target node t is the total activation it received through the spreading algorithm initiated at t:

$$rv_t(b_i) = \sum_{p=0}^{T} act_t^p(b_i)$$

Then b_i occurs in the context of t if it got activated, that is if $rv_t(b_i) > 0$. When we use the raw values directly in the model, the vector for t with basis B is

$$v_t = [rv_t(b_1), ..., rv_t(b_{|B|})]$$

This is equivalent to using the identity function as an association function.

Spreading activation allows us to leverage the network structure and to take into account indirect connections between concepts. By setting parameters we can control the size of the context and derive weights that naturally encode distance as well as the number of paths from the target node. If there are no thresholds, the size of the spread is solely determined by the maximum number of pulses parameter. A basis node receives more activation the more paths connect it to the target word. At the same time, the further away the node is from the target node, the more pulses are needed for it to receive activation, if it does so at all. This in turn means a larger distance translates to smaller activation levels, since at every pulse the firing nodes split their activation among their links. In addition, this splitting of activation between links means that the influence of a node is inversely proportional to the number of its links.
Introducing thresholds means nodes only fire if they have accumulated a significant amount of activation, in contrast to just firing as soon as they receive some activation. It naturally limits spread, reduces noise and introduces selection. In our model, we only use thresholds for concept nodes; relational nodes pass on any activation they receive. A standard way of setting thresholds is to simply have a fixed value:

$$\tau_i = \begin{cases} \tau, & \text{if type(i) = concept;} \\ 0, & \text{else} \end{cases}$$

for some constant $\tau \in [0, \infty]$. We perform a coarse grid search with $\tau \in \{10^{-1}, 10^{-2}, ..., 10^{-6}\}$, similar to the exploration in Gouws et al. [2010].

However, a fixed threshold ignores the fact that some nodes are more connected than others and will in general receive more activation. We hence introduce a dynamic threshold that is

inversely proportional to the number of links of the node in question.

$$\tau_i = \begin{cases} \tau \cdot nl(i), & \text{if type(i) = concept;} \\ 0, & \text{else} \end{cases}$$

The signal attenuation parameter β is often used to model declining levels of activation with distance and to ensure termination. In our version of the algorithm, the division of activation by the number of links at every firing in practice already addresses these points. We use signal attenuation to model relational path length. By setting the decay parameter to be smaller than 1, activation attenuates when it passes over sequential relational nodes. The experiments include setting of no decay ($\beta = 1$), 10% decay ($\beta = 0.9$), and 20% decay ($\beta = 0.8$) for relational nodes.

7.3.2.2 Association Function

The association function in a semantic space model is used to transform raw co-occurrence frequencies to account for different underlying frequencies of the basis elements. Similarly, we work with a variety of association functions to transform the raw spreading activation values to deal with different levels of connectedness of the basis concept nodes. The vector for t is then

$$v_t = [A(b_1, t), A(b_2, t), ..., A(b_{|B|}, t)]$$

Using the identity function as association function is equivalent to using the raw values.

The first set of association functions is based on the idea of *tf-idf* weighting originating in Information Retrieval, where term frequencies are modified by their inverse document frequency in order to reflect that terms that occur in many documents are less good descriptors of document content. For our network-based model, we divide the raw spreading activation value of a node i depending on its number of links. We experiment with weighting by the number of links, its square root and its logarithm, where the first produces the strongest, the latter the weakest weighting.

$$A_f(b_i, t) = \frac{rv_t(b_i)}{f(nl(i))}$$

where $f(x) = x$, $f(x) = \sqrt{x}$ or $f(x) = log(x)$.

The second set of association functions is inspired by Pointwise Mutual Information, in which $v_i = log\frac{P(b_i,t)}{P(t)P(b_i)} = log\frac{P(b_i|t)}{P(b_i)}$, which gives a positive value when the observed co-occurrence $P(b_i, t)$ is larger than expected purely by chance. In our model, we compare the activation a basis node receives from the target word with the average level it receives from a set of 100

randomly sampled nodes R.

$$rv_{avg}(b_i) = \frac{1}{|R|} \sum_{r \in R} rv_r(b_i)$$

We work with the raw activation fraction,

$$A_{rawFr}(b_i, t) = \frac{rv_t(b_i)}{rv_{avg}(b_i)}$$

the logarithm of the fraction,

$$A_{logFr}(b_i, t) = log(\frac{rv_t(b_i)}{rv_{avg}(b_i)})$$

and the positive logarithm

$$A_{pLogFr}(b_i, t) = max(0, A_{log}(b_i, t))$$

7.3.2.3 Similarity Measurement

The superiority of the cosine measurement has been established in several studies mentioned in Section 7.2.

$$\text{sim_rel}_{cosine}(x, y) = cos(\vec{v}(x), \vec{v}(y)) = \frac{\vec{v}(x) \cdot \vec{v}(y)}{\|\vec{v}(x)\| \ \|\vec{v}(y)\|}$$

However, in Padó and Lapata [2007], Lin's measure outperformed the cosine measure for the Pointwise Mutual Information based association function. As their contexts are syntax based and thus similar to ours, we also evaluate using Lin's measure.

$$\text{sim_rel}_{Lin}(x, y) = \frac{\sum_{b_i \in S(x) \cap S(y)} A(b_i, x) + A(b_i, y)}{\sum_{b_i \in S(x)} A(b_i, x) + \sum_{b_i \in S(y)} A(b_i, y)}$$

7.3.3 Experimental Methodology

The network-based semantic space model presented in the previous section opens a large number setting and parameter combinations. The aim of our experiments is to gain a broad understanding of the space and which settings tend to be preferable, not to finally decide on optimal parameters. Where we are dealing with values (eg. the threshold activation value and the decay), parameter search is intentionally coarse. In addition, in order to avoid gross overfitting, we follow Milne and Witten [2008] and Gouws et al. [2010] in randomly selecting 50 from the 353 word pairs for the setting and parameter exploration.

We conduct a detailed exploration for the 1 million sentences BNC network and establish good principal settings. From there we investigate how results change as the networks grows and the sensitivity of the parameters to this growth. We look into the effect of corpus quality and type by comparing results achieved on the plain Wikipedia network with those on a BNC of similar size. Finally, we isolate the effect of concept disambiguation by contrasting the plain

Wikipedia network with the Wikipedia network that incorporates the hyperlinks.

As a reference, we use the corpus-based semantic space model established by Mitchell and Lapata [2008]. It was optimised on the BNC and for the whole WordSim353 dataset and uses the 2,000 most frequent words as basis (minus stop words), a context-window of 5, cosine similarity and the probability ratio association function

$$v_i = \frac{P(b_i|t)}{P(b_i)}.$$

Using an implementation kindly provided by Edward Grefenstette, we recomputed this model for the corpus subsets we use for direct and fair comparison.

7.4 1 million sentences BNC Network

In this section, we look at a detailed exploration of the different settings and parameter choices of the model. Ideally, we would run all possible versions to get a complete picture, but this proved infeasible. We hence look at an imperfect exploration in two steps. First, we evaluate the spreading activation by itself, i.e. using the full basis B_{all} and the raw spreading activation values in the vector, which is equivalent to using the identity association function. We then pick a handful of promising settings for the second step, the basis filtering and the variation of the association function. Finally, we run the best model from the exploration on the full 353 word pairs and draw the final comparison to the network-internal baselines and the corpus-based model by Mitchell and Lapata.

7.4.1 Raw spreading activation

The aim of this is to gain an understanding of the contexts produced by the different spreading activation settings and to identify the promising ones as a basis for the further refinement of the model. When running the different settings, we found that the spreading activation algorithm can take a very long time to terminate if it runs until no node can fire anymore. We hence coarsely explored a cutoff after 3, 6, 9, 12 and 15 pulses and noticed that results improved with more iterations and but stabilised after about 12 pulses. We use this cutoff point throughout the experiments.

Table 7.1 shows the average size and composition of the contexts produced by the different combinations of fixed versus inverse proportional firing threshold type and the coarse settings for the threshold value. With a fixed firing threshold, the context size grows quickly with a smaller threshold. With a threshold value of $\tau = 10^{-4}$, the average context already covers about two thirds of all term nodes in the network. The inversely proportional threshold is more restrictive, only allowing a node i to fire when it received more than τ*numLinks(i) activation. For low values of τ, the condition is so restrictive that activation does not spread past one hop (e.g. the next term node). Even with the additional run with $\tau = 10^{-7}$, the largest average

context size is considerably smaller than those produced by the fixed threshold.

Average Context Statistics							
Spreading Activation Settings		**Context size**	**Context Composition in % of context**			**Context Selection in % activated**	
Threshold type	Threshold value τ		1 hop nodes	2 hop nodes	>2 hop nodes	2 hop nodes	>2 hop nodes
fixed	10^{-1}	2363	92.9	3.6	3.4	1.2	0.1
	10^{-2}	6937	53.1	43.4	3.4	7.1	0.2
	10^{-3}	96759	2.0	90.7	7.3	54.1	1.4
	10^{-4}	233716	0.8	74.6	24.6	92.1	14.7
	10^{-5}	307132	0.6	61.8	37.6	96.4	31.9
	10^{-6}	335379	0.6	56.8	42.6	96.6	39.6
inversely prop.	10^{-1}	2018	100.0	0.0	0.0	0.0	0.0
	10^{-2}	2018	100.0	0.0	0.0	0.0	0.0
	10^{-3}	2019	94.5	2.0	3.4	0.0	0.0
	10^{-4}	2092	81.2	15.2	3.6	0.1	0.0
	10^{-5}	3649	48.3	47.7	4.0	2.6	0.1
	10^{-6}	16239	13.9	80.8	5.4	12.0	0.3
	10^{-7}	110886	1.6	86.7	11.7	56.1	2.7
$\|B_{all}\|$		359,140					

Table 7.1: Average context sizes and context compositions produced by the different spreading activation settings over the 50 sample pairs.

From the context analysis, we can see clearly how the different spreading activation settings lead to contexts that contain different amounts of information from indirect links. With a fixed threshold of $\tau = 10^{-1}$, for example, the contexts are on average composed of 92.9% 1-hop nodes and only 3.6% of nodes that are two hops away. In contrast, the fixed threshold $\tau = 10^{-4}$ is composed of 0.9% 1-hop nodes, 74.6% 2-hop nodes and 24.6% nodes that are more than 2 hops away. Similarly, we can see from the context selection analysis how thresholds restrict activation. For example, the fixed threshold contexts with $\tau = 10^{-3}$ only contains 54.1% of the 2-hop nodes and 1.4% of those all those that are further away. In other words, the algorithm makes the choice that some hops that are 3 or more hops away are still more important to the target word than about 50% of those that are more directly connected to it.

Table 7.2 shows the results we get on the Semantic Relatedness task directly using these raw vectors, i.e. without any basis filtering or association function. This gives us an initial indication which of the contexts constitute good representations. We use both the cosine similarity and Lin's similarity measurement to compare the vectors, because we found that they give very different results. Lin's measure is clearly better suited to evaluate the raw spreading activation based vectors, giving significantly higher results. However, the different similarity measures also favour different contexts. Looking only at the cosine-based results, we could come to the con-

clusion that the context produced by the inversely proportional threshold with value $\tau = 10^{-5}$ is optimal while the more restricted context with a $\tau = 10^{-4}$ is 'too small' and the larger context from 10^{-6} is 'too large'. However, the Lin measure shows a continuous improvement up to a threshold of $\tau = 10^{-7}$. The larger contexts hence are not too noisy, they just need to be compared with a suitable similarity measurement. It appears that Lin's measure is better able to deal with the smaller raw values of more distant nodes in the context. The dependence on further aspects of the model such as the similarity measurement, means we cannot clearly decide on the best parameters for the spreading activation at this stage and we will rather take a handful of good settings into the further experiments in the next section.

Albeit there are the differences in results from the two similarity measurements, a comparison with the respective 1-hop baseline shows including indirectly connected nodes and thus drawing on the network structure significantly increases results. In fact, the highest result is achieved with an inversely proportional threshold and $\tau = 10^{-7}$, the average context for which is composed of 1.6% 1-hop and 86.7% 2-hop nodes as well as 11.7% of nodes at least 3 hops away.

Semantic Relatedness Results of Raw Model			
Spreading Activation		**Spearman Coefficient** ρ	
Threshold type	Threshold value τ	Cosine similarity	Lin's similarity
fixed	10^{-1}	0.006	-0.102
	10^{-2}	0.019	0.345
	10^{-3}	-0.031	0.281
	10^{-4}	0.031	0.116
	10^{-5}	0.056	0.124
	10^{-6}	0.056	0.121
	10^{-7}	0.57	0.416
inversely prop.	10^{-3}	0.01	-0.16
	10^{-4}	0.078	-0.073
	10^{-5}	0.135	0.23
	10^{-6}	0.057	0.234
	10^{-7}	-0.043	0.398
1-hop Baseline		0.007	-0.084

Table 7.2: Results on the Semantic Relatedness task using raw spreading activation values and full basis B_{all}. Based on 50 sample pairs.

At this point, it is difficult to decide whether an inversely proportional or a fixed threshold is more beneficial. If we compare similarly sized contexts - the inversely proportional threshold with $\tau = 10^{-7}$ and the fixed threshold $\tau = 10^{-3}$, the first achieves a higher score. Nevertheless, the reliability of these conclusions seems doubtful at this point. For the further refinements of the model we decided to continue working with both the inversely proportional threshold with $\tau \in \{10^{-5}, 10^{-6}, 10^{-7}\}$ and the fixed threshold with $\tau \in \{10^{-2}, 10^{-3}\}$, because they gave the

best results here under Lin's similarity measure.

Finally, we experimented with adding decay on the relational nodes but found that it did not improve on the results, but rather decreased them slightly. Table 7.3 shows the cosine-based results with 10% (d=0.9) and 20% (d=0.8) decay in comparison.

