Combined Distributional and Logical Semantics

Mike Lewis

Doctor of Philosophy

Institute for Language, Cognition and Computation

School of Informatics

University of Edinburgh

2014

Abstract

Understanding natural language sentences requires interpreting words, and combining the meanings of words into the meanings of sentences. Despite much work on lexical and compositional semantics individually, existing approaches are unlikely to offer a complete solution. This thesis introduces a new approach, which combines the benefits of distributional lexical semantics and logical compositional semantics.

Linguistic theories of compositional semantics have shown how logical forms can be built for sentences, and how to represent semantic operators such as negatives, quantifiers and modals. However, computational implementations of such theories have shown poor performance on applications, mainly due to a reliance on incomplete hand-built ontologies for the meanings of content words. Conversely, distributional semantics has been shown to be effective in learning the representations of content words based on collocations in large unlabelled corpora, but there are major outstanding challenges in representing function words and building representations for sentences.

I introduce a new model which captures the main advantages of logical and distributional approaches. The proposal closely follows formal semantics, except for changing the definitions of content words. In traditional formal semantics, each word would express a different symbol. Instead, I allow multiple words to express the same symbol, corresponding to underlying concepts. For example, both the verb *write* and the noun *author* can be made to express the same relation. These symbols can be learnt by clustering symbols based on distributional statistics—for example, *write* and *author* will share many similar arguments. Crucially, the clustering means that the representations are symbolic, so can easily be incorporated into standard logical approaches.

The simple model proves insufficient, and I develop several extensions. I develop an unsupervised probabilistic model of ambiguity, and show how this model can be built into compositional derivations to produce a distribution over logical forms. The flat clustering approach does not model relations between concepts, for example that *buying* implies *owning*. Instead, I show how to build graph structures over the clusters, which allows such inferences. I also explore if the abstract concepts can be generalized cross-lingually, for example mapping French verb *écrire* to the same cluster as the English verb *write*. The systems developed show good performance on question answering and entailment tasks, and are capable of both sophisticated multi-sentence inferences involving quantifiers, and subtle reasoning about lexical semantics.

These results show that distributional and formal logical semantics are not mutually exclusive, and that a combined model can be built that captures the advantages of each.

Acknowledgements

I can't imagine this thesis happening if I was supervised by anyone other than Mark—not just because his profound understanding of language and AI, but because of his enthusiasm, patience, and ambition. I'm also grateful to Ewan and Mirella for helpful conversations which improved this thesis.

My thesis would have been very different had I not spent a happy winter working with at Google in Zurich. Thanks Yasemin for giving me such an interesting project, and all the support and encouragement, which ultimately led to the main idea behind this thesis. Lots of other people made Switzerland a very fun place to be, including Enrique, Guillermo, Håvard, Jude, Katja, Sören and Sven.

Other members of Mark's research group have given countless interesting conversations, about work and otherwise—thanks to Aciel, Greg, Jeff, minimark, Omri, Teju, Tom, and Tos. B-man and Siva get special mentions for providing doughnuts when all seemed lost. I'm also grateful to everyone who was patient enough to explain the mysteries of machine learning to me using small words—Ben and Micha stand out here.

Also, a big thanks to everyone else in ILCC who's made the last few years so fun, who are far too many to enumerate, but a representative sample includes Alasdair, Annie, Carina, Dave, Des, Diego, Dominicos, Dominika, Eva, Kira, Lea, Luke, Maria, Michael, Sasa, and Stella.

Finally, my beautiful new wife (!) Cat for all her help and support in so many ways, and for dragging me away from my thesis for long enough to fill the last few years with amazing adventures.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualication except as specied.

The candidate confirms that the work submitted is his/her own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others

The following parts of this thesis are based on previously published material. All work in these publications it directly attributable to the candidate, with guidance from Mark Steedman.

- The Evaluation section of Chapter 3 is based on material published in Lewis and Steedman [2013a].
- Chapter 4 is based on Lewis and Steedman [2013a].
- Chapter 5 is based on Lewis and Steedman [2013b].
- Parts of the Future Work section of Chapter 6 are based on Lewis and Steedman [2014a].

(*Mike Lewis*)

Table of Contents

1 Introduction 1

1.1 Background . . . 2

1.1.1 Formal semantics . . . 2

1.1.2 Distributional semantics . . . 3

1.2 This Thesis . . . 4

1.2.1 Contributions . . . 4

1.2.2 Outline of Thesis . . . 5

2 Related Work 7

2.1 Introduction . . . 7

2.2 Logical Semantics . . . 8

2.2.1 Formal Semantic Approaches . . . 8

2.2.2 Natural Logic . . . 9

2.3 Supervised Semantics . . . 9

2.3.1 Domain Specific . . . 10

2.3.2 Wide Coverage . . . 10

2.4 Distributional Semantics . . . 11

2.4.1 Vector Space Approaches . . . 12

2.4.2 Pattern-based Approaches . . . 16

2.4.3 Compositional Symbolic Approaches . . . 18

3 A Computational Model of Natural Semantics 21

3.1 Introduction . . . 21

3.2 Combinatory Categorial Grammar . . . 22

3.2.1 Syntactic Categories . . . 23

3.2.2 Semantic Interpretations . . . 24

3.2.3 Combinatory Rules 24
3.2.4 Unary Rules 26
3.2.5 Type Raising 26
3.3 Natural Semantics for CCG 28
3.3.1 Quantifier Scope 28
3.3.2 Negation Scope 33
3.3.3 Correction to Natural Semantics 35
3.4 Adding Natural Semantics to CCG Syntactic Parsers 36
3.4.1 Automatic Lexicon for Content Words 37
3.4.2 Hand-built Lexicon for Function Words 39
3.4.3 Post-processing Syntax Trees 41
3.5 Conversion to First-Order Logic 43
3.5.1 Finding Skolem terms with a given scope 44
3.5.2 Replacing Skolem terms with Quantifiers 44
3.5.3 Main Translation Algorithm 45
3.5.4 Other 47
3.6 Experiments 48
3.6.1 Coverage 48
3.6.2 Qualitative Evaluation 48
3.6.3 Comparison with Boxer 55
3.6.4 Comparison with Dependency Syntax 55
3.6.5 Experiments on the FraCaS Suite 56
3.7 Future Work 59
3.8 Conclusions 60

4 Combined Distributional and Logical Semantics **61**

4.1 Introduction 61
4.2 Motivating Combining Distributional and Logical Semantics 62
4.3 A Simple Model for Combining Distributional and Logical Semantics 63
4.3.1 Initial Semantic Analysis 64
4.3.2 Distributional Relation Clustering 67
4.4 Modelling Ambiguity 69
4.4.1 Entity Typing 70
4.4.2 Topic Model 71
4.4.3 Typing in Logical Form 72

4.4.4 Distributional Relation Clustering 74
4.4.5 Semantic Parsing Using Relation Clusters 75
4.5 Experiments 76
4.5.1 Experimental Setup 77
4.5.2 Question Answering Experiments 78
4.5.3 Qualitative Evaluation 82
4.6 Comparison with Related Work 83
4.6.1 Unsupervised Semantic Parsing 84
4.6.2 Compositional Vector Space Models 84
4.6.3 Natural Logic 86
4.6.4 Distributional Inference Rules 86
4.7 Conclusions 87

5 Unsupervised Induction of Cross-lingual Semantic Relations 89
5.1 Introduction 89
5.2 Motivation 90
5.3 Overview of Approach 92
5.4 Predicate Extraction 92
5.5 Entity Semantics 95
5.5.1 Entity Linking 95
5.5.2 Entity Typing 96
5.6 Relation Clustering 97
5.6.1 Monolingual Clustering 97
5.6.2 Cross-lingual Cluster Alignment 98
5.7 Cross-lingual Question Answering Experiments 99
5.7.1 Experimental Setup 99
5.7.2 Baseline 101
5.7.3 Results 101
5.8 Translation Reranking Experiments 104
5.9 Related Work 106
5.10 Future Work 108
5.11 Conclusions 109

6 Directional Inference for Combined Distributional and Logical Semantics 111
6.1 Introduction 111
6.2 Global Learning of Entailment Graphs 112

6.2.1 Learning Entailment Graphs 114
6.3 Entailment Graphs for CCG 115
6.3.1 Training Data 116
6.3.2 Features 116
6.3.3 Lexical Entries 119
6.4 Implicative Verbs 120
6.5 Entailment with Combined Distributional and Logical Semantics . . . 121
6.5.1 Inference 121
6.5.2 Missing Predicates 122
6.6 Experiments 123
6.6.1 Dataset 123
6.6.2 Experimental Setup 124
6.6.3 Comparison Systems 126
6.6.4 Results 127
6.7 Future Work 130
6.7.1 Entity Typing 131
6.7.2 Temporal Semantics 131
6.7.3 N-ary Relations 133
6.7.4 Multiword Compositionality 133
6.7.5 Light Verb Constructions 134
6.8 Conclusion 135

7 Conclusions **137**

8 Appendix **141**
8.1 Questions used in Chapter 4 141
8.2 English Questions used in Chapter 5 155
8.3 French Questions used in Chapter 5 169

Bibliography **185**

CHAPTER 1

Introduction

This thesis describes a new approach to the automatic interpretation of natural language sentences. It brings together two successful previous models; the theory of formal semantics developed in the linguistics literature, and recent work on distributional semantics from the natural language processing community. I will argue that while both of these approaches are individually powerful, they ultimately have significant limitations which are likely to preclude them from offering a complete explanation of natural language semantics. However, their strengths and weaknesses are strikingly complementary, which gives a powerful motivation for developing a combined model.

Natural language is the dominant means by which people express and communicate knowledge. If computers could interpret natural language sentences, they would have access to this knowledge, which would have immense practical value for applications such as automated question answering and information retrieval.

This problem is also extremely challenging. Language is extremely productive, with new words and meanings being developed constantly—consequently, attempts to confine it to some manually constructed ontology have largely failed. Individual words can have many meanings, but also many words can express the same meaning. Whilst some words appear to express logical concepts like negation or quantification, others are much harder to formalise. Understanding individual words is not enough—the meanings of words must be composed into the meanings of sentences. The framework introduced in this thesis offers a partial solution to all of these problems.

The success of a system at interpreting natural language can be evaluated by its performance on the problem of *natural language inference*. Natural language inference involves understanding how the meaning of some sentences relates to the meaning of others—for example, knowing that the sentences *Obama was born in Kenya* and *Obama's birthplace is not Kenya* contradict each other. This general framework allows the evaluation of any mono-modal aspect of language understanding, but makes no assumptions as to *how* the language is being understood. Consequently, there is great diversity in existing approaches, which may represent sentences as bags of words, high-dimensional vectors, or first-order logical forms. The models developed in this thesis will be evaluated by their performance on this task.

In this chapter, I will first give a very brief background on two of the most successful approaches to the problem of natural language inference, discussing their advantages and drawbacks. I will then sketch a method for combining them, which I will argue gives the advantages of both. This method is the central idea of the thesis. Section 1.2.2 gives an overview of the structure of the thesis.