Effect of decay

thres type	thres value	0.8	0.9	1
fixed	10^{-1}	0.007	0.008	0.006
	10^{-2}	0.016	0.018	0.019
	10^{-3}	-0.026	-0.037	-0.031
	10^{-4}	0.011	0.005	0.031
	10^{-5}	0.038	0.044	0.056
inv.prop.	10^{-1}	0.004	0.007	0.007
	10^{-2}	0.004	0.007	0.007
	10^{-3}	0.003	0.006	0.01
	10^{-4}	0.037	0.055	0.078
	10^{-5}	0.075	0.111	0.135

Table 7.3: Results with varying decay on relational nodes.

7.4.2 Basis filtering and association function

This section looks at the effect of basis filtering and the use of non-identity association functions on the results. We first consider the two in separation before combining them.

7.4.2.1 Basis filtering

Table 7.4 and Table 7.5 show the effect of basis filtering on the results of each setting using the cosine and Lin similarity measurement, respectively. Basis filtering clearly can lead to an improvement under both similarity measures, with an average gain of 0.057 in the case of cosine and 0.029 for Lin's measure between the optimal basis and B_{all}. Given the results of the previous section, it is unsurprising that Lin's similarity measurement achieves the better results.

However, Lin's similarity is more sensitive to basis filtering. While results get continuously better with more filtering and peak at the 1000 top nodes basis for the cosine similarity, the optimal basis varies depending on the spreading activation setting when Lin's similarity is used. Picking the wrong basis size can even drastically *decrease* performance such as in the case of the inverse threshold at 10^{-7} with a drop from Spearman of 0.406 at B_{100k} to -0.04 at B_{10k}. It can even reach the point where no ranking can be established, such as in the case of a fixed 10^{-3} threshold and filtering to 2000 or less basis elements. The reason lies in the nature of Lin's similarity being a weighted Jaccard measure. If the two vectors that are being compared have a non-zero entry for the exact same entries, Lin's measure gives the value 1, independent of the actual value. Looking back at the context statistics in Table 7.1, the settings for which Lin's measure fails to give a ranking are exactly those which produce very large contexts covering

up to almost all of the network. This in particular means that all of the top nodes will get activated. Hence a basis filtering to just those top nodes leads to all vectors having the same non-zero entries and with that all pairs the same score under Lin's measure. We can conclude that, the larger the context or, equivalently the smaller the threshold value τ, the less basis filtering is advisable. Nevertheless, a modest filtering to B_{100k} achieves a gain in all but one of the cases, with an average modest gain of 0.02.

Basis filtering - Cosine Similarity								
Spreading Activation		**Bases**						
Th. type	Th. value τ	B_{all}	B_{100k}	B_{50k}	B_{10k}	B_{5k}	B_{2k}	B_{1k}
fixed	10^{-2}	0.019	0.027	0.024	0.024	0.035	0.044	**0.051**
	10^{-3}	-0.031	-0.025	-0.024	-0.013	-0.005	-0.008	**-0.003**
inv.	10^{-5}	0.135	0.148	0.152	0.152	0.164	**0.195**	**0.195**
prop	10^{-6}	0.057	0.066	0.072	0.07	0.094	0.13	**0.147**
	10^{-7}	-0.043	-0.042	-0.038	-0.035	-0.015	-0.005	**0.033**

Table 7.4: Results with raw spreading activation values, different levels of basis filtering and cosine similarity. The best basis for each case is highlighted

Basis filtering - Lin's Similarity								
Spreading Activation		**Bases**						
Th. type	Th. value τ	B_{all}	B_{100k}	B_{50k}	B_{10k}	B_{5k}	B_{2k}	B_{1k}
fixed	10^{-2}	0.345	0.353	0.342	0.328	0.345	0.355	**0.363**
	10^{-3}	0.281	**0.286**	0.241	0.173	-0.01	-	-
inv.	10^{-5}	0.23	**0.244**	0.243	0.24	0.233	0.181	0.172
prop.	10^{-6}	0.234	0.3	0.263	0.272	**0.335**	0.253	0.092
	10^{-7}	0.398	**0.406**	0.358	-0.04	0.107	-0.149	-

Table 7.5: Results with raw spreading activation values, different levels of basis filtering and Lin's similarity. The best basis for each case is highlighted.

7.4.2.2 Association function

In this section, we isolate the effect of applying different association functions to the raw values, *without* basis filtering. Table 7.6 lists the results based on the *tf-idf* inspired association functions, which divide the raw value by a function f of the number of links. The strictest weighting with f=id, which divides raw value of basis node b_i by its number of links consistently outperforms the other weightings for both measures. The average gain over using the raw value (A=id) is 0.174 and 0.138 for the cosine and Lin's similarity, respectively. We notice that for this group of association functions, Lin's similarity is still a better choice for comparison of the vectors.

This picture changes when we turn to the PMI-inspired association functions (cf. Table 7.7). When the cosine is used, PMI-log association function provides the best results. The raw fraction is the best PMI-inspired association when Lin's measure is used, but it only achieves a comparatively low gain of 0.062. We can see that the cosine measurement is better suited to compare contexts with PMI-based association function.

Association functions: tfidf-based									
Spr. Activ.		**Cosine similarity**				**Lin's similarity**			
type	τ	A=id	f=id	f=sqrt	f=log	A=id	f=id	f=sqrt	f=log
fixed	10^{-2}	0.019	**0.204**	0.102	-0.026	0.345	**0.46**	0.381	0.35
	10^{-3}	-0.031	**0.234**	0.10	0.037	0.281	**0.403**	0.339	0.293
inv.	10^{-5}	0.135	**0.168**	0.138	0.032	0.23	**0.362**	0.253	0.2
prop.	10^{-6}	0.057	**0.185**	0.117	-0.013	0.234	**0.491**	0.346	0.237
	10^{-7}	-0.043	**0.208**	0.051	-0.084	0.398	**0.462**	0.435	0.422

Table 7.6: Results with different tf-idf inspired association functions and full basis B_{all}.

Association functions: PMI-based									
Spr. Activ.		**Cosine similarity**				**Lin's similarity**			
type	τ	A=id	A_{rawFr}	A_{logFr}	A_{pLogFr}	A=id	A_{rawFr}	A_{logFr}	A_{pLogFr}
fixed	10^{-2}	0.019	0.259	**0.292**	0.207	**0.345**	0.236	0.241	0.125
inv.	10^{-5}	0.135	0.061	**0.446**	0.251	0.23	0.164	-0.191	**0.338**
prop.	10^{-6}	0.057	0.146	**0.455**	0.446	0.234	**0.343**	0.26	0.286
	10^{-7}	-0.043	**0.346**	0.303	0.339	0.398	**0.401**	0.123	0.196

Table 7.7: Results with different PMI-based association function and full basis B_{all}.

Overall, the strongest combinations turned out to be the *tf-idf* association function with Lin's similarity and the PMI-logFr with cosine similarity. The first achieved results up to 0.491 with a gain of 0.257, the second tops at 0.455 with a gain of 0.398 over the corresponding raw values.

7.4.2.3 Combining basis filtering and association functions

The two previous sections show that both basis filtering and association functions individually improve on the raw model. However, the optimal choices, the robustness of these optimal choices as well as the amount of gain differed depending on the similarity measure used. For the cosine similarity function, a basis filtering consistently led to an improvement of results, with B_{1k} clearly being the best choice. The best association function was overall PMI-logFr. When Lin's similarity measurement was used, the optimal basis choice was inconclusive, but we could see a consistent though moderate improvement with a small restriction to B_{100k}. Here, the best results were achieved with the tf-idf-id function.

We now bring all aspects of the semantic space together by combining spreading activation settings, basis filtering, association function and similarity measurement. Table 7.8 and 7.9 list

the optimal basis and association function combinations for each case. While the best model is not necessarily simply the combination of the separately best basis and association function, this prediction would not be too far off either. For both similarity measurements, the best association function when taken separately also turns out to be the best in combination. However, the association function and basis interact. For Lin's similarity, we had seen an advantage of using a moderate basis filtering, but in combination with the association function, it appears better to avoid filtering.

Curiously, despite the model differences, under both measures it is the same spreading activation setting that leads to the best result, with negligible differences in the final score: inversely proportional threshold with $\tau = 10^{-6}$. However, this result is exposed to less parameter sensitivity with the optimal model using Lin's similarity. Assuming B_{1k}, A=PMI-logFr and cosine similarity, the average Spearman coefficient of $\tau \in \{10^{-5}, 10^{-6}, 10^{-7}\}$ is $\rho = 0.398$, but it is $\rho = 0.44$ for B_{all}, A=tfidf-id and Lin's similarity. In other words, the latter model gives a better result more robustly.

Cosine					
		Raw SA	Best Model		
		ρ	Basis	Assoc.f	ρ
fixed	10^{-2}	0.019	B_{5k}	logFr	0.309
	10^{-3}	-0.031	B_{5k}	plogFr	0.425
inv.	10^{-5}	0.135	B_{all}	logFr	0.446
prop.	10^{-6}	0.057	B_{1k}	logFr	**0.496**
	10^{-7}	-0.043	B_{1k}	logFr	0.401
Expected best combination			B_{1k}	logFr	

Table 7.8: Best combinations of filtering and association function for the tf-idf and PMI inspired functions, respectively, using cosine similarity.

Lin's Similarity					
		Raw SA	Best Model		
		ρ	Basis	Assoc.	ρ
	10^{-2}	0.345	B_{all}	tfidf-id	0.46
	10^{-3}	0.281	B_{all}	tfidf-id	0.403
inv.	10^{-5}	0.23	B_{100k}	tfidf-id	0.387
prop.	10^{-6}	0.234	B_{all}	**tfidf-id**	**0.491**
	10^{-7}	0.398	B_{all}	tfidf-id	0.467
Expected best combination			B_{100k}	tfidf-id	

Table 7.9: Best combinations of filtering and association function for the tfidf and PMI inspired functions, respectively, using Lin's similarity.

7.4.3 Final results

Table 7.10 shows the result of our model applied to both the remaining 302 word pairs and the whole dataset of 352 word pairs. It also includes a comparison to all network and corpus baselines. The performance drops to 0.401 from 0.491 on the whole dataset. The question arises whether the optimal settings and parameter we found in the previous section had been grossly overfitting the sample data and, if so, which part of the model did so. We hence again explored the parameter space around our 'optimal setting' fixing Lin's similarity and an inversely proportional threshold but stepping through all combinations of τ, B_i and association functions on the remaining 302 word pairs. Again, we got the same optimal settings as in Table 7.9[3], but the final results were consistently lower. In other words, the settings and parameter exploration is robust and generalises, but the larger dataset is inherently more difficult to rank than our sample set was.[4]

Results - Full Dataset		
	302 pairs	352 pairs
Network-based semantic space		
Threshold: $\tau = 10^{-6}$,inv.prop; B_{all}; A=tfidf-id; Lin's	0.402	0.401
Corpus-based semantic space		
Mitchell & Lapata's model*	0.291	0.304
Network Baselines		
1-hop Network Baseline: B_{100k}, A=tfidf-id, cosine	0.263	0.256
Simple Average Activation	0.252	0.249

Table 7.10: Best models established in Section 7.4 on the full WordSim353 dataset. (*) Reimplemented and run on same 1 million sentences corpus.

Despite this drop, our network-based semantic space model well outperforms all baselines. If we look back at the composition of the contexts the 'winning' model produces (cf. Table 7.1), we see that it contains about 14% 1-hop nodes, 81% 2-hop nodes and 5% nodes even further away. The 1-hop network-baseline illustrates the impact of these indirect connections. To give a fair evaluation, we picked the optimal model for this 1-hop context, which gives the maximum score that can be achieved by taking only these nodes into account. The 'simple average activation' (SAA) baseline is based on [Harrington, 2009]. It measures the relatedness of a word pair (a,b) by the average of the activation b receives when a is fired and vice versa. It can be derived from the raw value vectors by picking only the corresponding two values in the vectors v_a and v_b, i.e.

$$SAA(a,b) = \frac{v_a(b) + v_b(a)}{2}$$

[3]The only difference was that for the outlier $\tau = E^{-5}$ the best basis turned out to be B_{10k} instead of B_{100k}, with a negligible result difference of 0.004

[4]This may be due to the fact that the sample set was relatively well spread over the whole scale of relatedness, meaning the pairs were usually well distinguishable. Taking the whole datasets closes these gaps and finer grained distinctions need to be made.

The significant difference in result illustrates the gain generated by taking into account the whole local context instead of just one node.

Finally, and most crucially, our network-based semantic space model clearly outperforms the optimised corpus-based model, demonstrating the gain achieved from using the network structure over the text directly.

7.5 BNC - Scalability of Results

From Table 7.11 we can see that our network-based semantic space model continues to outperform the optimised corpus based vector space model by far as we scale to up to more than 75m words. On the largest corpus we achieve a score of 0.537 versus 0.403 for the corpus model. Moreover, although both semantic space models show the diminishing returns of additional data classical to statistical approaches, the network still achieves a healthy growth of +0.022 from 3m to 4m sentences while the corpus-based model is already down to negligible changes of +0.003. Hence, the network not only makes better use of the underlying text, it also does so for longer.
This is a powerful result which for the first time demonstrates that automatically generated networks achieve robust, scalable and significant improvements over established corpus approaches on large corpora.

Results - Corpus Sizes				
	Sentences			
	1m	2m	3m	4m
Network-based semantic space				
$\tau = 10^{-6}$,inv.prop; B_{all}; A=tfidf-id; Lin's	**0.401**	**0.476**	**0.515**	**0.537**
Corpus-based semantic space				
Mitchell & Lapata's model*	0.304	0.361	0.397	0.403
Network Baselines				
1-hop Network Baseline	0.256	0.331	0.337	0.324
Simple Avg. Target SA	0.249	0.213	0.217	0.264

Table 7.11: Results and baselines on different corpus sizes. (*) Reimplemented and run on respective corpus.

7.6 Wikipedia Networks

In the previous sections we studied term networks built from a plain text corpus. In this section, we discuss how the quality of our semantic relatedness measure is affected by the use of a concept network. For this purpose, we look at the corpus of all first sentences from Wikipedia and contrast the term network build from the plain version of the corpus with the concept network

constructed by integrating the manual disambiguation provided by hyperlinks (cf. Section 5.2).

Semantic Relatedness is in itself a measure between concepts and we would hence expect better results on the concept network. However, this projected advantage comes with the added difficulty of having to map the words in the word pair to nodes in the network. As our nodes correspond to concepts with an article in Wikipedia, this is equivalent to finding the appropriate articles which disambiguate the words in the word pair. This necessary mapping is inherent to all resource-based methods and we will briefly review the strategies found in related work. The final performance of a method is ultimately composed of both the quality of the mapping and the measure of relatedness in itself.

7.6.1 Concept Mapping

The appropriate mapping from words to Wikipedia articles (or network nodes for that matter) is not straightforward, neither for humans nor automatic systems. On the one hand, there can be a plethora of candidates. For example, the disambiguation page for KING has 65 entries including MONARCH, KING_(CHESS) and KING_(BAND). On the other hand, some senses may not have a corresponding article such as *round* as part of a fight or the most general meaning of *delay* or *arrival*. This is in particular true for words that refer to abstract concepts such as *importance*. This mismatch can often be attributed to the encyclopaedic nature of Wikipedia, which does not necessarily cover the most general meaning of common words. The task of mapping each word in a word pair to a Wikipedia article for semantic relatedness then adds the difficulty of picking the right sense with regard to the respective other word. For example, for the pair *jaguar-cat* we want to pick the articles JAGUAR-CAT, but for *jaguar-car* the articles JAGUAR_CARS-AUTOMOBILE would be appropriate.

In the literature on WordNet and Wikipedia-based approaches to measuring semantic similarity and relatedness a variety of strategies for dealing with this step automatically can be found. These fall into two broad categories. The first type aims to find an explicit mapping for each of the two words to an appropriate concept (i.e. a synset in WordNet and an article in Wikipedia). The approaches then differ in how elaborate they are and which information is taken into account. The naive approach simply uses the most frequent sense of each word. In the case of Wikipedia, this is defined through the *anchors*, the words that are hyperlinked to an article. The most frequent sense is the article that the anchor was most often hyperlinked to. However, the naive approach does not consider the context given by the other word in the pair, which can lead to crucial mistakes such as in the case of *jaguar-car* vs *jaguar-cat*. A popular choice is to first retrieve all or at least the most important candidate senses for both words, compute the relatedness of all combinations using the proposed semantic relatedness measure and then pick the highest scoring pair (e.g. Zesch and Gurevych [2010]). As this results in $n * m$ semantic relatedness computations, where n and m are the number of candidate senses for the first and second word, respectively, this type of automatic disambiguation can be very expensive, but gives very good results. One of the most refined mapping algorithms can be

found in Milne and Witten [2008]. They have a sequential decision process that takes into account both candidate relatedness and sense frequency. In addition, they refine the decision in order to deal with specific subtle cases such as word pairs that form noun compounds. They evaluate their agreed manual mapping, the naive most frequent sense mapping and this full sophisticated mapping algorithm and find that the latter not only gives as good results as the manual mapping, but even manages to outperform it. Their findings illustrate that the mapping step can be automated at no loss compared to a supervised mapping.

The second type of automatic disambiguation avoids a decision on an explicit mapping altogether and takes into account all candidate senses, weighted in some way by their probability. In the case of random walks or personalised page rank algorithms this is commonly done by distributing the initial probability mass between the candidate senses according to their level of commonness as a sense of the target word [Hughes and Ramage, 2007, Agirre et al., 2009]. In this way, all candidate senses influence the final distribution, to varying extents.

In our work, the core objective is to evaluate the exact effect of using a concept network in contrast to a term network, keeping all other conditions as similar as possible. Therefore, it is for us most appropriate to work with an explicit mapping strategy. We make use of two mappings: one is manually created, the other is a straight forward naive automatic mapping. The first allows us to isolate the gain through the difference in network structure. The second gives a lower bound on performance for an automatic mapping. We refrain from reimplementing the automatic mapping algorithm from Milne and Witten's work, but consider their findings sufficient to be confident the results we achieve with the manual mapping can be achieved with an automatic mapping.

Agreed Mapping (AG and AG^+) The mapping was produced by five annotators[5]. They were presented with the word pairs and asked to give the corresponding Wikipedia article which most closely matches the meaning of each of the words or put down 'n/a' if such a mapping did not exist. Initially, two annotators provided a mapping of all WordSim-353 pairs. Additionally, three annotators mapped a subset of 92 pairs that the previous annotators disagreed on or that at least one of the annotators marked as a low confidence mapping. The annotator agreement rate reflected the difficulties described above, with 25 out of the 92 difficult pairs not achieving an agreement at all (less than 3 annotators agreeing on at least one of the words in the pair). A further 32 pairs only were decided on with low agreement (exactly 3 agreeing annotators). Out of the 327 agreed pairs, 40 contained at least one word for which annotators felt there was no appropriate Wikipedia article. Overall, we have a set of 287 word pairs with an agreed explicit mapping to article concepts nodes in the network. We still wanted to arrive at a score for the whole dataset and hence created an automatic extension of the agreed mapping. For those 64 word pairs for which there either was no agreement by the annotators or the annotators agreed there was no appropriate mapping, we use the 'most common sense' as retrieved in the naive

[5]We thank Brian Harrington, Debora Field, Edward Grefenstette and Andrei Akhvlediani for their efforts here.

explicit mapping. We will refer to the two sets as the 287 pairs agreed (AG) set and the 352 pairs agreed plus automatically extended (AG^+).

Table 7.12 shows examples of word pairs for which annotators agreed. The top part lists word pairs for which both words could be mapped. We can clearly see the pairwise disambiguation and abstraction performed by the annotators. *Drinkware* is mapped to the generic DRINKWARE article in combination with *artifact*, but to COFFEE_CUP in combination with *coffee*. Similarily, *atmosphere* in the word pair *peace - atmosphere* is mapped to MOOD_(PSYCHOLOGY) rather than the Wikipedia entry ATMOSPHERE, which refers to earth's atmosphere. The second part of Table 7.12 lists words for which annotators agreed no mapping existed and the most common sense mappings included in AG^+. In most cases, the mapped article refers to a more specific sense of the word than appropriate in the word pair. For example, the article ARRANGEMENT refers specifically to music arrangements rather than a general action of arranging something. Table 7.13 gives examples of word pairs for which annotators could not agree. While *investigation* demonstrates the potential availability of too many appropriate senses, the other case highlight the different choices annotators made deciding when all available senses are too far from the perfect mapping.

Naive Explicit Mapping (NM) This automatic mapping uses a simple algorithm. It queries Wikipedia with the target word and returns the retrieved page if there is a direct hit. If it hits a disambiguation page, it picks the most common sense as defined above using the target word as an anchor. It was implemented using the JWPL[6] Wikipedia API.

For the two full mappings - AG^+ and NM - we assign to all pairs for which no mapping is found a score equal to the median of the other scores. This follows a smoothing rationale that if we do not have any data to compute an appropriate relatedness score, we are better off assuming some average relatedness rather than 0.

7.6.2 Results

Table 7.14 presents the results based on the Wikipedia network using the different mapping strategies. Based on a sample set, we found that it was best to slightly adjust the model from the BNC networks to a lower threshold value of $\tau = 10^{-7}$ instead of $\tau = 10^{-6}$ and to a restricted basis of B_{50k}.

The best result with a Spearman coefficient of $\rho = 0.65$ is achieved on the 287 pairs for which we had an agreed mapping (AG), but the difference to $\rho = 0.644$ for the automatically extended mapping is minimal. However, the result drops noticeably $\rho = 0.542$ when the naive mapping is used. This illustrates the extent of the loss which can occur through using an unsophisticated mapping. Nevertheless, whichever mapping is used, the full model always outperforms the

[6] `http://code.google.com/p/jwpl/`. We thank to Johanna Völker for providing a full setup of the Wikipedia database and code for accessing and querying the database.

Mappings with annotator agreement for both words ((AG) and AG$^+$)

Both words have a corresponding article

Word Pair		Mapped Articles (AG)	
brother	monk	Brother_(Catholic)	Monk
computer	keyboard	Computer	Computer_keyboard
peace	atmosphere	Peace	Mood_(psychology)
minority	peace	Minority_group	Peace
drinkware	coffee	Coffee_cup	Coffee
drinkware	artifact	Drinkware	Artifact_(archaeology)

At least one word does not have corresponding article (italic)

Word Pair		Mapped Articles (AG$^+$)	
reason	*criterion*	Reason_(argument)	*Criterion_validity*
arrangement	*accommodation*	*Arrangement*	Dwelling
listing	proximity	*Johann_Benedict_Listing*	Distance
hotel	*reservation*	Hotel	*Indian_reservation*
announcement	news	*Announcement_computing*	News

Table 7.12: Sample word pairs with annotator agreement (at least three out of five annotators agreed). In those cases for which annotators agreed that there is no mapping, the automatically extended set (AG$^+$) uses the most common sense.

Word pairs without annotator agreement for at least one word

Word Pair		Annotator Suggestions		Mapped Articles (AG$^+$)	
investigation	*effort*	Criminal_procedure Clinical_trial Discovery_(observation) Scientific_Method	Energy_(psychology) Manual_labour n/a	*Private_investigator*	*Energy*
type	*kind*	Type-token_distinction Type_theory n/a	Kind_(type_theory) n/a Natural_kind	*Type_(metaphysics)*	*Kind_(type_theory)*
focus	life	n/a Purpose Focus_(optics) Attention	Life Personal_life	*Focus_(optics)*	Life

Table 7.13: Sample word pairs for which annotators could not agree on a mapping for at least one of the two words. AG$^+$ selects the most common sense.

Wikipedia results			
	287 Pairs (AG)	352 Pairs (AG^+)	352 Pairs (NM)
Wikipedia network	0.65	**0.644**	0.542
1-hop Baseline	0.617	0.556	0.449
Plain Wikipedia Network	-	0.454	
3m BNC Network	-	0.515	

Table 7.14: Wikipedia-based results using different mapping strategies and comparison with other reference networks.

1-hop baseline showing that we gain from indirect connections in these concept networks, too. The effect is considerably smaller than it was for the BNC networks though, if the mapping is of high quality. For the 287 agreed pairs, the 1-hop baseline gives $\rho = 0.617$ compared to $\rho = 0.65$ for the full model. In contrast, for the 3 million sentences BNC network, the results were $\rho = 0.337$ and $\rho = 0.515$, respectively. This larger improvement over the baseline for the BNC network than for the Wikipedia one could be due to the higher density of the first. The denser and more interconnected the network, the larger distinctions generated by the spreading activation.

Independently of the quality of the mapping, we can see a significant gain through the use of the Wikipedia *concept* network over the plain Wikipedia *term* network. The difference between the plain Wikipedia network performance and the Wikipedia network with 'optimal' AG^+ mapping (0.454 versus 0.644) isolates the real effect of term versus concept network. The NM mapping demonstrates the gain is even significant with suboptimal mappings. As we reviewed in the last section, previous related work has detailed algorithms for constructing mappings of manual quality automatically. We conclude that our observations hold independently of the mapping and use the AG^+ mapping in the following sections, knowing that the results can not only be achieved but potentially even be outperformed with a full automatic mapping algorithm.

7.7 Combining BNC and Wikipedia

The two most successful approaches to semantic relatedness are based on supervised combinations of one or more methods. Agirre et al. [2009] trains support vector machines to combine the ranks produced by two different WordNet-based methods and a web-corpus based vector space model. Haralambous and Klyuev [2011] combines a Wikipedia, a WordNet and a corpus model. In the same spirit, this section integrates the results from the plain 4 million sentences BNC network with those from the Wikipedia concept network. However, we use a simple *unsupervised* combination with equal 50-50 weight for the BNC and Wikipedia.

Table 7.15 shows that even with the naive weighting, the combination is considerably better than either of the two networks individually with a Spearman coefficient of $\rho = 0.70$ versus

0.537 and 0.644 for the BNC and Wikipedia, respectively. In order to get an idea of the ceiling such an integration can yield, we use an 'oracle' following [Agirre et al., 2009]. It knows for each pair whether the BNC or the Wikipedia networks gives a better result. With a score of $\rho = 0.854$, this upper boundary reveals that the potential of the integration is by far higher than what we accomplish with simple averaging.

BNC & Wikipedia combination				
	BNC (4 million)	Wikipedia	combination	combination ceiling
352 Pairs (AG^+)	0.537	0.644	**0.70**	0.854

Table 7.15: Results of combing the 4m BNC and the Wikipedia network ranks.

These observations have important implications for the application of the networks in a real-world scenario. The Wikipedia concept networks give a higher quality representation than plain networks, but Wikipedia is a limited resource. Nevertheless, we can build a Wikipedia network as a starting point and through integration with a plain network from any additional corpus significantly enhance the results. This is a promising road map for domain-specific applications.

7.8 Comparison with Related Approaches to Semantic Relatedness

In Section 7.1.1 we grouped related work into corpus-based, knowledge resource-based and hybrid approaches which use a supervised combination of one or more methods. Table 7.16 lists the best techniques from each group as reported on the WordSim353 dataset. We will address each set separately.

In the group of corpus-based vector space models, Finkelstein et al. [2002] and Gabrilovich and Markovitch [2007] use word-document models on encyclopedic articles while Mitchell and Lapata [2008] and Agirre et al. [2009] adopt word-word models on general corpora. The first two techniques leverage the division of the corpus into articles, each of which represents a distinct concept. Finkelstein et al. [2002] reports results based on the original model presented in Landauer and Dumais [1997], which employs singular value decomposition to detect latent topic structure in the word-document matrix. Gabrilovich and Markovitch [2007] work without this step and represent a word directly by the Wikipedia articles it occurs in.