1.1 Background

The task of language interpretation can be divided into the problems of interpreting the meanings of individual words, and composing them into the meanings of phrases and sentences. Words can further be divided into closed-class function words, and open-class content words, which pose distinct challenges for representing their meanings. Whilst these problems have received much attention individually, the most popular solutions are mutually incompatible.

1.1.1 Formal semantics

Theories of formal semantics aim to map sentences onto *logical forms*. These logical forms support inference (for example, using theorem provers).

Logical forms are built by first assigning an interpretation to each word, typically using lambda-calculus as glue language, and then combining these into the meaning of the sentences. Interpretations for content words can be generated automatically, by simply using the word itself as a symbol in the logical form.

The use of first-order logical forms makes it straightforward to model semantic phenomena such as negation and quantification, as these concepts are an integral part of first-order logic. Understanding negation is clearly extremely important—it is the difference between a question-answering system saying *yes* or *no*. Quantifiers are also potentially very powerful, as they express information over a number of individuals at once. If we are told, for example, that *Every dog is a mammal* then we know something about all the world's dogs.

Another key advantage of formal semantics is that composition is easily explained. In the theory of CCG used in this thesis, the meanings of expressions are combined using exactly the same standard function application and composition operators as are used in the syntax tree. Syntax trees can now be built automatically using treebank trained parsers with reasonable accuracy—and, given the syntax tree, semantic composition is straightforward.

Several attempts have been made to build wide-coverage semantic parsing systems based on formal semantics, but all have these have shown low recall on practical applications, such as entailment [Bos and Markert, 2005, Bobrow et al., 2007]. The main reason is that they have a weak model of the meanings of content words, which is critical to almost all natural language inference. Existing lexical resources such as WordNet [Miller, 1995] have proved of limited help in addressing this problem.

Whilst formal semantics elegantly explains compositionality and the meanings of function words, it has ultimately failed to show strong performance on real world applications. Being able to negate, quantify and compose meanings is of little use without a good model of what the underlying meanings are.

1.1.2 Distributional semantics

Distributional semantics[1] takes an orthogonal approach to formal semantics. It aims to induce the meanings of words from unlabelled text in an unsupervised way. A vector is constructed for every word based on its contexts in a large corpus (using one of many possible methods), and it is assumed that similarity in the vector space represents semantic similarity. It has been shown that the similarity of such vectors correlates well with human judgements of word similarity [McDonald, 2000, Huang et al., 2012].

[1] In this thesis, I use a general sense of the term *distributional semantics* which includes inference rules derived from co-occurrence vectors—following, for example, Lin and Pantel [2001].

However, there are many unsolved problems in distributional semantics. Despite many proposals, it is not clear how best to combine the meanings of words to capture the meanings of sentences [Mitchell and Lapata, 2008, Coecke et al., 2010, Socher et al., 2011], and so far there is relatively little evidence to show that vector-space representations of sentences are useful for entailment or question answering. Further problems are likely to be faced when trying to combine the meanings of sentences into those of documents, or whole encyclopedias.

It is also unclear how function words, such as *not* or *every*, should be modelled in distributional semantics. Various attempts have been made to model these in vector spaces [Socher et al., 2012, Baroni et al., 2012, Hermann et al., 2013, Grefenstette, 2013], but current work has significant limitations compared to logical approaches. Many other semantic phenomena—including modality, tense, coordination—have well developed solutions in the formal semantics literature, but will require much work to adequately model in vector spaces.

1.2 This Thesis

1.2.1 Contributions

The major contribution of this thesis is a new approach to combining distributional and logical semantics.

The first important contribution is the first implementation of a new theory of formal semantics [Steedman, 2012], with efficient mechanisms for reasoning about quantifier and negation scope. I also contribute a novel algorithm for converting the representations to standard first-order logic. I show the implementation has wide coverage, models linguistically complex constructions, and is capable of making complex inferences.

As discussed in Section 1.1, formal semantics is strong at modelling function words and compositionality, but weak at expressing the meanings of content words, meaning the performance of the purely logical approach is low on practical applications. Conversely, distributional semantics is able to learn the meaning of content words, but expressing function words and composing the meaning of words are both problematic.

The complementary strengths and weaknesses of these two approaches motivates trying to combine them in a way that captures the strengths of each. The approach introduced here is firmly rooted in formal semantics, apart from for the semantics of content words. I make the hypothesis that there is some set of discrete abstract concepts that words may express. Each of these concepts can be assigned a symbol, and the lexical semantics of a word can use this symbol as a predicate.

I aim to uncover these concepts using distributional semantics. Standard formal-semantics symbols, which capture predicate-argument structure, can be clustered based on their arguments in a large corpus. When parsing new sentences, the cluster identifier can be used as a symbol. The clustering converts noisy, continuous, high-dimensional vector-space representations of words into atomic symbols. The clustering aims both to uncover the latent underlying relations that language can express, and to show how to map words onto those relations. The induced symbols can be conjoined, negated or quantified, just like any other, preserving the advantages of formal semantics. However, because synonyms like *buy* and *purchase* will be represented by the same symbol, the system gains much of the power of distributional semantics.

The thesis also describes a series of developments to this model, addressing ambiguity, one-way implicative relations, implicative verb constructions, and a cross-lingual generalization. These models show strong performance on a range of natural language inference tasks, such as question answering and entailment.

1.2.2 Outline of Thesis

The rest of this thesis proceeds as follows:

Chapter 2 provides a survey of the current state of the start in computational semantics, which contains a huge range of approaches. It focuses on formal and distributional models of semantics, arguing that they have attractive properties that other approaches do not.

Chapter 3 describes the theory and first implementation of Natural Semantics, a model of formal semantics that gives a sophisticated treatment of phenomena such as negation and quantification. It describes how to create a lexicon for such a system in a way that gives high coverage of natural language text. I also introduce an algorithm for converting the semantic representation to standard first-order logic, and then the implementation is evaluated on a dataset of inference problems, showing the ability to reason about quantifiers in a sophisticated way.

Chapter 4 introduces the key idea of this thesis, which extends the model of Chapter 3 by modelling the meanings of content words with symbols derived from a distributional clustering. These symbols can seamlessly be integrated into the lexicon. A simple initial model is developed, and then it is refined to show how to model ambiguous words. The new model shows high performance on a question-answering task, due to the strength of the clustering, without affecting the accuracy of the model of formal semantics developed in Chapter 3.

Chapter 5 demonstrates how the model of Chapter 4 can be generalized cross-lingually, by clustering words in different languages on the basis on named-entity arguments. The resulting clustering can be viewed as a simple interlingua. The model outperforms a state-of-the-art model of machine translation on question answering and translation reranking tasks, despite requiring no parallel text for training.

Chapter 6 extends the model of Chapter 4 to model a greater range of semantic phenomena. I show how a more sophisticated approach to clustering can be used to learn richer lexical entries, which support inferences between words that only one in one direction. The model of formal semantics is also extended with modal logic operators, allowing it to better model the meaning of implicative verbs. These improvements show how advanced ideas from both the formal and distributional semantics literatures can be easily incorporated into the framework. Both are shown to lead to improvements on an entailment task compared with the model of Chapter 4 and a variety of existing approaches.

I also give a detailed discussion of the potential for future work in this framework. I identify a number of weakness of the current model, and suggest how they could be overcome to give a major step forward in automated natural language understanding.

Chapter 7 summarizes the key ideas developed in this thesis.

CHAPTER 2

Related Work

2.1 Introduction

Natural language semantics is a huge and diverse field, complicated by a range of theoretical frameworks, numerous potential applications, and the competing tensions of pragmatic short-term applications and long-term ambitions.

This thesis aims to be a step towards solving natural language inference problems in the long-term, and I will discuss the related work from this viewpoint.

- First I will discuss traditional logical approaches, based on linguistic theories of formal semantics.
- Then I will describe a variety of supervised approaches to semantics, which aim to reproduce annotated corpora. I will argue that the ontologies these annotations are based on are always likely to be incomplete.
- Finally, I will discuss distributional semantics, which has been successful in capturing some aspects of meaning, but fails to model the meaning of many function words. I will conclude by discussing some recent work that uses distributional statistics in a compositional symbolic framework.

2.2 Logical Semantics

2.2.1 Formal Semantic Approaches

Early work on computational semantics focused on building models of linguistic theories of formal semantics, which aim to compositionally combine logical forms representing the meanings of words onto logical forms capturing the meanings of sentences. Formal semantics has been highly successful theoretically at explaining many linguistic phenomena, including negation, quantification, plurality, anaphora, modality, tense, and aspect. However, despite many attempts, it has fallen out of favour as a method of modelling semantics. I briefly summarise two recent attempts.

The XLE system developed at Xerox PARC used a large hand-built lexical functional grammar for syntactic and semantic parsing [Bobrow et al., 2007]. A large lexicon is used, based both on existing annotations such as WordNet and VerbNet, and extensions to deal with deverbal nouns, implicative verbs, light verbs. Both syntactic and semantic ambiguity are handled using packed logical forms (as opposed to statistical disambiguation models), and an inference algorithm is used that reasons directly with packed logical forms.

Boxer [Bos, 2008] was an important breakthrough in this field, as it was able to build logical forms for sentences with high coverage without extensive grammar engineering. The system is based on CCG, which has a very strong link between the syntactic type and the semantic type of words. Wide-coverage CCG syntactic parsers already exist, such as the C&C parser [Clark and Curran, 2004], making it relatively straightforward to generate semantic interpretations for words. Hand-built lexical entries are supplied for function words like *not* and *every*. Discourse Representation Theory is used as a semantic formalism. Additional inference rules are added using resources such as WordNet, and inference can be performed using first-order theorem proving.

Both these approaches achieve high-precision on a textual entailment task, but recall is very low—largely because of the weak model of lexical semantics provided by the ontologies. This problem is certainly not through lack of effort—the XLE team had expert linguists, and seemingly devoted large resources over many years in developing their system—making it attractive to search for an alternative to further manual effort in ontology construction.

2.2.2 Natural Logic

MacCartney and Manning [2007] introduced a *natural logic* approach to interpretation, which maps sentences to polarity-annotated strings. Polarity is either positive, negative, or non-monotone, and encodes whether a word is in the scope of negation. A sentence will entail another if its positively polarised words are replaced with more general expressions, or its negatively polarized words are replaced with more specific ones. For example *Some farmers don't own any donkeys→Some people don't own any fat donkeys* because *farmers* is positively polarized and *donkeys* is negatively polarized.

Hand-built lexical entries for function words encode information about the polarity of the word's arguments. For example, *most* is non-monotone on its first argument, and upward-monotone on its second argument, so *most birds fly→most birds move*. They combine these polarities using a syntactic parse tree, to produce a polarity for each word in the sentence. Inference on these annotated strings is a series of atomic edits which transform the premise into the hypothesis, whilst keeping track of whether the edited sentence is inferred by the original. These edits can be efficiently computed using an edit-distance algorithm. Resources such as WordNet [Miller, 1995] are used to model lexical semantics. They show excellent performance on a dataset that emphasises a variety of complex linguistic phenomena, and match the precision of Bos and Markert [2005] with much higher recall. However, natural logic has a much weaker proof theory than first order logic, and is unable to handle inferences involving multiple sentences, make entailments where words are re-ordered, or model logical relations such as De Morgan's Laws.

2.3 Supervised Semantics

Much work on semantics has taken place in a supervised setting, where sentences are paired with some gold standard meaning representation, and systems learn to map between them. There are two challenges here—defining and annotating the gold standard representation, and learning the mapping. These approaches can broadly be divided into domain-specific database querying tasks, and broad-coverage semantic annotation projects such as OntoNotes [Hovy et al., 2006] and FrameNet [Baker et al., 1998]. All of these approaches require a predefined set of predicates which is used to annotate language. I will argue that such any manually constructed ontology is likely to be incomplete, limiting the effectiveness of supervised approaches, and motivating unsupervised distributional approaches to semantics.

2.3.1 Domain Specific

A large body of work, often called *semantic parsing*, has tackled the problem of mapping natural language questions on to database queries. Commonly used databases include Geoquery [Zelle and Mooney, 1996], ATIS [Dahl et al., 1994] and (more recently) Freebase [Bollacker et al., 2008]. This task is clearly useful, as it allows natural language interfaces to existing manually constructed databases. Freebase is by far the largest example of such a database, containing 2.4 billion facts[1], and will contain the answers to many common questions. However, even this is far too small from the point of view of wide coverage natural language understanding—Riedel et al. [2013] notes that its ontology cannot express such high-frequency predicates as *criticize*.

2.3.2 Wide Coverage

Alternative approaches have attempted to annotate all sentences in a corpus with semantic representations.

The PropBank and NomBank projects have annotated argument taking nouns and verbs respectively in the Wall Street Journal with predicate-argument structure [Kingsbury and Palmer, 2002, Meyers et al., 2004]. The annotations abstracts away from different syntactic realisations of arguments. For example, both *Shakespeare wrote Macbeth* and *Macbeth was written by Shakespeare* would be annotated as having the same semantic representation. OntoNotes extends this with other corpora, and maps words to their senses in WordNet. FrameNet goes a step further than VerbNet, by grouping predicates that have the same semantic arguments, even if they are realized differently syntactically. For example, *Shakespeare wrote Macbeth* and *Shakespeare is the author of Macbeth* would evoke the *text creation* frame, with *author* and *text* arguments.

Much work has also been done on automatically learning to map text onto these representations, including the fields of word-sense disambiguation [Navigli, 2009], frame-semantic parsing [Das et al., 2013] and semantic role labelling [Gildea and Jurafsky, 2002].

Creating such ontologies as WordNet, VerbNet and FrameNet is highly expensive. However, even using these resources, the problem of lexical semantic inference is far

[1] As of February 2014

from solved. Although they have been successfully used in downstream semantic tasks such as entailment and question-answering, state-of-the-art models incorporate many other sources of information [Hickl et al., 2006]. In contrast, parsers trained on the Penn Treebank [Marcus et al., 1993] (or derived Treebanks such as CCGBank [Hockenmaier and Steedman, 2007]), are normally sufficient when a syntactic analysis of a sentence is required.

Much of the problem lies with the difficulty of creating ontologies. WordNet contains over 200,000 lexical entries, but does not contain the information necessary for simple inferences such as *author of* $\Longrightarrow$ *wrote*. Senses are also notoriously fine-grained, and hard to annotate [Hovy et al., 2006]. Conversely, FrameNet representations can be overly coarse-grained for inference. For example *fry*, *bake*, and *boil* all evoke the same frame, but we would not want to infer that *John baked a cake*$\rightarrow$*John boiled a cake*. VerbNet also contains interesting logical form annotations. However, as with FrameNet, the predicates are too coarse-grained to support question-answering applications (e.g. *write*, *invent*, *mass-produce* and *synthesise* are given the same semantics). The ultimate difficulty is that language is extremely productive, and express a vast range of meanings, with subtle and complex relationships between the meaning of one word and others—capturing all this in an ontology is extremely challenging.

In conclusion, there have been many large-scale expensive attempts to construct ontologies for expressing the meaning of natural language. However, none of these ontologies has proved sufficient for capturing the diversity and subtlety of meaning that languages express, and consequently are insufficient for wide-coverage natural language understanding. These limitations motivate the use of distributional semantics, in an attempt to capture to learn a better representation than can be created by hand.

2.4 Distributional Semantics

Distributional semantics encompasses a wide range of approaches to modelling meaning. The key distinction is that methods attempt to learn the meanings of expressions from unlabelled corpora, rather than relying on existing ontologies. The methods assume the Distributional Hypothesis, which states that words with similar meanings should occur in similar contexts [Harris, 1954].

I broadly divide these approaches into ones in which context vectors are built for words and then composed to interpret longer expressions (Section 2.4.1), and ones where statistics are used to judge similarity between longer non-compositional patterns

(Section 2.4.2).

2.4.1 Vector Space Approaches

Vector space models of semantics have attracted a very large number of recent papers—see Baroni et al. [2013] for an overview. Section 2.4.1.1 discusses how words are represented in vector spaces, Section 2.4.1.2 describes how word vectors can be composed into vectors representing phrases and sentences, and Section 2.4.1.3 discusses attempts to model logical operators such as negation and quantification in vector spaces.

2.4.1.1 Distributional Models of Word Meaning

Vector space models of semantics represent the meaning of linguistic expressions as vectors. The standard approach is to create a vector space with a dimension for each of the N most common words in the corpus. Then, a vector is created representing each word, based on the context it occurs in—the entries in the vector are the counts of the corresponding context words. There are many possible versions of this approach, including methods for normalizing the dimensions, the size of the context window, the number of dimensions, the use of tensors and matrices instead of vectors, using syntactically connected words as the context, performing dimensionality reduction on the vectors, etc. A more recent alternative is to use auto-encoders to learn vector space embeddings for words using neural-network language models [Bengio et al., 2006].

It is assumed that the similarity of words in a vector space (e.g. using cosine or Euclidean distance to measure similarity) corresponds to semantic similarity. For example, *dog* may be closer to *cat* than to *television*, as the former two will share more contexts. Systems such as that of Huang et al. [2012] achieve good correlation with humans on word similarity evaluations.

There is much evidence that such representations of words are useful (particularly those of neural language models). For example, they have been used as features in supervised natural language processing tasks such as POS-tagging, named-entity recognition, noun-phrase chunking, CCG supertagging, dependency parsing and semantic role labelling [Turian et al., 2010, Collobert et al., 2011, Lewis and Steedman, 2014b]. In such tasks, rather than using purely unsupervised vectors, the vectors are typically 'fine-tuned' during supervised-training—by back-propagating errors into the vectors.

However, it is still unclear how sufficient vector space representations are for fully modelling word meaning. For example, antonyms are typically found to be much closer to their opposites than to unrelated words, as they often share similar contexts [Padó and Lapata, 2003]. A single concept of distance in vector space seems to be insufficient for representing the relation between antonyms. However, there has been promising recent work in this regard from Socher et al. [2013], who train a neural network for classifying WordNet relations between words, based on their vector representations.

2.4.1.2 Compositionality

Given that vector space representations of words have been shown to be useful, there has been interest in creating vector space representations of longer expressions and sentences.

The first approach to composing word vectors was Mitchell and Lapata [2008], who proposed addition and multiplication as composition functions. The functions allow both the words being combined to contribute to the meaning of the full expression. Both these functions are associative and commutative, meaning word-order is ignored, and all bracketings of the words are equivalent. Consequently, *Frogs eat herons* and *Herons eat frogs* will have the same representation. Despite their simplicity, they have shown good performance compared to much sophisticated models [Blacoe and Lapata, 2012]. Baroni et al. [2013] argues that this combination of simplicity and performance makes them "undoubtedly the best current choice for practical applications".

Other approaches have taken syntax into account, for example Coecke et al. [2010] and Grefenstette et al. [2011] represent words as tensors depending on their semantic type, and then use tensor products to combine them into the meanings of sentences. For example, nouns can be modelled as vectors, and adjectives as matrices—so then an adjective is a function from a noun to a noun. This maintains the close link between syntax and semantics that is a key attraction of categorial grammar. One disadvantage of this approach is that is can require extremely high order tensors to represent some categories. For example, syntactic categories such as $((N/N)/(N/N))/((N/N)/(N/N))$ are common in long noun-compounds, which requires an 8th order tensor for representation. Even if the dimension for nouns is only 50, such words would be represented with almost 40 quadrillion parameters.

Socher et al. [2011, 2012] model composition of word vectors using recursive autoencoders. Autoencoders are neural networks which aim to reconstruct their input after first compressing it to a lower dimensional hidden layer via a function f. For example, given vectors for two consecutive expressions $a, b \in \mathbb{R}^n$, the autoencoder aims to learn a matrix $W \in \mathbb{R}^{n\times 2n}$ such that $f(a,b) = g(W(a;b))$, where g is a function such as *tanh* that introduces non-linearity. Crucially, the dimension of the representation of the pair of expressions $f(a,b)$ is then the same as the dimension of the representation of the individual expressions—meaning that f can be applied recursively. The choice of which order to combine expressions can either be based on a syntactic parse [Socher et al., 2012] or choosing the combination with the minimum reconstruction error [Socher et al., 2011]. The use of autoencoders means that the matrix W can be learnt in a fully unsupervised way, to minimize the reconstruction error at every node in every tree in the corpus. Socher et al. [2012] extends this model with the MV-RNN, which represents words as a pair of its vector and a matrix representing its operator semantics—so a different matrix is used in each composition. A softmax layer can be added to the output layer, allowing supervised data to fine-tune the representations.

However, despite the large amount of work on this problem, there is relatively little evidence that vector representations of sentences support the kinds of inference required for tasks such as question-answering. Evaluations typically focus on very short expressions, such modelling the meaning of verb-object or adjective-noun combinations [Mitchell and Lapata, 2008]. Socher et al. [2011, 2012] show that such vectors can be combined with labelled data to perform tasks such as predicting sentiment, and detecting instances of a small number of relations. However, as discussed in Section 2.3 it seems unlikely we will ever have adequate hand-built representations that capture all aspects of meaning.

Baroni et al. [2013] make the following argument: "once you assume that words have distributional representations, it is hard to avoid the conclusion that phrases and sentences have distributional representations too". However, as discussed in the previous section, there are few current models attempting to model aspects of word meaning which are more complex than *similarity*. This problem becomes far worse when modelling the meaning of sentences, where *similarity* is rarely the most useful or meaningful metric. Applications such as question answering instead require *entailment*. Even if sentences could be modelled successfully as vectors, many important inferences require understanding longer texts, such as documents or whole encyclopedias—which are likely to bring their own problems for vector-space composition.