In contrast, Mitchell and Lapata [2008] and Agirre et al. [2009] use unstructured corpora and represent a word though its co-occurrence with other words within a context. They are hence directly comparable to our method and constitute the core point of reference for evaluating the gain accomplished by using the network structure instead of the text directly as basis for the semantic space model. Mitchell and Lapata [2008] is the optimised model which we used

Comparison with related approaches on WordSim353		
Single resource or method		ρ
Distributional, corpus-based		
Mitchell and Lapata [2008]	British National Corpus (100 million words)	0.42
Agirre et al. [2009] (A-VS1)	Web corpus (1 billion words)	∼0.52
Finkelstein et al. [2002]	Encyclopedic articles (4.6 million words)	0.56
Agirre et al. [2009] (A-VS2)	Web corpus (1.6 trillion words)	0.66
Gabrilovich and Markovitch [2007]	Wikipedia articles	0.75
Automatically Generated Semantic Network		
Harrington [2010]	ASKNet networks (4.7 million words)	0.62
WordNet		
Hughes and Ramage [2007]	WordNet graph (Random Walk)	0.55
Agirre et al. [2009] (A-WN1)	WordNet graph (Person. PageRank)	0.56
Agirre et al. [2009] (A-WN2)	WordNet graph incl. disamb. glosses	0.66
Wikipedia Structure		
Strube and Ponzetto [2006]	Category Structure	0.48
Yeh et al. [2009]	Graph from article links, categories and infoboxes (Random Walk)	0.49
Milne and Witten [2008]	In- & out going article links (Vectors)	0.69
Gouws et al. [2010]	Graph from article links (Spreading Activation)	0.70
Yazdani and Popescu-Belis [2010]	Graph from article links and content (Person. PageRank)	0.71
Multiple resources or methods (supervised combination)		
Agirre et al. [2009]	WordNet(+glosses)+Webcorpus	0.78
Haralambous and Klyuev [2011]	ESA+WordNet+GoogleBook corpus	0.86
Our networks		
4m sentences BNC network	British National Corpus (75 million words)	0.537
Wikipedia network (AG^+)	Wikipedia first sentences (50 million words)	0.644
4m BNC & Wikipedia (AG^+)	unsupervised combination	0.70

Table 7.16: Comparison of our results to previous approaches on the WordSim-353 dataset.

as a reference in the previous sections through a reimplementation. Our BNC network from 4 million sentences/75 million words comfortably outperforms their approach even on the full 100 million words. The results reported in Agirre et al. [2009] show how the quality of this corpus-based semantic relatedness measure develops on web-scale corpora. Their smallest corpus contains 1 billion words and is 10 times larger than the BNC (cf.A-VS1). Our 4 million sentences BNC network still outperforms this model, suggesting that we extract as much or more valuable information from text through the network structure than a standard model does from 10 times more text. On 10,000 times more text (1.6 trillion words) the result increases to 0.66 (cf.A-VS1). However, this is only slightly higher than our result using the Wikipedia concept network, illustrating the power of high quality networks. Furthermore, it is well outperformed by our simple combination of the BNC and the Wikipedia network with a result of $\rho = 0.70$.

In Harrington [2009] and Harrington [2010], Harrington reports a maximal score of $\rho = 0.62$ on his automatically generated concept networks, based on the simple average activation measure we reimplemented as a reference in previous sections. This is a high score given the small corpus of 4.7m words and it is considerably higher than the top result of 0.26 our reference measure gave on the 4 million sentences BNC network. However, a direct comparison between his and our results is problematic due to the different nature of the corpora. Harrington queries a search engine and the top five pages for each unique word in the WordSim353 dataset, leading to a corpus tailored to the test set. His own PMI-based baseline gives considerably better results on this corpus than on the BNC, underlining the specific quality of the corpus. While we need to take into account the disambiguation and differences in the settings of the spreading activation algorithm, it appears the results may not transfer to the large general corpus we used. Unfortunately, we cannot conduct a more direct comparison since the implementation of ASKNet does not scale to our corpus (cf. Section 4.2) and the exact spreading activation settings underlying the experiments are not available.

The best WordNet based approaches build a graph structure from the synsets and relations in the lexical database. Hughes and Ramage [2007] and A-WN1 make use of synsets and taxonomic relationships only, A-WN2 also includes relationships from the disambiguated glosses. The latter constitutes a high quality semantic network based directly on a manually established taxonomy and disambiguated definitions. Nevertheless, our Wikipedia networks give competitive results indicating that automatically generated concept networks provide a viable alternative to manually created resources.

In the group of Wikipedia approaches all but Milne and Witten [2008] construct some form of graph from the various types of hyperlinks available. Gouws et al. [2010] use hyperlinks in the article body text only, resulting in a directed graph of articles with edges A$\rightarrow$ B if there is a hyperlink to B in article. Yeh et al. [2009] and Yazdani and Popescu-Belis [2010] in addition include links or information from info boxes and the category structure, both of which give

higher quality typed links between articles than the somewhat loose association provided by untyped links in the body of the article text. Since we also form a network based on article body text links only and use spreading activation, Gouws et al. [2010] is most related to our work. In theory, the comparison between their method and ours could show whether our grounding of connections in deep linguistic analysis provides a gain over a graph from untyped links. However, the data we have is inconclusive because Gouws et al. [2010] use all full articles to form their network while ours is based solely on first sentences.

Finally, we complete the picture with the two techniques that use a supervised combination of two or more resources (Agirre et al. [2009], Haralambous and Klyuev [2011]). They are the best performing systems on the WordSim353 dataset, but are exposed to over fitting through training and testing on the same dataset.

7.9 Summary

In this chapter, we constructed a network-based semantic space model, in analogy to widely applied corpus-based vector space models. It makes use of spreading activation to select the relevant context of a target node and to induce raw values for each node in this context. The spreading activation pattern is translated into a vector and further refined through basis filtering, the use of an association function and the selection of an appropriate similarity measurement. We conducted an extensive experimental exploration of a variety of settings for all aspects of the model and established that the best model made considerable use of indirect relations.

We found that through leveraging the network structure our model outperformed the optimised corpus-based reference model by far. This advantage proved to be independent of network size and the margin the network achieved increased with larger networks. Although we also observed diminishing returns of added data, results flattened less than for our reference model. A final comparison with the results reported by a corpus-based model on a web-scale corpus suggest that the gain achieved through the network structure is comparable to using a corpus one order of magnitude larger with a standard model.

Results were yet significantly improved through using the Wikipedia concept network, illustrating the boost created by this additional level of semantics. We were able to achieve results competitive to the best WordNet-based approach, which uses both the taxonomic structure and disambiguated glosses. This demonstrates that automatically generated concept networks can provide a viable alternative to manual resources.

Finally, we showed that through combining the BNC and Wikipedia networks in an unsupervised way, we could get better results than through either of them individually. This opens the possibility to have one core Wikipedia-based concept network which can be complemented

as needed by on-the-fly generation of a simple term network from plain text. The simple combination achieved results competitive to state-of-the art unsupervised models.

Chapter 8

Network-based Unsupervised Noun Compound Paraphrasing

Noun compounds (NCs) such as *apple pie* are both very frequent and very productive. Baldwin and Tanaka [2004] found that 2.6% of all tokens in the BNC are noun compounds, amounting to a total of 265k tokens. At the same time, their frequencies follow a power law distribution, with more than 60% of all noun compounds occurring only once, illustrating their productivity. It is then crucial for NLP applications to be able to deal with noun compounds well, which often means understanding the relation between the two nouns, i.e. that *apple pie* is a 'pie made with apples'. For example, a Textual Entailment system that cannot decode noun compound relations may draw the wrong conclusion in the following case (taken from Nakov [2008a]).

> T: As McMillan collected, she also quietly gave, donating millions of dollars to create scholarships and fellowships for [...] *professors* **who study** *women* in the developing world.
> H: As McMillan collected, she also quietly gave, donating millions of dollars to create scholarships and fellowships for [...] *women professors* in the developing world.
> A: NO

Similarly, Machine Translation systems can benefit from augmented parallel corpora and search engines can increase recall by using the paraphrases in query expansion [Nakov, 2008b].

We can again either try to manually encode the desired relations in a lexical knowledge resource, or automatically retrieve them from corpora using a statistical approach. However, the productivity of noun compounds as well as the implicitness of the relation create challenges for both strategies. Only the most frequent compounds can be included in a lexical resource. According to Baldwin and Tanaka [2004], even of the relatively frequent compounds with 10 or more tokens in the BNC, only 27% were found in a dictionary. In addition, general purpose lexical resources often do not cover the relations necessary for noun compound description. WordNet, to take

the most prominent example, contains primarily hierarchical relations between concepts. If, on the other hand, the relations are to be derived from corpora, the prevalent productivity means sparseness is a major issue. In addition, the fact that the relation between compound parts is usually implicit and understood by humans from the context means that explicit statements of the relations are rare.

Noun compound paraphrasing is an interesting application for us for several reasons. The semantic relatedness task primarily allowed us to evaluate the overall structure of the network and the extent to which we can derive reliable *quantitative* predictions about the relation between two concepts. Here, we can investigate whether direct and indirect paths in the network are suited as *qualitative* descriptions of the exact relation between concepts. Noun compound relations in particular make a good case for establishing the semantic networks as a hybrid between lexical resources and statistical approaches. On the one hand, we can show that they are more flexible and more widely applicable than manually created lexical resources. Not only can we work with the necessary breadth of relations, we can also do so for any domain and can enhance coverage simply by adding relevant text. On the other hand, we can majorly reduce sparseness issues faced by the statistical approaches, meaning we can get good results on relatively small corpora.

We tackle the SemEval2010 task on noun compound interpretation by paraphrasing [Butnariu et al., 2009], which is concerned with ranking a given set of paraphrases according to their suitability for describing the underlying relation. Our approach is very naive and just uses the raw frequency counts of the paraphrase on paths of a certain maximum length between the compound parts. Nevertheless, it is sufficient to illustrate the substantial gain over using the text directly, to outperform previous unsupervised approaches and to yield results competitive to the best performing web-scale system.

The chapter is structured as follows. Section 8.1 introduces the noun compound paraphrasing, introduces the evaluation setting and gives an overview over related work. Section 8.2 describes our path-based model for paraphrase ranking. Section 8.3 presents and analyses the results on the plain text term networks, investigating the effect of corpus size and type. Section 8.4 looks at the contrast between term and concept networks by the means of the term and concept Wikipedia-based networks. Section 8.5 puts our results into perspective by drawing a comparison with related work and Section 8.6 summarises our findings.

8.1 Noun Compound Paraphrasing

Traditionally, the view taken by some theoretical linguists is that noun compounds can be described by a fixed set of relations. For example, Lev [1978] uses 12 relations including CAUSE$_1$ (as in *tear gas*) or HAVE$_1$ (as in *apple pie*). This restriction to a fixed set is neat and paves the way for a standardised lexical resource covering noun compound relations. However, the pro-

ductivity of noun compounds in language means manual acquisition cannot scale and relations need to be generated automatically from text. In attempts to do so, verb-centred paraphrases have been shown to be a good source for learning fixed relations. Nakov and Hearst [2006] use paraphrases extracted from the web to predict Levi's relationships, Butnariu and Veale [2008] use a the Google *n-gram* corpus and web-validation to match a more fine-grained fixed relationship set.

However, the work on the use of paraphrases for modelling noun compound semantics has also given rise to the idea of using the paraphrases *directly* in applications. Nakov [2008a] advocates the argument that a weighted set of possible paraphrases is a better and more usable description than a specific relation from a fixed set. He argues that the abstract relations underspecify the semantics and cannot directly be used. For example, in the case of *malaria mosquito*, the fine-grained list of weighted paraphrases *carry(23), spread(16), cause(12), transmit(9), etc.* gives a more appropriate picture of the relation than CAUSE_1. Direct application of paraphrases in NLP applications has been successful in improving machine translation [Nakov, 2008b, Baldwin and Tanaka, 2004] and appears to have potential in recognising textual entailment Nakov [2008a]. Due to these developments, a new SemEval task was introduced in 2010 specifically concerned with the interpretation of noun compounds using paraphrasing verbs and prepositions.

8.1.1 Task and Evaluation

The focus of the SemEval 2010 Task "The Interpretation of Noun Compounds Using Paraphrasing Verbs and Prepositions" [Butnariu et al., 2009] is on creating a ranking of a set of given paraphrases and not generation of the paraphrases as such. The rationale is that paraphrases can be generated rather easily from corpora, while the decision on the most central ones is the difficult part. The dataset contains 250 training and 388 test noun compounds. It covers lexicalised compounds such as *town hall*, non-lexicalised ones such as *january temperature* and infrequent words, such as *absorption hygrometer*. For each of them an average number of 71 human annotators were asked to provide three paraphrases consisting of a verb or a verb plus prepositions. The following shows an excerpt of the dataset for the noun compounds *afternoon rain* and *sea urchins*. The paraphrases are sorted by the frequency with which they were given by the annotators.

afternoon rain	occur in	20	sea urchins	live in	47
afternoon rain	come in	19	sea urchins	be found in	34
afternoon rain	fall in	19	sea urchins	be in	11
afternoon rain	happen in	16	sea urchins	dwell in	10
afternoon rain	be found in	7	sea urchins	reside in	10
	..			...	
afternoon rain	be predicted for	1	sea urchins	be bred in	1
afternoon rain	be produced in	1	sea urchins	be found throughout	1
afternoon rain	be related with	1	sea urchins	be housed in	1
afternoon rain	come down during	1	sea urchins	flourish in	1

The aim for systems is to recreate the ranking, given the possible paraphrases. The produced ranking is compared to the gold standard ranking for each noun compound individually using the Spearman correlation coefficient ρ. The final score for the system is the average Spearman coefficient over all compounds. We only consider the 388 test pairs for our evaluation, as our approach is unsupervised.

8.1.2 Approaches

Approaches to paraphrase ranking are best described by their level of supervision and the data sources used. An overview is given in Table 8.1. The supervised systems outperformed the unsupervised ones by some margin, with the exception of UCD-PN by the University College Dublin, which achieves competitive results to the supervised systems. Out of the supervised approaches, Wijaya and Gianfortoni [2011] performs best despite not using any additional external resources. To establish the ranking for a noun compound, they first retrieve the k nearest neighbour noun compounds from both the training and test set by employing a weighted distributional similarity measurement of the set of paraphrases given for the compounds. From these nearest neighbours they retrieve conditional probabilities of a paraphrase occurring for a noun compound given another of its paraphrases. A combination of the individual conditional probabilities leads to the final ranking.[1] The UVT-MEPHISTO system by the University of Tilburg trains a classifier to decide for each paraphrase pair which one should be ranked higher and produces a final ranking for a noun compound by combining the pairwise scores. It has an extensive feature set that incorporates the training data, WordNet and frequency counts from the Google n-gram corpus. The second system by the University College Dublin, UCD-GOOGLE-III, also makes use of both the training data and external sources. It combines two supervised methods based on the training data and one unsupervised method based on both the Google n-gram corpus and WordNet in a hybrid system. The supervised part trains preferences between paraphrases similar to UVT-MEPHISTO. The unsupervised part looks at verb-argument frequencies in the Google n-gram corpus, boosted by WordNet. We will describe this unsupervised subpart in more detail in our further discussion and list it separately as UCD-GOOGLE-0.

As our system is unsupervised, these systems are of more importance to our investigations. All but UCD-PN provide valuable comparisons to our work as they are in some form or another based on frequency observations on a corpus. In particular, the UCAM system by the University of Cambridge is based on the BNC and Miklosch [2011] uses all first sentences from Wikipedia and an associated hyperlink graph, both allowing for an immediate comparison of results. UCD-GOOGLE-0 and NC-INTERPRETER use web-scale data, illustrating the effect of 'unlimited' data. While UCD-PN is the best-performing unsupervised system, its status is somewhat peculiar. While it does not use the training data and is hence not supervised in a narrow sense, it re-

[1] The score of the system needs to be considered with caution, as it relies on the test set itself to retrieve information. If the test set was reduced to only one noun compound, it is hence likely to achieve a worse result than the same sample gets when other samples are being tested, too.

Approaches			
	supervised	SemEval paraphrase data	external data
Wijaya and Gianfortoni [2011]	yes	yes	-
UVT-MEPHISTO Wubben [2010]	yes	yes	Google n-gram, WordNet
UCD-GOOGLE-III [Li and Lopez-Fernandez, 2010]	hybrid	yes	Google n-gram, WordNet
UCD-PN [Nulty, 2010]	no*	yes	-
UCD-GOOGLE-0	no	no	Google n-gram, WordNet
UCAM	no	no	BNC
Miklosch [2011]	no	no	Wikipedia
NC-INTERPRETER	no	no	Web snippets, WordNet

Table 8.1: Overview of approaches to noun compound paraphrase ranking.

lies on *all the test data* to score a test sample. From the sets of human provided paraphrases for all test samples it estimates conditional probabilities of a paraphrase occurring for a noun compound given another of its paraphrases. Its performance hence depends on the size of the test data. This in its full consequence means that if it was provided only with a single noun compound to produce a paraphrase ranking for, it could not do so - a problem not faced by the other systems that can score any number of noun compounds independently. To solve this, the system would need to learn the probabilities from the training data, making it a supervised system. We hence exclude it from the group of unsupervised systems.

The unsupervised subsystem UCD-GOOGLE-0 uses patterns such as *Head VP DET Mod* to extract <*head, paraphrase, modifier*> triples from the Google n-gram corpus. In order to deal with data sparseness, these triples are split into partial triples, <*head, paraphrase, **> and <*, *paraphrase, modifier*>. Furthermore, the triples set is extended through an inference layer that adds alternative heads and modifiers provided they are sufficiently similar to the original head using a WordNet-based similarity measurement. From the frequency triple counts, the conditional probabilities of a paraphrase given a noun compound are estimated. The suggested paraphrases for a test noun compound are then ranked by these conditional probabilities. UCAM and NC-INTERPRETER both use verb-argument frequencies more directly, on the BNC and Web snippets, respectively.

Similarly to us, Miklosch [2011] uses a Wikipedia-based graph built from all first sentences of Wikipedia. Their graph model is directed and has an edge $s \rightarrow t$ if there is a hyperlink to article t in the first sentence of the article s. The paraphrases for a noun compound are ranked in two steps. First, paths are retrieved between the corresponding articles of the two nouns in the graph. Secondly, the number of occurrences of the paraphrases in these sentences is counted. The paraphrases are then ranked according to the frequencies with additional refinements. A plethora of settings for link weightings in the graph, path extraction and path filtering methods

alongside several settings for the refinement of frequencies and incorporations of special nodes on the graph are evaluated against each other, based on the test set. This gives a comprehensive view on what can be achieved with this method on the same corpus that we work with.

8.2 Path-based Noun Compound Paraphrasing

Our approach is very simple. For a given noun compound, we extract paths of a certain maximum length between the nodes corresponding to the compound paths. The paraphrases are then scored according the number of times they occurred on the paths. By varying the maximum length, we control the extent to which indirect paths are taken into account. Our hypothesis is that we make better use of the underlying corpus. We reduce sparseness by utilising indirect relationships while ensuring high quality relations through the underlying linguistic analysis. For example, in Figure 8.1, the direct connection between *apple* and *cake* only gives the paraphrase *contain*. However, by taking into account paths over the concepts *fruit* and *banana*, we also find *baked with* and *made from*.

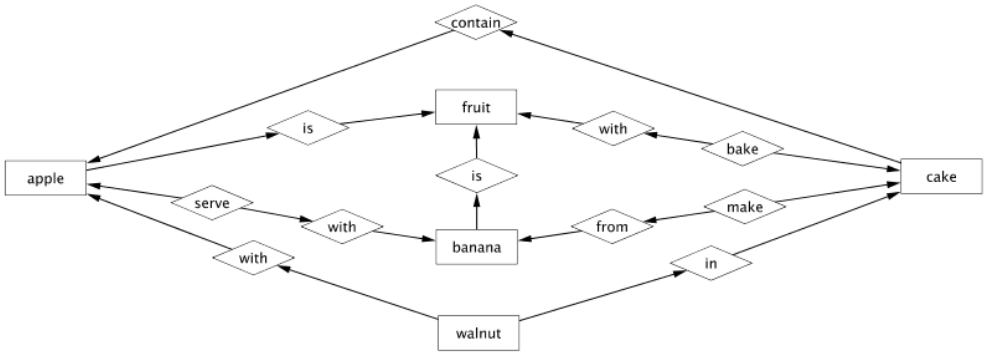


Figure 8.1: Illustration of some paths relevant to noun compound paraphrasing of *apple cake*.

We need to specify what we mean by path and length. When we speak of a path, we are referring to a path in the full network model such as '*apple*-serve-with-*banana*-from-make-*cake*'. If the relation 'make from' between *cake* and *banana* occurs twice or if there is another relation between them, these would count as different full paths. However, we want to define *length*, in terms of hops from concept node to concept node and hence also look at the *skeleton path*. The concept-nodes-only skeleton path *apple-banana-cake* has a length of two and the direct skeleton path *apple-cake* over 'contain' has a length of one. Thus, one skeleton path will usually correspond to several full paths.

In terms of implementation, the path retrieval consists of in two steps. We first use the skeleton graph (cf. Section 6.2.1) to extract the skeleton paths of the chosen maximum length between the two nouns in question. In the above example and using a maximum length of 3, this step

gives us five skeleton paths between *apple* and *cake* - one of length 1 (*apple-cake*), three of length 2 (over *fruit*, *banana* and *walnut*) and one of length 3 (*apple-banana-fruit-cake*. In a second step, we retrieve all full descriptions of each relevant edge, such as 'bake-with' for the edge *fruit-cake*. Multiple occurrences of the same description for an edge are kept.

For each of the noun compounds paraphrases given in the dataset, we count the number of full paths that match the paraphrase. We require each part of the paraphrase to occur in order and as adjacent nodes on the path, but we ignore both position and direction. Lemmatised versions of the paraphrases are used since the labels of the network nodes are lemmas themselves. Given the paraphrase 'baked with' for 'apple cake', we look for retrieved paths which contain two adjacent relational nodes 'bake' and 'with', while it is irrelevant which one occurs first in direction of the path. In the above sample network, two such paths exist - one over *apple-fruit-cake* and *apple-banana-fruit-cake*. Finally, we rank paraphrases according to their path count.

While the task we tackle here is only concerned with ranking of paraphrases, we could also *generate* potential paraphrases automatically by retrieving all subpaths which match a certain pattern, such as two adjacent relational nodes with one verb and one preposition. This method would potentially lead to a very large number of suggested paraphrases, which would need to be sorted by their relevancy for describing the underlying relationship. In the pure ranking task, relevancy is pre-defined by the fact that paraphrases are given by the annotators and frequency is a reasonably sufficient measure of quality. In the generation task, however, we may retrieve frequent verbs on the path which carry no descriptive value of the relationship between the nouns. Hence, amending the algorithm to provide paraphrase generation would need a more refined weighting which takes into account global paraphrase frequency.

8.3 Plain Networks

In this section we present the results on the plain networks from both the BNC and the plain version of the Wikipedia corpus. We start off with a detailed analysis of the performance on the 1 million sentences BNC network and then see in the following section how the performance is influenced by network growth. We finish with a comparison between the Wikipedia corpus and the BNC to illustrate the sensitivity of the results to corpus quality and type.

8.3.1 BNC Networks

8.3.1.1 1 Million network

For our first experiments we conservatively restricted the skeleton path length to be maximally two, allowing for one intermediate concept node between the two target nodes. This follows the rationale that we expect shorter paths to be more valuable for describing the underlying relation between the compound parts. For the corpus baseline, the paraphrases for a noun com-

pound are ranked according to their frequency counts in sentences that contain both compound parts. In order to achieve fair levels of recall, we account for plural versions of the nouns when extracting the sentences and use lemmatised versions of both the sentences and paraphrases for the paraphrase matching.[2]

Table 8.2 gives an overview over the results as well as coverage and recall statistics for both the network-based approach and the corpus baseline. The results are given as the average Spearman correlation coefficient, the official evaluation measure of the task. We will use the notation $\overline{\rho}$ when we refer to this average and the standard ρ when we look at the score for a single noun compound. The network convincingly beats the baseline with $\overline{\rho}$=0.28 versus $\overline{\rho}$=0.15 on the respective set of covered noun compounds. At the same time, it achieves a higher coverage of noun compounds with 376 versus 334 out of the total 388 samples in the test set. 328 compounds are covered by both the corpus baseline and the network. There are 6 compounds for which the network does not retrieve paths despite the two nouns co-occurring in sentences. These are cases for which the fine-grained distinctions created by the linguistic analysis tools show that none of those sentences contain a valid relationship and that there are not any available over a further intermediate node either. On the other hand, there are 46 compounds which are covered by the network but not by the baseline. For these cases, the two compound parts were never mentioned together in the text and the paraphrase ranking established by the network purely relies on indirect relationships over an intermediate concept node. Nevertheless, when restricted to the 328 compounds also covered by the baseline, the score is only marginally better, meaning the purely indirectly derived rankings are only at negligible disadvantage.

Overall, the network approach achieves both an increase in precision through the linguistic foundation as well as an increase in paraphrase recall through the indirect paths. For example, the two nouns *rotation* and *period* co-occurred in four sentences. The corpus baseline approach wrongly finds the paraphrase 'depend' in the sentence "[..] determination of the rotation age is also crucial and depends on the species and the time period required for optimum growth", alongside several other incorrect ones. On this compound it retrieves 8.3% of the paraphrases and scores a negative Spearman of $\rho = -0.007$. The network approach does not match these paraphrases as they do not constitute a relation between the two nouns. It consequently retrieves only 1.8% of the paraphrases within a path length of 1, while scoring $\rho = 0.21$. Using the length of 2 the indirect connections increase the paraphrase recall to 56.48% and the score to $\rho = 0.286$.

Analysing the performance of the approach on the individual compounds, we found a significant correlation between the recall of paraphrases of the compound and its final Spearman score. The connection was even stronger when we considered only the recall of the frequent paraphrases, eg. those which had been given by more than one annotator, yielding a correlation coefficient

[2]For the lemmatisation we make use of `morpha`, developed by Kevin Humphreys, John Carroll, and Guido Minnen from the University of Sussex

1 million sentences based results					
		NCs	Avg. paraphrase recall		Avg. number
	$\overline{\rho}$	covered	all	frequent	of skeleton paths
Network	0.28	374 (96.4%)	40.23%	53.61%	313.52
Network (328 NCs)	0.283	n/a	43.40%	57.24%	353.18
Baseline	0.15	334 (86.1%)	13.03%	18.33%	-

Table 8.2: Average Spearman results ($\overline{\rho}$) and statistics of the network-based approach with maximum length two and the corpus baseline on the 1 million sentences BNC corpus.

of r=0.60 between frequent paraphrase recall and Spearman score. The paraphrase recall can only be influenced indirectly through the number of paths that are retrieved and there is a high correlation between the recall of frequent paraphrases and the number of retrieved skeleton paths ($r = 0.72$). Interestingly, the correlation between the frequent paraphrase recall and the number of *full network paths* is lower, which at least indicates that newly added connections may be more influential than strengthened connections when the network grows.

In terms of skeleton paths, noun compounds with very few paths indeed caused many of the lowest scores. For example, the compounds with less than 10 retrieved skeleton paths have an average Spearman of only $\overline{\rho} = 0.17$ against $\overline{\rho} = 0.286$ for those with at least 10 paths. However, a small number of paths alone does not necessarily cause issues. The noun compounds *metallurgy industry* and *impeachment trial* both have five skeleton paths and similar paraphrases with 15.8% versus 10.4%, respectively. However, *impeachment trial* reaches a score of $\rho = 0.275$ while *metallurgy industry* achieves a low $\rho = 0.076$. An indication of the reason for this difference lies in the actual paths that were retrieved. For *metallurgy industry* the intermediate nodes were *development*, *chemical*, *centre*, *beginning* and *example*, while those for *impeachment trial* were *court*, *place*, *parliament*, *proceedings* and *question*. In the first case, only one of the intermediate nodes is topically related to the compound while the other four are rather general concepts. In the second case, it is the other way round with four out of five having an obvious semantic connection to the compound. It is a weakness of our simplistic model which treats paraphrases from any path equally and does not distinguish between those that go through 'common' versus more 'specific' nodes. A more refined model could filter out or weigh paraphrases according to the commonness of the intermediate nodes.