2.4.1.3 Attempts to Model Logic in Vector Spaces

A major motivation for the development of formal semantics was the apparent need to represent the meanings of words such as *every*, *not* and *or*. These words have obvious interpretations when meanings are represented in first order logics, but it is rather less clear what their interpretation should be in a vector space. I briefly review several attempts to tackle this problem.

Grefenstette [2013] show how to hand build tensor representations of function words that simulate logical connectives. They assume a vector-space which is very different from standard distributional approaches. In domains with n objects, objects are represented as one-hot vectors in $\{0,1\}^n$, $true$ and $false$ are represented as $(1,0)^T$ and $(0,1)^T$, and predicates are the $2 \times n$ matrices that map objects onto either $true$ or $false$. Simple 2×2 matrices can be defined that simulate logical negation and connectives. It is not possible to represent quantifiers in this way, so an alternative predicate representation is defined. Sets are represented by $\{0,1\}^n$ vectors where object k is in the set S iff $S_k = 1$. Then, predicates p are redefined to be functions that map sets to subsets, which can be modelled with diagonal matrices M_p. Existential quantification of the form $\exists x[p(x) \wedge q(x)]$ is then a function that checks whether M_p and M_q have a non-zero intersection. Similarly, $\forall x[p(x) \implies q(x)]$ can be modelled with a function that checks if the intersection of M_p and M_p is equal to M_p (i.e. the extension of p is a subset of the extension of q).

There are a number of challenges facing applying Grefenstette [2013]'s model to text. In particular, the logical operators used assume object and predicate representations which are quite unlike those which have been learnt in an unsupervised way (which is the motivation for using vector spaces in the first place). The dimension of the space, and hence the size of the predicate representations, also grows linearly in the number of objects in world, which may prove problematic at scale.

Socher et al. [2012] show that the MV-RNN model learns to model interesting non-Boolean cases of negation involving adjectives—for example learning that *not great* does not mean *terrible*. They also show that with one-dimensional vectors, the model can learn function representations of words that model propositional logic connectives. Of course, the one-dimensional case is very different to the kinds of vectors that are normally used to represent content words, and it is yet to be shown how to generalise this approach to deal with both logical negation and distributional representations of content words. Hermann et al. [2013] argue that the MV-RNN model allows negation to take too broad a scope, and introduces an extension which limits how far the functional representation of a word can propagate.

Baroni et al. [2012] implement the only system I am aware of that attempts to model the meanings of quantifiers in vector spaces. They build vectors for pairs of quantifiers and nouns, and then try to classify the validity of inferences such as *all dogs→some dogs*. The classifier is trained based on labelled examples of entailment with other quantifiers, such as *every dog→many dogs*. The experiments show that the extracted vectors do contain some information about the meanings of the quantifiers. As in other work modelling logic in vector spaces, they do not test the interaction with the semantics of content words—for example *every animal→many dogs*. Of course many properties of quantifiers are untested, such as monotonicity, scoping, and effect on the verb phrase. It would be interesting to know if the model could be scaled up to learn examples like *Every person danced→All girls moved*. Capetola (2013) argues such inferences may be possible in vector spaces, but argues inferences like *every dog barks→Fido barks* require model-theoretic semantics.

2.4.1.4 Conclusion

Vector space models of meaning remain a very active area of research, with much progress made in recent years. However, there are many outstanding challenges. To my knowledge, no research has yet demonstrated a vector-space model in which logical aspects of function words interact with distributional representations of content words. There are also unsolved problems in compositionality, and most research still concentrates on modelling the meanings of short phrases. There is limited evidence so far that vector representations of sentences support the kind of inference needed for tasks such as question answering, although they have proved very useful on tasks such as detecting sentiment. Future work may well make progress with these challenges, but they suggest that symbolic meaning representations are still worth pursuing.

2.4.2 Pattern-based Approaches

Given the difficulty of compositionality in distributional semantics, an alternative approach is to build non-compositional models of longer expressions. Distributional similarity of these expressions is used to create *inference rules*, which determine whether one can be substituted for another.

There are several possible ways to define longer expressions, the most commonly used representations are Reverb patterns [Fader et al., 2011], and dependency paths [Lin and Pantel, 2001]. Reverb patterns are are short sequences of words connecting two noun phrases, filtered by POS-tag. Whilst many common expressions can be captured using these patterns, it will not find relations between noun-phrases that are separated by more than a few words, as it does not use syntax. Dependency paths are the fragment of a dependency tree connecting two noun phrases. These patterns can capture long-range dependencies, but will drop modifiers, such as adverbs, determiners and negation. For example, the sentences *Every American supports Obama* and *Most Americans don't support Obama* contain the same dependency path between *American* and *Obama*.

Statistics can then be gathered on the two noun-phrases arguments of these patterns. The seminal DIRT system [Lin and Pantel, 2001] represented a pattern with two vectors, containing the arguments of each of its slots in a large corpus. For example, the *X wrote Y* pattern and the *X is the author of Y* pattern may have similar nouns instantiating X and Y, providing evidence they are semantically similar. The idea here is closely related to other distributional semantic approaches; similarity in vector space is intended to correspond to semantic similarity. To compare one pattern with another, the average similarity of the argument vectors was computed using a information-theoretic metric. Many other possible metrics have subsequently been proposed [Weeds and Weir, 2003, Kotlerman et al., 2010].

Pantel et al. [2007] made an important contribution to this area, by noticing that the inherent ambiguity in such inference rules could be resolved by adding types to arguments. For example, the verb means something quite different in *charging a criminal* and *charging a battery*, but knowing that the objects are different kinds of thing suggests the verb means something different in each case. Schoenmackers et al. [2010] used an alternative model of types based on Hearst patterns [Hearst and Schütze, 1996], whilst Yao et al. [2011] treated types as latent variables in a topic model, and Yao et al. [2012]'s model learns types using agglomerative clustering. Rather than assume that a single type fully disambiguates relations, Melamud et al. [2013] built a distribution over types, and marginalised this distribution out during inference.

Berant et al. [2011] further developed these ideas by building *entailment graphs*. An entailment graph contains a directed edge between every pair of predicates where an inference is predicted to hold. The key observation is that entailment is a transitive relation, so the entailment graph must be closed under transitivity—greatly limiting the possible graphs. Learning entailment graphs is therefore a constrained optimisation problem, where the objective is maximizing the probability of the edges in the graph, whilst respecting the transitivity constraint.

Riedel et al. [2013] introduced a novel related approach. This model builds a matrix in which rows correspond to pairs of entities, columns correspond to predicates, and entries indicate the probability of a relation holding between the pair of entities. The matrix is initially populated with directly observed predicates, and then an unsupervised model is used to complete it based on correlations between the extensions of predicates. This approach has several advantages compared to previous work. When judging the truth of a statement, such as *Google bought YouTube*, it can take into account all the relations in the corpus observed between the entities—whereas inference-rule approaches only make pairwise decisions like *Google purchased YouTube→Google bought YouTube*. It also integrates seamlessly with existing knowledge bases, as their relations can be added as predicates—making the mapping from textual relations to knowledge-base schemas straightforward in both directions. Whilst this model has clear practical applications, there are limitations. Modelling logical concepts such as negation, quantification, disjunction and modality may prove difficult in this matrix framework. The textual patterns used are non-compositional, so the model would have to learn the relation between *buy* and *did not buy* based on distributional statistics.

2.4.3 Compositional Symbolic Approaches

2.4.3.1 Unsupervised Semantic Parsing

Unsupervised Semantic Parsing [Poon and Domingos, 2009, 2010, Titov and Klementiev, 2011] is an important recent development in semantics.

USP maps dependency parsed sentences to logical forms where the symbols are cluster identifiers. Every word is assigned to exactly one cluster. For example a *buying* cluster may contain verbs such as *buy*, *purchase* and *acquire*. The cluster also contains a set of roles, which will be represented as a distribution over dependencies. The role corresponding to the purchaser may be likely to be realised by dependencies such as *nsubj* or *agent*. Another role may correspond to the seller, and be realised by the dependency *from*. These clusters can be learnt in an unsupervised way, as similar predicates are likely to have similar arguments. Poon and Domingos [2010] extend this model by learning hierarchies of predicates.

There are several limitations of current models of USP. The clustering is computationally expensive, and can only be run on small datasets such as the 20,000 sentence Genia corpus [Kim et al., 2003]. Ambiguous predicates are not modelled—a limitation that is perhaps not exposed more because of the relatively specific biomedical domain the approach is tested on. Whilst the model could be extended to deal with this weakness, there would be a corresponding computational overhead. The models assume that the predicate and dependencies realising a role are conditionally independent given a cluster. This assumption may mean it has limitations with clustering predicates which express the same meaning using different dependencies, such as *buy* and *sell*. Current work also makes no attempt to model function words, such as negatives and quantifiers.

2.4.3.2 Distributional Logical Axioms

Garrette et al. [2011], Beltagy et al. [2013] introduce a method for improving the performance of logic-based systems, by adding distributionally induced inference rules as logical axioms.

Garrette et al. [2011] judge the probability that a WordNet-derived inference rule is valid in a given context, based on the similarity of the vectors representing the words. This softens deterministic WordNet rules, by making them probabilistic—and hence is aimed at improving precision, rather than recall. Beltagy et al. [2013] extend this work by creating axioms between all pairs of words. They also create axioms between multi-word items, based on compositional vector space similarity.

This strand of research is in a similar spirit to that developed in this thesis; a comparison between the approaches is given in Section 4.6.4.

ness, there would be a need [illegible]. The [illegible] is that the predicates and arguments [illegible] are conditionally independent given a cluster. This assumption may [illegible] which [illegible] such as [illegible]. Current work [illegible] to model [illegible], such as negatives and quantifiers.

2.4.5.2 Distributional Logical Axioms

Garrette et al. [2011], Beltagy et al. [2013] introduce a method for improving the performance of logic-based systems, by adding distributionally induced inference rules as logical axioms.

Garrette et al. [2011] judge the probability that a WordNet-derived inference rule is valid in a given context, based on the similarity of the vectors representing the words. This softens deterministic WordNet rules, by making them probabilistic—and hence is aimed at improving precision, rather than recall. Beltagy et al. [2013] extend this work by creating axioms between all pairs of words. They also create axioms between multi-word items, based on compositional vector space similarity.

This strand of research is in a similar spirit to that developed in this thesis, a comparison between the approaches is given in Section 4.6.4.

CHAPTER 3

A Computational Model of Natural Semantics

3.1 Introduction

This chapter develops the first wide-coverage implementation of the theory of *natural semantics*, a CCG-based approach to semantics that is effective at modelling quantifier and negation scope [Steedman, 2012]. This implementation provides the backbone for the distributional semantics extensions that will be developed in the rest of this thesis. The description of CCG will focus on the theory as it is currently implemented in treebanks, parsers, and semantic analysis tools, which differs somewhat from textbook treatments.

The goals of this chapter are to:

- Build first-order logical forms for open domain text with high-coverage.
- Model the underlying predicate argument structure, i.e. to identify which objects participate in which relations. CCG allows us to handle a variety of linguistically complex constructions.
- Model the meaning of function words, particularly quantifiers and negatives, to allow the system to make powerful logical inferences.

- To verify the correctness of Steedman [2012]'s theory of natural semantics, particularly the mechanisms for representing scope ambiguities.

However, in this chapter I do not attempt a model of lexical semantics that goes beyond using the word itself as a symbol.

Section 3.2 offers a brief introduction to the CCG theory of syntax and semantics.

Section 3.3 sketches a more sophisticated theory of semantics, based on that of Steedman [2012], that is used in this thesis.

Section 3.4 describes how to build a wide-coverage implementation of this theory, by showing how to create lexical entries for words based on their syntactic category.

Section 3.5 gives an algorithm for converting the natural semantics representation into first-order logic, which can be used in standard theorem provers.

Section 3.6 evaluates the implementation. I show it has high-coverage, with valid output for 99.6% of sentences. Investigating the output shows that it successfully models complex syntactic constructions and scope ambiguities. I also evaluate the quality of these logical forms on the FraCaS suite, showing that they are capable of sophisticated multi-sentence inferences involving quantifiers.

Section 3.7 discusses the limitations of the system, and future directions for computational models of formal semantics.

The FraCaS evaluation has previously been published in Lewis and Steedman [2013a].

3.2 Combinatory Categorial Grammar

Combinatory Categorial Grammar [Steedman, 2000, 2012] is a strongly lexicalized theory of language, in which (almost) all the decisions made during syntactic and semantic parsing are assignments of definitions to words. During parsing, each word is first assigned a *lexical entry*. A lexical entry is a triple of a word, its syntactic category, and its semantic interpretation, denoted:

word $\vdash$ category : *interpretation*

For example, the following lexical entry asserts that *Shakespeare* can be a noun-phrase, interpreted as a *shakespeare* symbol.

Shakespeare $\vdash$ NP : *shakespeare*

Many words act as functions. Their lexical entries have syntactic categories that are functions from one category to another, and interpretations that take logical forms

as arguments using lambda calculus. For example, the following lexical entry says that the transitive verb *wrote* is syntactically a function from two noun-phrases to a syntactic sentence, and semantically a function from two entities to a predicate on events[1] (representing a semantic sentence).
wrote $\vdash (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP} : \lambda x\lambda y\lambda e.write(y,x,e)$

A small set of combinators define how categories and interpretations can combine, of which function application is by far the most common. The same combinator applies to both the syntax and semantics. The process that combines the meanings of all the words in a sentence is called a *derivation*, for example:

$$
\begin{array}{ccc}
\text{Shakespeare} & \text{wrote} & \text{Macbeth} \\
\hline
NP & (S\backslash NP)/NP & NP \\
shakespeare & \lambda y\lambda x\lambda e.write(x,y,e) & macbeth \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP \quad >} \\
 & \multicolumn{2}{c}{\lambda x.write(x,macbeth,e)} \\
\hline
\multicolumn{3}{c}{S \quad <} \\
\multicolumn{3}{c}{write(shakespeare,macbeth,e)}
\end{array}
$$

3.2.1 Syntactic Categories

CCG categories are either ground or functional.

Ground categories include S (sentence), NP (noun phrase), PP (prepositional phrase) and N (noun). Ground categories may subcategorize with agreement features. For example PP_{in} refers to a prepositional phrase headed by *in*, and S_{pss} refers to a passive-voice sentence.

Functional categories can be constructed as functions between other categories. If X and Y are categories then X/Y and $X\backslash Y$ are functions that return the category X if applied to the argument Y. Backward slashes require that the argument occurs to the left of the function in the sentence, and forward slashes require that it occurs to the right.

Each category has a semantic type, which are e, t, ev (entities, truth values, and events), or functions between these. The types of function categories can easily be constructed from the types of ground categories.

Some example categories and their semantic types are given in Table 3.1.

[1]Event variables [Davidson, 1967] are useful semantically for analysing constructions such as adverbs. Sometimes they will be omitted from derivations for brevity, but the implementation uses event variables for analysing all verbs and argument taking nouns.

Category	Description	Semantic Type
N	Noun	$<e,t>$
NP	Noun Phrase	$<e>$
PP	Prepositional Phrase	$<e>$
PR	Phrasal Verb Particle	$<e>$
S	Sentence	$<ev,t>$
$S\backslash NP$	Intransitive Verb	$<e,<ev,t>>$
$(S\backslash NP)/NP$	Transitive Verb	$<e,<e,<ev,t>>>$
N/PP	Argument-taking Noun	$<e,<e,t>>$
N_i/N_i	Adjective	$<<e,t>,<e,t>>$
$(S\backslash NP)\backslash(S\backslash NP)$	Adverb	$<<e,<ev,t>>,<e,<ev,t>>>$

Table 3.1: Some example CCG categories, and their semantic types. In this interpretation, sentences are viewed as predicates on events.

3.2.2 Semantic Interpretations

As well as a syntactic category, each lexical entry also provides a *semantic interpretation*. In this work, I will express interpretations using lambda-calculus and a first-order logic, but other representations are possible. The crucial point is that *the syntactic category and semantic interpretation of a word must have the same semantic type*. For example the transitive verb *loves* has the category $(S\backslash NP)/NP$, so its semantic interpretation must have type $<e,<e,<ev,t>>>$. One interpretation meeting this restriction is: $\lambda x \lambda y \lambda e.love(x,y,e)$. Note that because the verb takes two noun-phrase arguments in the syntax, it must also take two entity arguments in the semantics.

3.2.3 Combinatory Rules

A small set of binary combinators is used, that can combine two categories X and Y to a category Z, and performs a corresponding operation on the semantics. The combinators guarantee that the syntactic and semantic types will match for the result of the combination, if they did for each of the arguments. The combinatory rules used are listed in Table 3.2.

Rule	Left	Right	Result	Symbol
Forward Application	$X/Y : \lambda y.f(y)$	$Y : y$	$X : f(y)$	$>$
Backward Application	$Y : y$	$X\backslash Y : \lambda y.f(y)$	$X : f(x)$	$<$
Forward Composition	$X/Y : \lambda y.f(y)$	$Y/Z : \lambda z.g(z)$	$X/Z : \lambda z.f(g(z))$	$> B$
Backward Crossed Composition	$X/Y : \lambda y.f(y)$	$Y\backslash Z : \lambda z.g(z)$	$X/Z : \lambda z.f(g(z))$	$> B_X$
Forward Substitution	$(X/Y)/Z : \lambda z\lambda y.f(y,z)$	$Y/Z : \lambda z.g(z)$	$X/Z : \lambda z.f(g(z),z)$	$> S$
Backwards Crossed Substitution	$Y/Z : \lambda z.g(z)$	$(X\|Y)/Z : \lambda z\lambda y.f(y,z)$	$X/Z : \lambda z.f(g(z),z)$	$< S_X$
Forward 2-Composition	$X/Y : \lambda y.f(y)$	$(Y/Z)/W : \lambda w\lambda z.g(z,w)$	$(X/Z)/W : \lambda w\lambda z.f(g(z,w))$	$> B^2$

Table 3.2: Example CCG Combinatory rules. See the derivations throughout this chapter for example instantiations of these rules.

3.2.4 Unary Rules

A small number of unary rules are also used, which convert one category to another. I use a similar set of rules to those implemented by the C&C parser [Clark and Curran, 2004], and extend them with semantic interpretations[2]. Again, these rules perform operations on the syntax and semantic in tandem, and guarantee that the category and interpretation of the result will have the same semantic type.

For example, in reduced relatives such as *the boy playing football*, a verb-phrase acts as a post-modifier on a noun:
$S\backslash NP : \lambda x \lambda e.p(x,e) \rightarrow N\backslash N : \lambda q \lambda x.\exists e[q(x) \wedge p(x,e)]$

3.2.5 Type Raising

Type-raising a category converts an argument into a function over functions. If X is a ground category, then $Y/(Y\backslash X)$ and $Y\backslash(Y/X)$ are its type-raised forms. For example, the following is a type-raised lexical entry:
Shakespeare $\vdash$ S/(S\NP) : $\lambda p.p(shakespeare)$ This lexical entry defines Shakespeare as a function from verb-phrases to sentences, and can be used in the subject position. Similarly, noun phrases occuring in the object position can use lexical entries such as the following, where the category is a function from a transitive verb to a verb phrase:
Macbeth $\vdash$ (S\NP)\((S\NP)/NP) : $\lambda p.p(macbeth)$

Type-raised categories can be space-consuming and difficult to read, so I will normally abbreviate them as $NP^{\uparrow}$.

Here is a type-raised derivation for *Shakespeare wrote Macbeth*:

$$\begin{array}{ccc}
\text{Shakespeare} & \text{wrote} & \text{Macbeth} \\
\hline
S/(S\backslash NP) & (S\backslash NP)/NP & (S\backslash NP)\backslash((S\backslash NP)/NP) \\
\lambda p.p(shakespeare) & \lambda y \lambda x.write(x,y) & \lambda p.p(macbeth) \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP \quad <} \\
 & \multicolumn{2}{c}{\lambda x.write(x,macbeth)} \\
\hline
\multicolumn{3}{c}{S \quad >} \\
\multicolumn{3}{c}{write(shakespeare,macbeth)}
\end{array}$$

In most existing work, e.g. CCGBank [Hockenmaier and Steedman, 2007] and derived parsers, type-raising is implemented as a unary rule. In contrast, the categories

[2] One exception is that I handle type-raising in the lexicon, as discussed in 3.4.3.1. I also use an option on the C&C parser to disable 'noisy' rules, which eases semantic interpretation, at small cost of coverage.

NP, *PP* and *PR* are always type-raised in the lexicon in theoretical treatments of CCG. My system implements type-raising in the lexicon—to my knowledge, it is the first to do this, and to show that lexicalized type-raising is possible for practical wide-coveage parsing.

There are several reasons for type-raising all such categories:

- Most importantly for this thesis, type-raising allows us to give accurate semantic interpretations to generalized quantifiers. The most natural interpretation for *every* is $\lambda p \lambda q \forall x.p(x) \rightarrow q(x)$. However, this has the semantic type $<< e,t >,<< e,t >,t >>$, which is not type-transparent to the none-type-raised determiner category NP/N (of semantic type $< e,< e,t >>$). Schematized determiner categories, such as $NP^{\uparrow}/N$ can be used instead.

  every $\vdash$ (S/(S\NP))/N : $\lambda p \lambda q \forall x.p(x) \rightarrow q(x)$

- It naturally explains case-marking on noun-phrases. For example, the fact that *I* is used in the subject and *me* in the object is explained by the lexicon containing entries such as:

  I $\vdash$ S/(S\NP) : $\lambda p.p(me)$
  me $\vdash$ (S\NP)\((S\NP)/NP) : $\lambda p.p(me)$

  This distinction is particularly important in languages which make more use of case-marking, such as Hindi.