For the second experiment, we wanted to see if an extension to paths of length 3 can help improve the results. In general, we would expect paths to be less descriptive of the relationship between the two parts the longer they are and that including them means a more sophisticated model with weighting is necessary. Nevertheless, we added a fallback strategy to rerun the algorithm with maximum length 3 for the noun compounds for which no or only a few skeleton paths were found. We rather randomly set the number of paths that we consider to be "too few" to be less than 10. The runtime for extracting all such paths proved infeasibly long, so we only worked with the first 50 paths extracted per noun compound. While this is not ideal,

it was sufficient for investigating if longer paths can be useful even in our simple model.

Table 8.3 shows the results of the fallback strategy. The average Spearman coefficient on the 31 noun compounds with less than 10 skeleton improved to $\overline{\rho} = 0.234$ from $\overline{\rho} = 0.17$ using only length 2. While this is still lower than the overall average of the length 2 algorithm on all noun compounds, it in particular helped with the originally worst cases. The five originally lowest scoring saw an improvement of +0.197 on average. With the fallback, six out of the twelve noun compounds not previously covered could be included, with surprisingly good results of an average of $\overline{\rho} = 0.294$. The improvement gained by this for the overall score on all pairs is only marginal because the fallback only affected less than 10% of the test set. However, it means that both higher coverage and better quality were achieved at the same time. We can conclude that the consideration of longer paths in the whole model, potentially with weighting, would be a fruitful path for future work.

Fallback Strategy		
	NCs affected or coverage	$\overline{\rho}$
NCs with less than 10 length 2 paths	31	0.234
NCs without length 2 paths	6/12	0.294
Full fallback model	380 (97.9%)	0.282

Table 8.3: Results for the fallback strategy to length 3 paths for noun compound with few (<10) or no retrieved paths at length 2.

8.3.1.2 Effect of network growth

With the growth of the network we found that both the coverage and the performance of the algorithm increased. On the three million sentences network, the length 2 algorithm alone achieved an average Spearman coefficient of $\overline{\rho} = 0.311$ with a coverage of 99.5%, corresponding to 386 out of 388 test noun compounds (cf. Table 8.4). The remaining two compounds contained a word that had no corresponding node in the network. As before, the corpus baseline stays behind with a coverage of 93.6% and a score of $\overline{\rho} = 0.169$.

A further core strength becomes apparent when not only the mean of the Spearman coefficients of the noun compounds is considered, but the whole distribution. Figure 8.2 shows the histograms of the distribution of the Spearman coefficients for the one million sentences network and for the corresponding corpus baseline algorithm. The plot for the network-based algorithm resembles a normal distribution and it has a sample standard deviation of $\sigma = 0.117$. In contrast, the histogram for the corpus baseline algorithm is not most naturally fitted by a normal distribution and at the same time has a higher standard deviation of $\sigma = 0.128$. Hence, the network approach not only achieves better results, it also does so more consistently. This contrast increases with network growth as the standard deviation *decreases* continuously to

$\sigma = 0.106$ for the network at four million sentences, but *increases* to $\sigma = 0.130$ for the corpus algorithm.[3] We can conclude that a larger corpus increases the results as well as the robustness of the results in the case of the network-based approach while it leads to higher variance in results for the corpus baseline.

Effect of Network Growth				
	Number of Sentences			
	1 million	2 million	3 million	4 million
Length 2				
coverage	96.4%	99.0%	99.5%	99.5%
median paraphrases retrieved	40.4%	53.53%	61.6%	64.9%
avg. Spearman $\overline{\rho}$	0.28	0.30	0.311	**0.315**
Length 3 fallback				
coverage	97.9%	98.7%	99.5%	99.5%
median paraphrases retrieved	39.32%	53.73%	61.7%	65.1%
avg. Spearman $\overline{\rho}$	0.282	0.299	0.312	**0.315**
Corpus Baseline				
coverage	86.1%	90.7%	93.6%	95.1%
avg. Spearman$\overline{\rho}$	0.15	0.161	0.169	0.169

Table 8.4: Overview over the effect of corpus/network size on the statistics and results for both strategies.

We can gain a more detailed understanding of the effect of network growth from one to four million sentences looking at the exact changes in Spearman score for the individual noun compounds. Using the larger network improved the score for 269 (71.9%) out of the 374 compounds covered by the 1 million network at length 2, but made it worse in 105 cases. The negative change has a median of -10.1% while the median of the positive change is 23.1%. In other words, if we pick a random noun compound, it is more likely that using the large network will improve the result than make it worse. At the same time, when we get an improvement the change is expected to be larger than when the result gets worse. Figure 8.3 shows the plot of the distribution of in- and decrease in Spearman score for each noun compound, expressed in percentage of change.[4] It is visibly right-skewed with positive mean, median and mode.

[3] We consider only those compounds covered already by the one million sentences-based algorithms to ensure we are making a fair comparison. However, taking all covered noun compounds into account does not majorly change the picture. The standard deviations for three million sentences are $\sigma = 0.110$ for the network and $\sigma = 0.137$ for the corpus.

[4] This measure was not applicable to three compounds with had a negative Spearman score on the one million network, which however turned positive in all cases on the three million network. We also removed four far outliers to the right in the plot so that the distribution is more clearly visible in the figure.

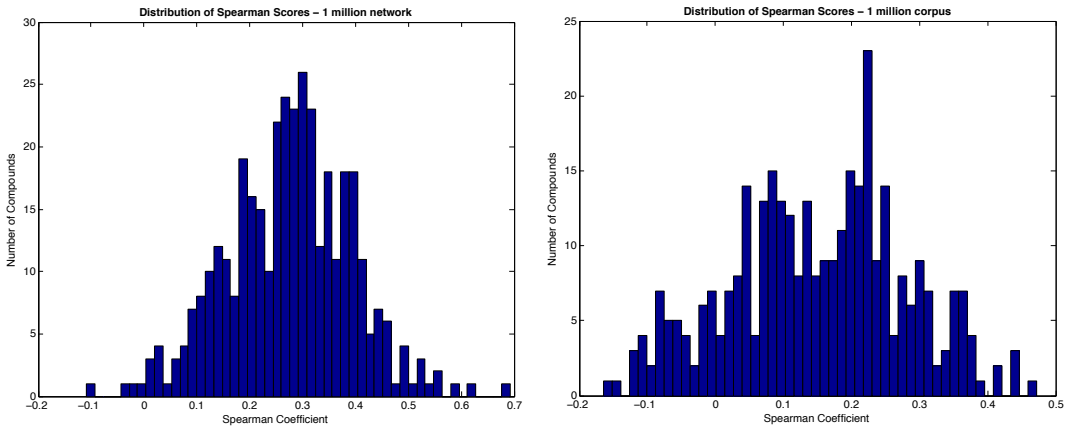


Figure 8.2: Histograms for the distribution of Spearman coefficient scores for the one million sentences based (a) network approach (b) corpus baseline.

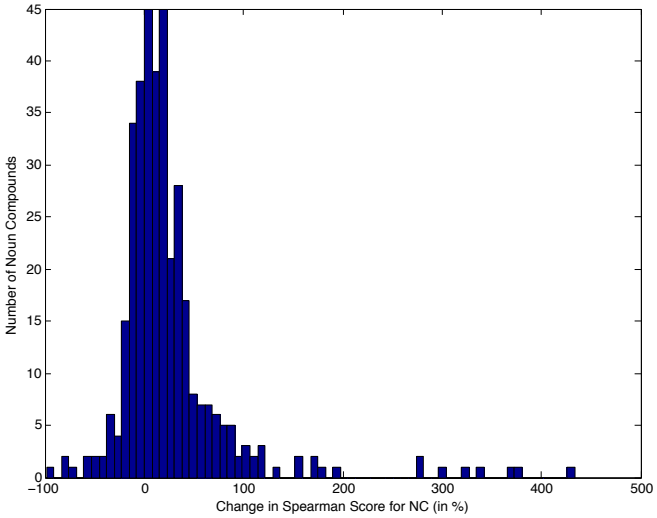


Figure 8.3: Histogram of the change in ρ from the 1 to the 4 million sentences network in percent.

8.3.2 Plain Wikipedia Network

We now look at the effect of using the corpus of all first sentences from Wikipedia without hyperlinks compared to the BNC for building a *plain* network. As the Wikipedia corpus contains about 2.7 million sentences, it seems most appropriate to compare it to the 3 million sentences BNC corpus. However, the comparison of the corpus baseline (cf. Table 8.5) shows that both the coverage and the median number of sentences retrieved for each compound is more similar to the 2 million sentences BNC corpus. In other words, the amount of information on the nouns in question contained in those two corpora is more comparable. Surprisingly, the baseline achieves slightly better results on the BNC than on the Wikipedia corpus. Due to their 'definitory' nature, we expected the first Wikipedia sentences to contain better quality descriptions of relations between concepts.

Corpus Baseline Comparison			
	$\overline{\rho}$	NCs covered	Median number of sentences per NC
BNC (3 million)	0.168	93.6%	52.5
BNC (2 million)	0.161	89.9%	40.0
Wikipedia first sentences	0.152	90.2%	42.5

Table 8.5: Average Spearman results and statistics of the corpus baseline on the 2 and 3 million sentences BNC and the Wikipedia first sentences corpus.

Looking at the networks, the BNC networks also achieve slightly better results (cf. Table 8.6). However, we note that the median number of retrieved skeleton paths is considerably smaller for the plain Wikipedia network than for the BNC-based networks. This in turn means that the plain Wikipedia network achieves comparable results based on considerably less information. We may consider this a hint that the quality of the relations and the overall network is higher for the Wikipedia network. Nevertheless, the data we have is somewhat inconclusive. For this task of noun compound paraphrasing, the results appear only marginally, if at all, influenced by the underlying 'quality' or type of corpus.

Network Approach Comparison			
Network	$\overline{\rho}$	Compounds covered	Median number of skeleton paths
Length 2			
BNC (3 million)	0.311	99.5%	493
BNC (2 million)	0.30	99.0%	345
Wikipedia first sentences	0.291	96.6%	146
Length 3 Fallback			
Wikipedia first sentences	0.292	98.7%	-

Table 8.6: Average Spearman results and statistics of the network-based approach on the 2 and 3 million sentences BNC based network and the Wikipedia first sentences network

8.4 Wikipedia Network

In this section, we compare the plain Wikipedia network with the version that uses the hyperlinks for disambiguation. The contrast allows us to understand the impact of working with terms versus concepts in the network. As in the case of the Semantic Relatedness, we are faced with having to come up with a mapping from words to Wikipedia articles. This step crucially influences the results, maybe even more so than in the previous task. The only other Wikipedia-based approach, Miklosch [2011], works with both a manual and an automatic mapping. Similarly to the findings by Milne and Witten [2008] for the Semantic Relatedness task, the results were slightly better using the automatic mapping. However, we decided to use the manual mapping that the author kindly provided us with for direct comparison. It contains only 363 out of the 388 compounds, because 25 had at least one noun for which no corresponding article was found.

Table 8.7 shows the comparison of results. The first core point to note is the considerable drop in coverage when the maximum length is set to the default of 2. Similarly, the median number of skeleton paths is a mere 6 compared to 146 for the plain version of the network. In fact, a total of 140 noun compounds meet the previously set threshold for activating the fallback of less than 10 skeleton paths - five times as many as for the plain version. There are two ways in which the disambiguation provided by the hyperlinks causes a drop in coverage and available paths. The first comes from the more fine-grained distinction into concepts. For example, the term *battery* can be associated with a variety of Wikipedia articles including Battery_(electricity), Battery_(crime), Battery_(chess) and Battery_cage. We would expect this to have a positive effect on the result such as in the case of *battery technology*, where the hyperlink integrated network achieves a Spearman coefficient of $\rho = 0.44$ using the Battery_(electricity)-Technology) mapping - compared to $\rho = 0.16$ on the plain network.

While the hyperlinks in this way lead to a desirable increase in precision, their inconsistent availability throughout the text can lead to an undesirable drop in recall of valuable paths. For instance, there are four sentences which contain both words *banana* and *industry*, on the basis of which the corpus baseline achieves a Spearman of $\rho = 0.07$ for the compound *banana industry*. The plain Wikipedia network in turn has a basis of 37 paths of length 2 to reach a convincing score of 0.35. However, not all occurrences of either *banana* and *industry* in the text are hyperlinked to their respective articles. In fact, we end up without a single sentence that contains both hyperlinks and the hyperlink integrated network is left with only a single skeleton path of length 2. This does not leave enough information to achieve good results and the Spearman drops to $\rho = 0.06$.

The average Spearman coefficient $\overline{\rho}$ achieved on the Wikipedia concept network is a relatively weak $\overline{\rho} = 0.198$ when only length 2 paths are considered and it covers a mere 65.3% of the NCs. Using the length 3 fallback increases the score to $\overline{\rho} = 0.221$ and the coverage to 91.5%. This leaves 31 noun compounds for which we cannot retrieve a ranking. For 11 of those, one of the

articles does not exist in the network, for the remaining 20 the path length of 3 is still too short to find a connection. The missing nodes are either due the fact that the manual mapping in Miklosch [2011] allowed for disambiguation and list pages to be used, both of which are excluded in our network or they are caused by retrieval errors in the corpus extraction part. Those with missing paths can primarily be traced back to unsatisfying mappings such as "growth center" to Economic_growth Center_(geometry), "jute product" to Jute Product_(biology) and "air pocket" to Atmosphere_of_earth Pocket.

Network Approach Comparison			
Network	$\overline{\rho}$	Compounds covered	Median number of skeleton paths
Plain Network - length 2	0.291	96.6% (of 388)	146
Plain Network - fallback	0.292	98.7% (of 388)	-
Hyperlink Network - length 2	0.198	65.3% (of 363)	6
Hyperlink Network - fallback	0.221	91.5% (of 363)	-

Table 8.7: Average Spearman results and statistics comparing the plain against the hyperlink integrated Wikipedia network.