- Typed-raised derivations allow left-branching derivations, which support incremental interpretation, which be useful for tasks such as language modelling. For example:

$$
\begin{array}{cccl}
\text{Shakespeare} & \text{wrote} & \text{Macbeth} & \\
\cline{1-1}\cline{2-2}\cline{3-3}
S/(S\backslash NP) & (S\backslash NP)/NP & S\backslash(S/NP) & \\
\lambda p.p(shakespeare) & \lambda y \lambda x.write(x,y) & \lambda p.p(macbeth) & \\
\cline{1-2}
\multicolumn{2}{c}{S\backslash NP} & & {>}\mathbf{B} \\
\multicolumn{2}{c}{\lambda y.write(shakespeare,y)} & & \\
\cline{1-3}
\multicolumn{3}{c}{S} & {<} \\
\multicolumn{3}{c}{write(shakespeare,macbeth)} &
\end{array}
$$

  Note that this derivation produces the same logical form as the right-branching version.

Boxer [Bos, 2008], another wide-coverage CCG semantic parser, takes an alternative approach. Their system allows non-typeraised *NP* categories to have the type-raised semantics, and making corresponding changes to the interpretations of words taking *NP* arguments so that semantically they take function arguments. For example, the intransitive verb *sleep* would have the interpretation $\lambda p.p(\lambda x.sleep(x))$. I believe that syntactically type-raising *NP*s in the syntax is a better solution, as it means words can have simpler and more intuitive interpretations (at the price of more complex syntactic categories).

3.3 Natural Semantics for CCG

The next section outlines the theory of Natural Semantics proposed by Steedman [2012], which is used in this thesis. The major change over the semantics outlined so far is that existential quantifiers are replaced with *generalized Skolem terms*. This change is intended to simplify reasoning about the scope of quantifiers and negation. Steedman argues that this semantics is 'natural' because of the transparent interface between syntax and semantics—which means that the approach fits into the natural logic tradition (starting with Aristotle) that attempts to define a logic which matches the grammar of natural language.

3.3.1 Quantifier Scope

Quantifiers are determiners that express how many entities are in a relation. Examples include *a*, *some*, *every*, *most*, *no*, *more than three* etc. Determining which quantifiers have scope over which others is a major issue in building logical forms and interpreting language. For example, the sentence *Every man loves a woman* has two interpretations—one where there is a woman that all men love, and one where each man may love a different woman— corresponding to the following two logical forms:

(1)
a. $\exists w[woman(w) \wedge \forall m[man(m) \Rightarrow love(m,w)]]$
b. $\forall m[man(m) \Rightarrow \exists w[woman(w) \wedge love(m,w)]]$

In the former interpretation, the existential is said to take *wide scope*, and in the latter it takes *narrow scope*. The latter reading is called the *surface scope* reading, as the ordering of the quantifiers in the sentence is the same as in the logical form—the former called the *inverse scope* reading.

Determining the set of possible interpretations can be surprisingly complex. For example, Geach [1973] points out that the sentence *Every boy admires and every girl detests some saxophonist* appears to have exactly two readings—the saxophonist must either be wide-scope, or narrow-scope with respect to both boys and girls. There appears to be no reading which means that all girls detest the same saxophonist, but all boys may admire different ones.

3.3.1.1 Existing Work on Processing Scope

There have been many attempts in the linguistics literature to deal with such problems. I briefly sketch several important approaches.

Perhaps the most obvious idea is to allow the syntax of the derivation to determine which quantifiers scope over which others. For example, the following derivation gives the narrow-scope reading:

$$
\begin{array}{ccc}
\text{Every man} & \text{loves} & \text{a woman} \\
\hline
NP^{\uparrow} & (S\backslash NP)/NP & NP^{\uparrow} \\
\lambda p.\forall m[man(m) \Rightarrow p(m)] & \lambda y \lambda x.love(x,y) & \lambda q.\exists w[woman(w) \wedge q(w)] \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP \quad <} \\
 & \multicolumn{2}{c}{\lambda x.\exists w[woman(w) \wedge love(x,w)]} \\
\hline
\multicolumn{3}{c}{S \quad >} \\
\multicolumn{3}{c}{\forall m[man(m) \Rightarrow \exists w[woman(w) \wedge love(m,w)]]}
\end{array}
$$

The corresponding wide-scope logical form can be derived with a left-branching derivation:

$$
\begin{array}{ccc}
\text{Every man} & \text{loves} & \text{a woman} \\
\hline
NP^{\uparrow} & (S\backslash NP)/NP & NP^{\uparrow} \\
\lambda p.\forall m[man(m) \Rightarrow p(m)] & \lambda y \lambda x.love(x,y) & \lambda q.\exists w[woman(w) \wedge q(w)] \\
\cline{1-2}
\multicolumn{2}{c}{S/NP \quad >\mathbf{B}} & \\
\multicolumn{2}{c}{\lambda y.\forall m[man(m) \Rightarrow love(m,y)]} & \\
\hline
\multicolumn{3}{c}{S \quad >} \\
\multicolumn{3}{c}{\exists w[woman(w) \wedge \forall m[man(m) \Rightarrow love(m,w)]]}
\end{array}
$$

However this method only allows the sentence *Every boy admires and every girl detests some saxophonist* to have a single (wide scope) reading, as the right-node-raising forces a left-branching derivation. It also means that purely semantic scope ambiguities have to be reflected in different syntactic parses, and would require richer syntactic treebanks that make this distinction.

Montague [1973] uses a non-monotonic *quantifying in* operation to model scope ambiguities. Very loosely, this approach interprets the sentence with the quantifiers replaced by pronouns, and then the quantifying-in operation is used to substitute the actual semantics of the quantifier into the interpretation. For example, it can derive the wide-scope reading by first considering the meaning of *He loves some woman*, and then using the quantifying-in operation to replace the symbol for *he* with that of *every man*.

Cooper Storage [Cooper, 1975, 1983] builds an *underspecified logical form* from a syntactic parse, which is accompanied by a store of quantifiers. The example sentence might parse to $love(M,W)$, where M and W index a store $< M = \lambda p.\forall m[man(m) \rightarrow q(m)], W = \lambda q.\forall w[woman(w) \rightarrow q(w)] >$. The values of M and W can be substituted in from the store to create a fully-specified interpretation. Applying M then W gives $\exists w[woman(w) \wedge \forall m[man(m) \Rightarrow love(m,w)]]$, and applying W then M yields $\forall m[man(m) \Rightarrow \exists w[woman(w) \wedge love(m,w)]]$.

Both quantifying-in and Cooper Storage can overgenerate readings, for example it is easy to see how they derive 4 interpretations of the *saxophonist* sentence. These approaches all also generate semantically spurious equivalent readings. The sentence *A man loves a woman* has no scope ambiguities, however the methods suggested so far will generate two different (but equivalent) logical forms:

(2)
a. $\exists w[woman(w) \wedge \exists m[man(m) \wedge love(m,w)]]$
b. $\exists m[man(m) \wedge \exists w[woman(w) \wedge love(m,w)]]$

In the worst case, the number of interpretations is the factorial of the number of quantifiers. For example Koller and Thater [2006] note that the English Resource Grammar [Flickinger, 2000] generates 3960 readings for the sentence *For travellers going to Finnmark there is a bus service from Oslo to Altara through Sweden*—all of which are semantically equivalent re-orderings of existential quantifiers.

3.3.1.2 Generalized Skolem Terms

Steedman [2012] proposes replacing almost all generalized quantifiers with *generalised Skolem terms*, in order to simplify reasoning about quantifier scope.

Singular generalised Skolem terms represent entities, and carry the following information:

- *Restrictor condition*—a predicate on the entity the Skolem term represents. For example, the restrictor condition for the noun-phrase *a fat farmer* would be $\lambda x.farmer(x) \wedge fat(x)$.
- *Scope*—the set of universally quantified variables that the Skolem term is a function of.
- *Polarity*—either positive, negative or unspecified (marked $+, -, \circ$), which determines whether the Skolem term is in the scope of a negation operator (discussed more in Section 3.3.2).

- *Identifier*—an identifier, allowing coreference between multiple Skolem terms referring to the same noun-phrase.

For example, the Skolem term $+sk^{(y,z)}_{27:\lambda x.farmer(x)}$ represents an entity that is a farmer, which is not negated, has arbitrary unique identifier 27, and is scoped by two universally quantified variables y and z (as in one reading for *Every student knows that every donkey is owned by some farmer*). The unique identifiers can be automatically assigned.

Plural Skolem terms represent sets. Here, the restrictor condition is a predicate on every member of the set. These also have a *cardinality condition*, which is a predicate on the cardinality of the set. For example, the wide-scope interpretation of the NP *More than 3 farmers* would be represented as: $+sk^{()}_{91:\lambda x.farmer(x)\,;\,\lambda s.|s|\geq 3}$

To simplify derivations, normally some of these features will be suppressed. For example, the identifier will not be shown when no other term shares the identifier, and the polarity will not be shown in sentences with no negation.

3.3.1.3 True Universals

The use of Skolem terms means that no entities are represented with existential quantifiers[3]. However, a small number of determiners introduce universal quantifiers, such as *each* and *every*:

every $\vdash \mathrm{NP}^{\uparrow} : \lambda p \lambda q.\forall x[p(x) \rightarrow q(x)]$

These are treated as being *true universals*, and have different properties to other determiners—for example:

- True universals can invert scope. *An Englishman won every gold medal* is ambiguous between the wide-scope reading (with a single Englishman) and the narrow-scope reading (with multiple Englishmen). On the other hand, *An Englishman won three gold medals* can only be interpreted as having a single Englishman.

- When conjoined, true universals take singular agreement. For example, *Every boy and every girl is dancing* vs. *A boy and a girl are dancing*.

[3]For simplicity, events are represented with existentially quantified variables. However, we could go further, and represent events with Skolem terms to reason about whether *Three boys watched Macbeth* refers to a single event or three separate watchings.

- True universals are not compatible with collective verbs, such as gather. For example, *The players gathered on the pitch* vs **Every player gathered on the pitch*

3.3.1.4 Processing Scope

The scope of Skolem terms is initially underspecified—meaning the number of quantifiers it is bound by is not determined. However, at any point in the parse a specification operation may take place, in which the scope is fixed to be the set of enclosing universal quantifiers in the logical form.

For example, in the sentence *Every man loves a woman*, the Skolem term representing *a woman* may specify either at the start of the derivation (and hence take wide scope), or at the end of the parse (and take narrow scope). The two readings are:

(3)
a. $\forall m[man(m) \Rightarrow man(m, sk^{()}_{\lambda w.woman(w)})]$
b. $\forall m[man(m) \Rightarrow man(m, sk^{(m)}_{\lambda w.woman(w)})]$

A key advantage of using Generalised Skolem Terms is that logical forms with different scopes are still structurally homomorphic—i.e. they are identical except for the scope of the two Skolem terms. In contrast, using existential quantifiers the semantics will be structurally different:

(4)
a. $\forall w[woman(w) \Rightarrow \exists m[man(m) \wedge love(m,w)]]$
b. $\forall m[man(m) \Rightarrow \exists w[woman(w) \wedge love(m,w)]]$

The fact that the Skolemized logical forms are structurally homomorphic allows both readings to be stored in a single shared structure:

(5) $\forall m[man(m) \Rightarrow love(m, \left\{ \begin{array}{c} sk^{()} \\ sk^{(m)} \end{array} \right\} \lambda w.woman(w))]$

Curly brackets here represent disjunctive packing of the logical form—cf. Maxwell and Kaplan [1995]; Crouch [2005].

It is straightforward to derive such structures in a derivation: whenever a universal quantifier takes scope over a new Skolem term, a new possible interpretation of that Skolem term is created. For example:

$$\dfrac{\text{Every man}}{\begin{array}{c} NP^{\uparrow} \\ \lambda p.\forall m[man(m) \Rightarrow p(m)] \end{array}} \quad \dfrac{\dfrac{\text{loves}}{\begin{array}{c}(S\backslash NP)/NP \\ \lambda y\lambda x.love(x,y)\end{array}} \quad \dfrac{\text{a woman}}{\begin{array}{c}NP^{\uparrow} \\ \lambda q.q(sk^{()}_{\lambda w.woman(w)})]\end{array}}}{\begin{array}{c}S\backslash NP \\ \lambda x.love(x, sk^{()}_{\lambda w.woman(w)})]\end{array}}<$$

$$\overline{\begin{array}{c} S \\ \forall m[man(m) \Rightarrow love(m, \left\{ \begin{array}{c} sk^{()} \\ sk^{(m)} \end{array} \right\} \lambda w.woman(w))] \end{array}}>$$

Another key advantage is that logical form for *A man loves a woman* is unambiguous, despite containing multiple quantifiers:
$love(man(sk^{()}_{\lambda m.woman(m)}), sk^{()}_{\lambda w.woman(w)})$
This approach largely avoids the problems of underspecification, where sentences with n quantifiers can have $n!$ spuriously equivalent readings.

Note that if the same Skolem term occurs multiple times in a logical form, only readings where each instance of that Skolem is scoped by the same number of quantifiers are valid. For example, the packed logical form for *Every boy admires and every girl detests some saxophonist* is:
$\forall x[boy(x) \Rightarrow admire(x, \left\{ \begin{array}{c} sk^{()}_{35} \\ sk^{(x)}_{35} \end{array} \right\}_{\lambda y.sax(y)})] \wedge \forall z[girl(z) \Rightarrow detest(z, \left\{ \begin{array}{c} sk^{()}_{35} \\ sk^{(z)}_{35} \end{array} \right\}_{\lambda y.sax(y)})]$ As before, the 35 identifier indicates that the Skolem terms all refer to the same noun-phrase. The wide-scope reading follows from taking the first interpretation in the list for each instance of sk_{35}:
$\forall x[boy(x) \Rightarrow admire(x, sk^{()}_{35:\lambda y.sax(y)})] \wedge \forall z[girl(z) \Rightarrow detest(z, sk^{()}_{35:\lambda y.sax(y)})]$

The narrow scope reading is the second entry in the list for each instance of sk_{35}:
$\forall x[boy(x) \Rightarrow admire(x, sk^{(x)}_{35:\lambda y.sax(y)})] \wedge \forall z[girl(z) \Rightarrow detest(z, sk^{(z)}_{35:\lambda y.sax(y)})]$

In conclusion, the method means that a single packed logical form can be created from a single syntactic parse sentence that captures the genuine ambiguities, while not generating spuriously equivalent logical forms.

3.3.2 Negation Scope

Negation scope is also crucial for correctly building accurate logical forms that support the correct inferences. For example *Obama didn't visit any EU countries*$\rightarrow$*Obama didn't visit Britain*, but *Obama didn't visit some EU countries*$\not\rightarrow$*Obama didn't visit Britain*.

If a noun-phrase is in the scope of negation, it is said to be *negatively polarized*, and otherwise it is *positively polarized*. During inference, positively polarized noun-phrases can be replaced with more general ones, whereas negatively polarized noun-phrases can be replaced with more specific ones. For example, the sentence *Some farmer owns no animal→Some person owns no donkey* because *farmer* is positively polarised and *donkey* is negatively polarised.

Polarity marking was used successfully by MacCartney and Manning [2007] to draw such inferences—however we take the more general approach of doing this at the level of logical form, rather than in syntax trees. To implement this in the semantics, Generalised Skolem terms are marked with ± markers. For example:

(6)
a. Some farmer owns some donkey
   $own(+sk_{farmer}, +sk_{donkey})$
b. Some farmer doesn't own some donkey
   $\neg own(+sk_{farmer}, +sk_{donkey})$
c. Some farmer owns no donkey
   $\neg own(+sk_{farmer}, -sk_{donkey})$
d. No farmer owns some donkey
   $\neg own(-sk_{farmer}, +sk_{donkey})$
e. No farmer owns any donkey
   $\neg own(-sk_{farmer}, -sk_{donkey})$

Lexically, Skolem terms may have fixed positive or negative polarity, or the unspecified polarity $\circ$ (slightly simplifying Steedman [2012], who also has inverting and non-monotone polarities). If the polarity of a Skolem term is unspecified, it takes polarity from its environment (i.e. negative if it is in the scope of negation, or positive otherwise). Certain determiners fix the polarity of the Skolem term in the lexicon, e.g.:
some $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(+sk_{\lambda x.p(x)})$
any $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(-sk_{\lambda x.p(x)})$
a $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(\circ sk_{\lambda x.p(x)})$

Constants representing named entities always have fixed positive polarity.

For example, in the following, the determiner *some* ensures that *exam* is positively polarized, despite the negation.

$$
\begin{array}{ccc}
\text{I didn't pass} & \text{some} & \text{exam} \\
\hline
S_{dcl}/NP & NP^{\uparrow}/N & N \\
\lambda x.\neg pass(+i,x) & \lambda p\lambda q.q(+sk_{\lambda x.p(x)}) & \lambda x.exam(x) \\
\cline{2-3}
& \multicolumn{2}{c}{NP^{\uparrow}} \\
& \multicolumn{2}{c}{\lambda p.p(+sk_{\lambda x.exam(x)})} \\
\hline
\multicolumn{3}{c}{S_{dcl}} \\
\multicolumn{3}{c}{\neg pass(+i,+sk_{\lambda x.exam(x)})}
\end{array}
$$

However, in the following, the determiner *an* leaves the polarity of *exam* unspecified, so it takes negative polarity when it falls in the scope of negation.

$$
\begin{array}{ccc}
\text{I didn't pass} & \text{an} & \text{exam} \\
\hline
S_{dcl}/NP & NP^{\uparrow}/N & N \\
\lambda x.\neg pass(+i,x) & \lambda p\lambda q.q(\circ sk_{\lambda x.p(x)}) & \lambda x.exam(x) \\
\cline{2-3}
& \multicolumn{2}{c}{NP^{\uparrow}} \\
& \multicolumn{2}{c}{\lambda p.p(\circ sk_{\lambda x.exam(x)})} \\
\hline
\multicolumn{3}{c}{S_{dcl}} \\
\multicolumn{3}{c}{\neg pass(+i,-sk_{\lambda x.exam(x)})}
\end{array}
$$

This reading is equivalent to that for *I didn't pass any exam*. The sentence also has a positive polarity interpretation (in which *an exam* does refer to a particular exam), where the Skolem term specifies early in the positive environment. An obvious extension would be to also build a packed logical form capture negation scope ambiguities, but I do not explore that here.

Steedman [2012] also uses polarity in the syntax, for example to disallow sentences such as **Some farmer owns any donkey*. The present implementation only uses polarity in the semantics.

3.3.3 Correction to Natural Semantics

As a consequence of implementing the theory, I discovered that Steedman [2012]'s description of Skolem specification overgenerates readings. Problems can occur when Skolem terms are nested inside others (as happens in relative clauses), because the two terms are allowed to choose their scope independently. When the nested term takes narrower scope than its parent, the resulting semantics does not have an interpretation.

For example, in *Every man loves a woman who read a book*, the theory allows the *woman* Skolem term to specify early, taking wide-scope, and the *book* Skolem term to specify late and be bound to the variable representing *man*. In this reading, every man loves the same woman, but she must have read one book per man.

Similarly, in *John doesn't love some woman who read a book*, it is possible for the *book* Skolem term to specify late in the scope of negation, and be negatively polarized, whilst the *woman* Skolem term is positively polarized. However, *book* is clearly positively polarized—the sentence does not entail *John doesn't love some woman who read a book by Tolstoy*.

The problem in both cases is that nested Skolem terms are taking narrower scope than their parent. This problem can be solved by not allowing Skolem terms to be bound by negatives or quantifiers outside their parents, unless their parents are also in that scope—which can be implemented by making the specification operation also force the specification of all nested Skolem terms.

Unpacking the packed logical forms is then slightly more complex, as there are dependencies between Skolem terms. A restriction has to be added so that nested Skolem terms must be in the scope of all variables that their parents are in.

For example, the sentence *Every man loves a woman who read a book* yields the following packed logical form :

$$\forall m[man(m) \Rightarrow love(m, \left\{ \begin{array}{c} sk^{()} \\ sk^{(m)} \end{array} \right\} \lambda w.woman(w) \wedge read(w, \left\{ \begin{array}{c} sk^{()} \\ sk^{(m)} \end{array} \right\} \lambda b.book(b)))$$

The new restriction on unpacking means that this subsumes three logical forms, not the four predicted by the original theory. The incorrect blocked reading is:
$\forall m[man(m) \Rightarrow love(m, sk^{()}_{\lambda w.woman(w) \wedge love(w, sk^{(m)}_{\lambda b.book(b)})})]$

While the required correction is relatively minor, I believe this emphasises the importance of building testable computational implementations of linguistic theories to validate their correctness.

3.4 Adding Natural Semantics to CCG Syntactic Parsers

Building wide coverage semantic parsers for CCG is relatively straightforward. First, a syntactic parse from a CCGBank-trained parser can be used to assign categories to each word and determine the combinatory rules used [4]. Then, the semantic parser must assign semantics to each word, based on its syntactic category, to build a complete lexical entry. As long as the choice of lexical entry means that interpretation of the word

[4]The C&C parser implements a subset of the rules and categories in CCGBank (based on frequency cutoffs), which greatly simplifies semantic interpretation. CCGBank contains a large number of rare categories and rules, many of which were the result of noise in the conversion process from CCGBank.

has the same semantic type as its syntactic category, it is guaranteed that the interpretation will be compatible throughout the derivation. For example, it will not try to apply an expression expecting an entity argument to an event. This is a key advantage of the close link between syntax and semantics in CCG, and greatly simplifies the process of creating wide-coverage logical forms.