With the considerable drop in recall caused by missing sense annotations, it is impossible to truly isolate the effect of having a concept network. In order to do so, we would need a fully annotated corpus. However, in order to find out whether disambiguation was having a positive effect despite the overall lower results compared to the plain network, we manually looked at a few examples in detail. Out of the 236 noun compounds covered by both the plain network and the concept network using a path length of 2, the concept network gave better results in 58 cases, which we would expect to be due to the disambiguation. We randomly picked five examples for which at least one of the two nouns was ambiguous - *carbon deposit*, *building site*, *cancer cell*, *storm cloud* and *battery technology*. A comparison of the skeleton paths retrieved by the term and the concept network showed that in each case, the paths in the term network included intermediate nodes clearly referring to a different underlying sense of one of the words while this was not the case in the concept network. For instance, *carbon deposit* included paths over *bond*, *credit* and *share*, referring to the financial meaning of 'deposit' and *storm cloud* had paths over *service*, *user* and *software* in relation to the computing resource meaning of 'cloud'. Table 8.8 lists problematic intermediate nodes for the other samples. In order to test whether these paths stemming from missing disambiguation indeed caused the low score of the term network, we manually filtered out the skeleton paths which clearly referred to a different sense and reran the paraphrase ranking. In all cases, the result improved considerably, to Spearman coefficients between 0.40 and 0.47. For all but one of the sample noun compounds, this exceeds the Wikipedia result. It indicates that having both disambiguated concepts and high recall gives the best performance and that the overall low results of the Wikipedia concept networks are caused through lacking coverage of hyperlinks. An interesting step would be to use automatic annotation tools such as Wikify! [Mihalcea, 2007] to annotate all remaining

nouns with hyperlinks giving full disambiguation and repeat the experiment.

Sample Noun Compounds					
Noun compound	Problematic paths	Term ρ	Paths excl.	Wikipedia mapping	Concept ρ
carbon deposit	bond, credit, instrument, share, term	-0.01	0.40	Carbon Deposit_(geology)	0.29
building site	protocol, computer, newspaper, pyramid	0.14		0.43 Building Construction_site	0.36
cancer cell	device, gun, proliferation, institution	0.21	0.43	Cancer Cell_(biology)	0.35
storm cloud	service, user, system, software, film	0.15	0.44	Storm Cloud	0.36
battery technology	artillery,force, army, weapon, law	0.26	0.47	Battery_(electricity) Technology	0.48

Table 8.8: Sample noun compounds with paths in term network belonging to other noun senses.

8.5 Comparison with Related Unsupervised Systems

Table 8.9 compares the results of previous unsupervised approaches to ours. To ensure a fair comparison, we corrected our scores for missing noun compounds by assuming a score of 0 for each of them. For all but the hyperlinked Wikipedia network, the change is marginal (<0.005) as the coverage was very high. For the Wikipedia network, the results drop from 0.221 to 0.204.

Even our smallest BNC network outperforms the previous unsupervised systems on the test set. In particular, we achieve better results than the previous BNC-based system, UCAM. Unfortunately, it is difficult to properly interpret the difference in result, because no further details are available on the methods used for the system. Judging by the difference between UCAM and our baseline, it appears likely that the basis for paraphrase ranking is more sophisticated than raw frequencies. However, we cannot conclude which exact aspect of our approach caused our results to be better.

For UCD-GOOGLE-0, results are only given for the training set. As our models are computationally expensive, we could not additionally run all of them on the training set. We opted for re-running only the 3 million sentences BNC network to get at least some idea of how training and test set compare. With an average Spearman coefficient of 0.27, the result is slightly lower than UCD-GOOGLE-0 at 0.31. Considering, however, that their system uses the entire Google n-gram corpus, which is based on more than 95 *billion* sentences, and additionally employs WordNet to deal with sparsity and ranks paraphrases according to conditional probability rather than simple raw frequencies, this is a very satisfying result.

Comparison with Related Approaches				
	Resources	Description	$\overline{\rho}$ test	$\overline{\rho}$ training
UCD-GOOGLE-0	Google n-gram, WordNet	Prob. model using template frequencies estimated from WordNet extended Google n-Gram corpus		0.31
UCAM	BNC	Model using verb-argument frequencies from the BNC	0.267	
Miklosch [2011]	Wikipedia	Paraphrase matching on first sentences of Wikipedia articles on selected paths in Wikipedia graph	0.191	
NC-INTERPRETER	Web snippets, WordNet	Verb-argument frequencies from parsed Web snippets and WordNet smoothing	0.186	
4m BNC network			**0.315**	
3m BNC network				0.270
1m BNC network			0.28	
Plain Wikipedia term network			0.288	
Wikipedia concept network			0.204	
Baseline - 4m BNC			0.169	

Table 8.9: Comparison of results with related approaches. Scores are adjusted for missing noun compounds by assuming a 0 score for them. UCD-GOOGLE-0 only has a result on the training set.

Our Wikipedia network outperforms the previous Wikipedia graph-based approach [Miklosch, 2011], despite the drop in overall score from adjusting for non-covered compounds. This shows the higher quality of the graph representation we achieve through deep linguistic information, in particular when we take into account the naivety of our actual paraphrase scoring algorithm. Miklosch [2011] use a highly optimised model that includes e.g. dynamic path length and paraphrase scoring refinements depending on paraphrase length and position on the path.

8.6 Summary

In this chapter, we demonstrated that the paths in our automatically built networks provide a powerful basis for high quality descriptions of relations between concepts. We tackled the task of unsupervised noun compound paraphrase ranking using a simple approach which counts the frequency of a paraphrase on paths of a fixed length between the noun compound parts. On the basis of our BNC term networks, we saw that the network-based strategy outperformed an equivalent corpus-based approach by far. Through the network structure, we were able to combine high quality relations from deep linguistic analysis and with the ability to leverage indirect paths, leading to both increased precision and increased recall compared to the text-based method. The results then not only increased continuously with network size, but also became more robust - a property not found for the corpus equivalent.

On the Wikipedia concept network, our naive approach outperformed the previous Wikipedia

approach, which uses a simpler graph model but employs a sophisticated and highly optimised paraphrase matching, demonstrating the quality of our network structure. While we also found indications of higher quality of the Wikipedia concept network compared to the Wikipedia term network, the latter achieved better results, which we attributed to the loss of recall in the concept network due to imperfect coverage of the text with hyperlinks.

Overall, even our smallest BNC term network outperformed the previous systems on this task and we achieved results competitive to the best performing system which employs the whole web-scale Google n-gram corpus and WordNet.

Chapter 9

Conclusion and Future Work

This thesis addressed the gap between manually constructed resources, which constitute high quality resources yet often lack coverage and cannot be created on demand, and statistical corpus-based techniques, which are automatic but rely on very large corpora to achieve good results. We demonstrated that automatically generated networks from text can contribute to filling this gap. They can be generated quickly and on demand in contrast to manually created resources. At the same time, they make it possible to achieve higher quality representations from less text than corpus-based methods, relieving the requirement of very large scale corpora.

We devised scaleable frameworks for building term networks from plain text and concept networks from Wikipedia. The difference between two types of networks lies in the disambiguation of the nodes with term networks having lexemes as nodes while concept networks are based on senses. In our framework, we used the hyperlinks given in Wikipedia as sense annotations. Our largest network covers the entire 88m word written language section of the British National Corpus and contains 1.2m terms and 21m relations. Our Wikipedia concept network covers 2.7m concepts and 7m relations. We provided serialisations into a variety of standardised formats such as RDF and GML and can hence directly be integrated with existing resources.

We proved the quality of our networks with two applications, Semantic Relatedness and unsupervised noun-compound paraphrasing. The first demonstrated the ability of the networks to provide a quantitative measure of relatedness and evaluated the quality of local networks as context. In parallel to widely applied corpus-based vector space models, we constructed a network-based semantic space model using spreading activation. We found that through leveraging the network structure our model not only outperformed the optimised corpus-based reference model, but the results suggest that the gain achieved through the network structure is comparable to using a corpus one order of magnitude larger with a standard model. This demonstrates the ability of the networks to extract higher quality representations from less text. In addition, the Wikipedia concept network outperformed the term network on the same corpus, illustrating the boost created by the additional level of semantics. Its results were

competitive to the best WordNet-based approach, demonstrating that automatically generated concept networks can provide a viable alternative to manual resources.

The second application, unsupervised noun compound paraphrase ranking, showed that indirect paths in the networks lead to high quality explicit descriptions of relations between concepts. Again, we found that using the network achieved considerably better results than using the text directly. The BNC network outperformed previous approaches on the same or comparable corpora and gave comparable results to the best performing system which employs the whole web-scale Google n-gram corpus. However, in this task the Wikipedia concept network gave lower results than its term equivalent. Our analysis suggests this may stem from the lack of coverage in sense annotations derived from hyperlinks, which leads to a low recall of concept occurrences.

This thesis demonstrated that automatically generated semantic networks present a scalable and versatile basis for lexical representation. Through leveraging the network structure, we consistently outperformed corpus-based approaches, achieving higher quality representations from less text. The results were robust across network sizes and showed that network-based representations are scalable even to large corpora such as the BNC. We conclude that automatically generated networks can contribute to filling the gap left by manually created resources and statistical methods that require very large corpora in domain-specific situations in which neither is available.

9.1 Future Work

We see three areas of future work: the building of frameworks and infrastructures, further NLP tasks and the creation of a network-based conceptual knowledge acquisition suite for semi-supervised Ontology Engineering.

Automatically disambiguated networks Our investigation showed that, while term networks already provide considerable gain over using the underlying corpus directly, having the additional semantic layer of disambiguated concepts can yield substantial further improvements. We worked with supervised disambiguation provided by Wikipedia hyperlinks, which has imperfect coverage and is restricted to the Wikipedia resource. In a next step, a general domain or Wikipedia specific sense disambiguation tool can be employed to derive sense annotations. Those can be incorporated without changes to our framework, as the implementation is independent of the source of annotation. This additional processing step also allows verbs to be disambiguated, which would further increase the quality of the representation.

Creation of a standardised markup We found that standard graph languages are insufficient to express all aspects of the type of semantic network advocated here. We were able to provide a serialisation into the Semantic Web language RDF. However, it is not the most

natural fit due to its foundation in logic-based knowledge representation. In the context of wider uses of semantic networks based on social and other media, the creation of a standardised markup language specifically for semantic networks is an important step towards their wide application. This view has already been advocated and sketched out in Harrington and Wojtinnek [2011].

In this thesis we looked at two applications and our methods were based on the idea of transferring techniques from corpus-based statistical NLP to the networks. In a similar fashion, other applications can be approached. To outline one:

Text Similarity The most basic text similarity algorithms are based on word overlap of the texts, refined by weighting and accounting for text length. From a resource perspective, this approach can be enhanced by using WordNet to account for synonyms and to take advantage of indirect relations between words in the texts. From a statistical view, a text can be represented by a combination of the vectors of individual words and then the similarity of these combined representations can be compared to measure text similarity [Grefenstette et al., 2011]. Following our network-based semantic space model, we could generate a representation of a text using spreading activation by activating its contained terms in the network and retrieving the activation pattern. In this way, we would also construct a combined statistical representation, which is also able to pick up on indirect relationships.

One of our core motivations for this work was to illustrate the ability of automatically generated networks to present a basis for high quality conceptual knowledge acquisition for domains and real world applications from a specific corpus. Ontology engineers are often faced with having to model and construct ontologies within very specific guidelines. To give an example, Sesen et al. [2010] builds an ontology for regulatory compliance in pharmaceutical manufacturing formalising regulation texts.

Semi-supervised ontology engineering In an ontology engineering process, domain experts and ontology engineers work together to create the resource, often guided by a collection of texts[1]. It involves a variety of steps including the identification of key terminology, the construction of a taxonomy and the specification of relations and attributes. In a semi-supervised process, conceptual information is automatically extracted from the text to guide the engineering process. Our network structure could enhance this process as it pre-structures the underlying information in a form which already closely resembles an ontological structure. We would like to build an ontological engineering suite based on the networks and our RDF serialisation already provides a core ingredient to its implementation. Even more, partially motivated by our work, Augenstein et al. [2012] have shown that the output of the semantic analysis tool Boxer, which provides the basis of the networks, can be automatically translated into the logic

[1] A full tutorial outlining the steps of manual ontology engineering can be found at `http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf`

used in ontology modelling. This bridges the gap in formalism and allows for easy and practical transition.

Appendix: Published Articles and Papers

Journals

- P.-R. Wojtinnek, J. Völker, and S. Pulman. Building Semantic Networks from Plain Text and Wikipedia with Application to Semantic Relatedness and Noun Compound Paraphrasing. *International Journal of Semantic Computing (IJSC). Special Issue on Semantic Knowledge Representation*, 2012.

Conference Proceedings

- P.-R. Wojtinnek and S. Pulman. Semantic Relatedness from Automatically Generated Semantic Networks. *In: Proceedings of the Ninth International Conference on Computational Semantics, IWCS11.* Association for Computational Linguistics, 2011
- B. Harrington and P.-R. Wojtinnek. Creating a Standardized Markup Language for Semantic Networks. In *Proceedings of the 5th IEEE International Conference on Semantic Computing*, 2011
- P.-R. Wojtinnek, B. Harrington, S. Rudolph, and S. Pulman. Conceptual Knowledge Acquisition Using Automatically Generated Large-Scale Semantic Networks. In: *Proceedings of the 18th International Conference on Conceptual Structures*, 2010.

Bibliography

E. Agirre and A. Soroa. UBC-AS : A Graph Based Unsupervised System for Induction and Classification. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, 2007.

E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

A. Almuhareb and M. Poesio. Attribute-based and value-based clustering: An evaluation. In *Proceedings of EMNLP*, pages 158–165, 2004.

I. Augenstein, S. Padó, and S. Rudolph. Lodifier: generating linked data from unstructured text. *The Semantic Web: Research and Applications*, pages 210–224, 2012.

T. Baldwin and T. Tanaka. Translation by Machine of Complex Nominals: Getting It Right. In *Proceedings of the Workshop on Multiword Expressions*, pages 24–31. Association for Computational Linguistics, 2004.

M. Berland and E. Charniak. Finding parts in very large corpora. In *Proceedings of the Annual Meeting of the ACL*, volume 37, pages 57–64. Association for Computational Linguistics, 1999.

C. Biemann. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93, 2005.

J. Bos. Wide-coverage semantic analysis with boxer. In J. Bos and R. Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Research in Computational Semantics, pages 277–286. College Publications, 2008.

J. Bos, S. Clark, M. Steedman, J. R. Curran, and J. Hockenmaier. Wide-Coverage Semantic Representations from a CCG Parser. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, 2004.

J. A. Bullinaria and J. P. Levy. Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study. *Behavior research methods*, 39(3):510–26, Aug. 2007.

C. Butnariu and T. Veale. A Concept-Centered Approach to Noun-Compound Interpretation. *Computational Linguistics*, (August), 2008.

C. Butnariu, S. N. Kim, P. Nakov, D. Ó Séaghdha, S. Szpakowicz, and T. Veale. Semeval-2010 task 9: the interpretation of noun compounds using paraphrasing verbs and prepositions. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 100–105, 2009.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, 2010.

R. P. Chaves. Wordnet and automated text summarization. In *In Proceedings of the 6 th Natural Language Processing Pacific Rim Symposium, NLPRS*, 2001.

P. Cimiano and S. Staab. Learning concept hierarchies from text with a guided hierarchical clustering algorithm. *Learning and Extending Lexical Ontologies by using Machine Learning Methods*, pages 6–16, 2005.

P. Cimiano and J. Wenderoth. Automatically learning qualia structures from the web. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 28–37. Association for Computational Linguistics, 2005.

P. Cimiano, A. Hotho, and S. Staab. Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text. In *ECAI*, volume 16, page 435, 2004.

P. CIMIANO, J. VÖLKER, and R. STUDER. Ontologies on demand?: A description of the state-of-the-art, applications, challenges and trends for ontology learning from text. *Information Wissenschaft und Praxis*, 57(6-7):315–320, 2006.

S. Clark and J. R. Curran. Parsing the WSJ using CCG and Log-Linear Models. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

A. Collins and E. Loftus. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407, 1975.

F. Crestani. Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

J. Curran. *From distributional to semantic similarity*. PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics., 2004.

D. Davidov and A. Rappoport. Classification of semantic relationships between nominals using pattern clusters. *ACL08*, 2008.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

B. Dorow, D. Widdows, K. Ling, J. Eckmann, D. Sergi, and E. Moses. Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination. In *Workshop MEANING-2005*, 2004.

A. Faaborg and H. Lieberman. A goal-oriented web browser. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 751–760, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124883. URL `http://doi.acm.org/10.1145/1124772.1124883`.

C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

R. Ferrer i Cancho and Solé. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1482):2261–2265, 2001.

R. Ferrer i Cancho, R. Solé, and R. Köhler. Patterns in Syntactic Dependency Networks. *Physical Review E*, 69(5):1–8, 2004.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing Search in Context: The Concept Revisited. *ACM Trans. Inf. Syst.*, 20(1), 2002.

J. R. Firth. Asynopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, (45):1–32, 1957.

E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical intelligence*, 2007.

S. Gouws, G.-j. V. Rooyen, and H. A. Engelbrecht. Measuring Conceptual Similarity by Spreading Activation over Wikipedia's Hyperlink Structure. In *COLING*, number August, pages 46–54, 2010.

E. Grefenstette, M. Sadrzadeh, S. Clark, B. Coecke, and S. Pulman. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 125–134. Association for Computational Linguistics, 2011.

S. Guadarrama and M. Garrido. Concept-analyzer: A tool for analyzing fuzzy concepts. In *Proceedings of IPMU*, volume 8, page 1085, 2008.

S. Harabagiu, D. Moldovan, M. Paşca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Gîrju, V. Rus, and P. Morărescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 282–289, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1073012.1073049. URL `http://dx.doi.org/10.3115/1073012.1073049`.

Y. Haralambous and V. Klyuev. A Semantic Relatedness Measure Based on Combined Encyclopedic, Ontological and Collocational Knowledge. *CoRR*, abs/1107.4723, 2011.

B. Harrington. *ASKNet: Automatically Creating Semantic Knowledge Networks from Natural Language Text*. PhD thesis, University of Oxford, 2009.

B. Harrington. A Semantic Network Approach to Measuring Semantic Relatedness. In *COLING'10: Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.

B. Harrington and S. Clark. ASKNet: Automated Semantic Knowledge Network. In *AAAI'07: Proceedings of the 22nd Conference on Artificial intelligence*, 2007.

B. Harrington and P.-R. Wojtinnek. Creating a Standardized Markup Language for Semantic Networks. In *Proceedings of the 5th IEEE International Conference on Semantic Computing*, 2011.

C. Havasi, J. Pustejovsky, and A. Rumshisky. An evaluation of the Brandeis Semantic Ontology. In *Proceedings of the Fourth International Workshop on Generative Approaches to the Lexicon*, 2007.

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

I. Hendrickx, S. Kim, Z. Kozareva, P. Nakov, D. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 94–99, 2009.

A. Herdağdelen, K. Erk, and M. Baroni. Measuring Semantic Relatedness with Vector Space Models and Random Walks. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*. Association for Computational Linguistics, 2009.

D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 268–275. Association for Computational Linguistics, June 1990.

G. Hirst and D. St-Onge. *WordNet: An electronic lexical database*, chapter Lexical chains as representations of context for the detection and correction of malapropisms, pages 305–332. Cambridge, 1998.

T. Hughes and D. Ramage. Lexical Semantic Relatedness with Random Graph Walks. In *EMNLP-CoNLL'07: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

M. Jarmasz and S. Szpankowicz. Roget's Thesaurus: a Lexical Resource to Treasure. In *Proceedings of the NAACL WordNet and Other Lexical Resources workshop*, 2001.

K. Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon.* PhD thesis, University of Pennsylvania, 2005.

H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an english dictionary. In *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, pages 232–239. Association for Computational Linguistics, 1993.

T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104 (2):211–240, Apr. 1997.

B. Lemaire and G. Denhière. Incremental Construction of an Associative Network from a Corpus. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, 2002.

D. B. Lenat. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38 (11):33–38, Nov. 1995. ISSN 0001-0782. doi: 10.1145/219717.219745. URL `http://doi.acm.org/10.1145/219717.219745`.

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

J. Lev. *The Syntax and Semantics of Complex Nominals.* Academic Press, New York, 1978.

G. Li and A. Lopez-Fernandez. UCD-Goggle : A Hybrid System for Noun Compound Paraphrasing. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 230–233, 2010.

D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics*, pages 768–774, Morristown, NJ, USA, Aug. 1998. Association for Computational Linguistics. doi: 10.3115/980691.980696. URL `http://portal.acm.org/citation.cfm?id=980691.980696`.

H. Liu and P. Singh. ConceptNetA Practical Commonsense Reasoning Tool-kit. *BT technology journal*, 22(4):211–226, 2004a.

H. Liu and P. Singh. Commonsense reasoning in and over natural language. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3215 of *Lecture Notes in Computer Science*, pages 293–306. 2004b.

W. Lowe. Towards a theory of semantic space. In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, pages 576–581, 2001.

K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 28:203–208, 1996. ISSN 1554-351X.

R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In *HLT-NAACL*, pages 196–203, 2007.

Y. Miklosch. Wikipedia-based Analysis of Nominal Compounds (Wikipedia-basierte Analyse Nominaler Komposita). Diplomarbeit, Universität Mannheim, 2011.

G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

D. Milne and I. H. Witten. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *In Proceedings of AAAI 2008*, 2008.

J. Mitchell and M. Lapata. Vector-based Models of Semantic Composition. *Computational Linguistics*, (June):236–244, 2008.

M. Mohler and R. Mihalcea. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 567–575, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1609067.1609130`.

P. Nakov. Paraphrasing verbs for noun compound interpretation. In *Proceedings of the Workshop on Multiword Expressions*, 2008a.

P. Nakov. Improved Statistical Machine Translation Using Monolingual Paraphrases. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 338–342. IOS Press, 2008b.

P. Nakov and M. Hearst. Using verbs to characterize noun-noun relations. *Artificial Intelligence: Methodology, Systems, and Applications*, pages 233–244, 2006.

R. Navigli and P. Velardi. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1075–1086, July 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.149. URL `http://dx.doi.org/10.1109/TPAMI.2005.149`.

P. Nulty. UCD-PN : Selecting General Paraphrases Using Conditional Probability. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 234–237, 2010.

S. Padó and M. Lapata. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199, June 2007.

P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2006.

E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. W3C Recommendation, 15 Jan. 2008. Available at `http://www.w3.org/TR/rdf-sparql-query/`.

J. Pustejovsky, C. Havasi, J. Littman, A. Rumshisky, and M. Verhagen. Towards a generative lexical resource: The brandeis semantic ontology. In *Proceedings of the Fifth Language Resource and Evaluation Conference*, 2006.

M. R. Quillian. Word concepts: a theory and simulation of some basic semantic capabilites. *Behavioral Science*, 12(5), 1967.

K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch. A Word at a Time : Computing Word Relatedness using Temporal Semantic Analysis. *Time*, pages 337–346, 2011.

P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9.

S. Richardson, W. Dolan, and L. Vanderwende. Mindnet: acquiring and structuring semantic information from text. In *Proceedings of the 17th International Conference on Computational linguistics*, pages 1098–1102. Association for Computational Linguistics, 1998.

A. Rozovskaya and R. Girju. Identifying semantic relations in context: Near-misses and overlaps. In *Proceedings of the International Conference RANLP-2009*, pages 381–387. Association for Computational Linguistics, 2009.

H. Rubenstein and J. Goodenough. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

M. Sahlgren. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* PhD thesis, University Stockholm, 2006.

H. Schütze. Dimensions of meaning. In *Proceedings of Supercomputing*, pages 787–796, 1992.

M. Sesen, P. Suresh, R. Banares-Alcantara, and V. Venkatasubramanian. An ontological framework for automated regulatory compliance in pharmaceutical manufacturing. *Computers & chemical engineering*, 34(7):1155–1169, 2010.

J. F. Sowa. Conceptual graphs. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*, pages 213–237. Elsevier, 2008.

M. Strube and S. P. Ponzetto. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *AAAI'06: proceedings of the 21st national conference on Artificial intelligence*, 2006.

F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semant.*, 6(3):203–217, Sept. 2008. ISSN 1570-8268. doi: 10.1016/j.websem.2008.06.001. URL `http://dx.doi.org/10.1016/j.websem.2008.06.001`.

M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management*, pages 67–74. ACM, 1993.

S. Szumlanski and F. Gomez. Automatically Acquiring a Semantic Network of Related Concepts. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10)*, 2010.

S. Szumlanski and F. Gomez. Evaluating a Semantic Network Automatically Constructed from Lexical Co-occurrence on a Word Sense Disambiguation Task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, 2011.

M. Tatu and D. Moldovan. A semantic approach to recognizing textual entailment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 371–378, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220622. URL `http://dx.doi.org/10.3115/1220575.1220622`.

H. W. Tinholt and A. Nijholt. Computational humour: Utilizing cross-reference ambiguity for conversational jokes. In *Proceedings of the 7th international workshop on Fuzzy Logic and Applications: Applications of Fuzzy Sets Theory*, WILF '07, pages 477–483, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73399-7. doi: 10.1007/978-3-540-73400-0_60. URL `http://dx.doi.org/10.1007/978-3-540-73400-0_60`.

G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos. I.: Word sense disambiguation with spreading activation networks generated from thesauri. In *20th Int. Joint Conf. in Artificial Intelligence*, pages 1725–1730, 2007.

J. Véronis. Hyperlex: Lexical Cartography for Information Retrieval. *Computer Speech & Language*, 18(3):223–252, 2004.

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

J. Weeds, D. Weir, and D. McCarthy. Characterising Measures of Lexical Distributional Similarity. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004.

D. Wijaya and P. Gianfortoni. Nut case: What does it mean?: Understanding semantic relationship between nouns in noun compounds through paraphrasing and ranking the paraphrases. In *Proceedings of the 1st International Workshop on Search and Mining Entity-Relationship Data*. ACM, 2011.

P. Wojtinnek et al. Conceptual knowledge acquisition using automatically generated large-scale semantic networks (this is not the final version). Technical report, 2010a.

P.-R. Wojtinnek and S. Pulman. Semantic Relatedness from Automatically Generated Semantic Networks. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11. Association for Computational Linguistics, 2011.

P.-R. Wojtinnek, B. Harrington, S. Rudolph, and S. Pulman. Conceptual Knowledge Acquisition Using Automatically Generated Large-Scale Semantic Networks. In *Proceedings of the 18th International Conference on Conceptual Structures*, 2010b.

P.-R. Wojtinnek, J. Völker, and S. Pulman. Building Semantic Networks from Plain Text and Wikipedia with Application to Semantic Relatedness and Noun Compound Paraphrasing. *International Journal of Semantic Computing (IJSC). Special Issue on Semantic Knowledge Representation*, 2012.

Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

S. Wubben. UvT: Memory-based Pairwise Ranking of Paraphrasing Verbs. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pages 260–263, 2010.

M. Yazdani and A. Popescu-Belis. A Random Walk Framework to Compute Textual Semantic Similarity: A Unified Model for Three Benchmark Tasks. *2010 IEEE Fourth International Conference on Semantic Computing*, pages 424–429, Sept. 2010. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5629087`.

E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. WikiWalk: Random Walks on Wikipedia for Semantic Relatedness. In *TextGraphs-4: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, Morristown, NJ, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-54-1.

T. Zesch and I. Gurevych. Wisdom of Crowds versus Wisdom of Linguists – {M}easuring the Semantic Relatedness of Words. *Natural Language Engineering*, 16(01):25–59, 2010.

Z. Zhang, A. Gentile, and F. Ciravegna. Harnessing different knowledge sources to measure semantic relatedness under a uniform model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 991–1002. Association for Computational Linguistics, 2011.

Z. Zhang, A. L. Gentile, and F. Ciravegna. Recent advances in methods of lexical semantic relatedness - a survey. *Natural Language Engineering*, pages 1–69, 2012.