Of course, the fact that it is easy to produce logical forms does not guarantee that they are useful—which must be validated experimentally.

3.4.1 Automatic Lexicon for Content Words

It would not be possible to enumerate all possible $(word, category)$ pairs in advance that might need a semantic interpretation, so the semantic interpretations must be generated 'on the fly'. In this section, I describe an algorithm for doing this.

The simplest approach would be to manually write a template semantic interpretation for each possible category, which optionally make use of a special *LEMMA* symbol which is to be instantiated with the lemma of the word. Boxer [Bos, 2008] does this. However, the system developed here uses a much larger set of lexical categories—due to lexicalizing type-raising, and a different analysis of prepositions introduced in Chapter 4. Instead, template interpretations can be created for many categories automatically, based on the interpretations of simpler categories.

Below, I discuss how lexical entries are assigned in a number of important constructions. First, I explain how to assign semantics to some simple categories, and then show how to build the semantics of more complex categories recursively.

3.4.1.1 Simple Categories

First, I manually create templates for straightforward base cases:
Common nouns: **LEMMA** $\vdash \mathrm{N} : \lambda x.LEMMA(x)$
Named Entities: **LEMMA** $\vdash \mathrm{NP}^{\uparrow} : \lambda p.p(+LEMMA)$
Prepositions: **LEMMA** $\vdash \mathrm{PP}^{\uparrow}/\mathrm{NP} : \lambda x \lambda p.p(x)$
Intransitive verbs: **LEMMA** $\vdash \mathrm{S}\backslash\mathrm{NP} : \lambda x \lambda e.LEMMA(e) \wedge arg0(x,e)$

Categories of the form X_i/X_i and $X_i\backslash X_i$ are modifiers, e.g. adjectives and adverbs. The interpretations of these re-states the semantics of the expression they are modifying, and then adds their lemma as an additional predicate. For example:
LEMMA $\vdash \mathrm{N/N} : \lambda p \lambda x.p(x) \wedge LEMMA(x)$
LEMMA $\vdash (\mathrm{S}\backslash\mathrm{NP})\backslash(\mathrm{S}\backslash\mathrm{NP}) : \lambda p \lambda x \lambda e.p(x,e) \wedge LEMMA(e)$

3.4.1.2 Entity Arguments

This section explains how arguments with the categories *NP* and *PP* are modelled, which is the most common way arguments are expressed.

Noun-phrase arguments are labelled with a number in the style of PropBank [Kingsbury and Palmer, 2002], such as *arg*0, *arg*1 etc. By default, noun-phrase arguments are counted from the left, in the order they appear in the sentence. For example, the following template is generated for ditransitives:

LEMMA $\vdash ((\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP})/\mathrm{NP} : \lambda x \lambda y \lambda z \lambda e.LEMMA(e) \wedge arg0(z,e) \wedge arg1(x,e) \wedge arg2(y,e)$

Prepositional phrase arguments are labelled based on the preposition, for example:

LEMMA $\vdash ((\mathrm{S}\backslash\mathrm{NP})/\mathrm{PP}_{\mathrm{to}})/\mathrm{PP}_{\mathrm{from}} : \lambda x \lambda y \lambda z \lambda e.LEMMA(e) \wedge arg0(z,e) \wedge from(x,e) \wedge to(y,e)$

I add a special case for passive constructions, in order to give the same analysis for *Shakespeare wrote Macbeth* and *Macbeth was written by Shakespeare*. For passive verbs (headed with the S_{pss} category), the argument keys of noun phrases are incremented by 1, and any argument supplied by PP_{by} is given the *arg*0 key. For example:

LEMMA $\vdash (\mathrm{S}_{\mathrm{pss}}\backslash\mathrm{NP})/\mathrm{PP}_{\mathrm{by}} : \lambda x \lambda y \lambda e.LEMMA(e) \wedge arg0(y,e) \wedge arg1(x,e)$

Genitives can introduce arguments to nouns. In Honnibal et al. [2010]'s version of CCGBank, they are analysed with an additional PP argument on the noun. Such arguments are given an arbitrary *arg* interpretation. For example, the following template would be used for *gift* in *Rome's gift of peace to Europe*):

LEMMA $\vdash ((\mathrm{N}/\mathrm{PP})/\mathrm{PP}_{\mathrm{to}})/\mathrm{PP}_{\mathrm{of}} : \lambda x \lambda y \lambda z \lambda e.LEMMA(e) \wedge arg(z,e) \wedge of(x,e) \wedge to(y,e)$

This approach means that rather than enumerating every verb category in advance, as was done by Boxer [Bos, 2008], the system can generate them automatically.

3.4.1.3 Verb-Particle Constructions

The semantics of verb particle constructions often bears little resemblance to the semantics of the verb or particle, for example: *take up*, *take on* and *take over*. For this reason, I choose to treat these as non-compositional, and implement this by appending the particle to the main predicate. The verb then discards its particle argument. For example, *take* in *He took over the world*, would use the template:

LEMMA $\vdash ((\mathrm{S}\backslash\mathrm{NP})/\mathrm{PR}_{\mathrm{over}})/\mathrm{NP} : \lambda x \lambda y \lambda z \lambda e.LEMMA\_over(e) \wedge arg0(y,e) \wedge arg1(z,e)$

3.4.1.4 Predicative Complements

Many categories take predicative complements. For example, control verbs such as *ask* and *promise* take verb-phrase arguments. For the semantic analysis to be valid, these argument functions must themselves be supplied the correct arguments—for example, a verb-phrase argument with category $S\backslash NP$ expects an entity and event argument.

The CCGBank co-indexing on categories is helpful here. If parts of a category are co-indexed, they refer to the same object—which allows arguments to themselves take arguments (non-coindexed arguments of arguments can be supplied with existential quantifiers). This allows us to create the subject-control template, where the subject is the argument of the verb-phrase complement:

LEMMA $\vdash ((\mathrm{S_{dcl}}\backslash \mathrm{NP_i})/(\mathrm{S_{to}}\backslash \mathrm{NP_i}))/\mathrm{NP} : \lambda x \lambda p \lambda y \lambda e.LEMMA(e) \wedge arg0(y,e) \wedge arg1(x,e) \wedge \exists e'[p(y,e')]$ The corresponding object-control template is:

LEMMA $\vdash ((\mathrm{S_{dcl}}\backslash \mathrm{NP})/(\mathrm{S_{to}}\backslash \mathrm{NP_i}))/\mathrm{NP_i} : \lambda x \lambda p \lambda y \lambda e.LEMMA(e) \wedge arg0(y,e) \wedge arg1(x,e) \wedge \exists e'[p(x,e')]$

3.4.1.5 Function Words

Lexical entries can also be generated for many function words, based on their syntactic category. Function words can be identified based on POS-tag, and are given a logical form that mirrors their syntactic category, without introducing a new predicate. For example, by using the argument co-indexation the system can automatically generate lexical entries such as the following, which is used by relative pronouns such as *that* and *which*:

LEMMA $\vdash (\mathrm{N_i}\backslash \mathrm{N_i})/(\mathrm{S}\backslash \mathrm{NP_i}) : \lambda p \lambda q \lambda x.q(x) \wedge \exists e[p(x,e)]$

I also add default templates for categories such as determiners and conjunctions, though the function words lexicon will override most cases of these:

LEMMA $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(sk_{\lambda x.p(x)})$

LEMMA $\vdash (\mathrm{X}\backslash \mathrm{X})/\mathrm{X} : \lambda p \lambda q \lambda \ldots.p(\ldots) \wedge q(\ldots)$

3.4.2 Hand-built Lexicon for Function Words

The lexicon from Section 3.4.1 can be extended with a hand-built lexicon of function words. I believe this is a good pragmatic choice, as there are a relatively small number of function words, whose semantics can be quite complex. The semantics of such words have also seen attention in the linguistics literature, making it straightforward to utilise ideas. I am not aware of any unsupervised NLP work that has learned the semantics of complex function words.

There have also been attempts to learn the meanings of function words from distributional statistics—see Section 2.4.1 for some discussion.

The universal quantifiers *each* and *every* have the following semantics:

each $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.\forall x[p(x) \rightarrow q(x)]$

every $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.\forall x[p(x) \rightarrow q(x)]$

All is given a different semantics. The fact that a universal quantifier is not used in its definition means that it cannot invert scope.

all $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(all_{\lambda x.p(x)})$

The determiner *Some* ensures that the introduced Skolem term is positively polarized, so will not support downward-monotone inferences. For example, *I don't own some donkeys* cannot be interpreted as meaning *I don't own any donkeys.*

some $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(+sk_{\lambda x.p(x)})$

The determiner *a* takes polarity from its environment:

a $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(\circ sk_{\lambda x.p(x)})$

Steedman [2012] defines *any* as forcing its noun-phrase to take negative polarity. When implemented, I found this reduced coverage, as often the positively polarized *free-choice any* is used, as in *Any farmer who owns a donkey feeds it.* As the current model of syntax does not mark polarity, the system cannot distinguish these cases. Instead, I give the same semantics as *a*, and allow:

any $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(\circ sk_{\lambda x.p(x)})$

The determiner *no* is often given the semantics $\lambda p \lambda q.\forall x[p(x) \rightarrow \neg q(x)]$. However, it does not invert scope: *Some farmer owns no donkey* does not seem to have an inverse-scope reading, which would be equivalent to *No donkey is owned by every farmer.* Instead, the definition uses a negatively polarized Skolem term:

no $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.\neg q(-sk_{\lambda x.p(x)}]$

Not negates its verb-phrase argument:

not $\vdash (\mathrm{S_{dcl}}\backslash\mathrm{NP})/(\mathrm{S_b}\backslash\mathrm{NP}) : \lambda p \lambda x \lambda e.\neg p(x,e)$

n't $\vdash (\mathrm{S_{dcl}}\backslash\mathrm{NP})/(\mathrm{S_b}\backslash\mathrm{NP}) : \lambda p \lambda x \lambda e.\neg p(x,e)$

Lexical entries for numbers can be generated 'on the fly', by using a simple algorithm for parsing string representations of numbers. For example:

three $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(sk_{\lambda x.p(x)\,;\,\lambda s.|s|=3})$

Some quantifiers are themselves compositional, for example *at least five* or *fewer than three.* Rather than attempt to handle these in a fully compositional way, I defined

a short list of numeric modifiers (such as *at least* and *fewer than*) and their effect on the cardinality. Then, the system can automatically generate lexical entries such as:
at least three $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(sk_{\lambda x.p(x)\,;\,\lambda s.|s| \geq 3})$

Upper bounds are modelled by negating lower bounds, so *Fewer than three farmers walk* means that there is no set of three farmers, all of whom walk:
fewer than three $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.\neg q(-sk_{\lambda x.p(x)\,;\,\lambda s.|s| \geq 3})$

Some other multiword quantifiers are handled non-compositionally, for example:
at least a few $\vdash \mathrm{NP}^{\uparrow}/\mathrm{N} : \lambda p \lambda q.q(sk_{\lambda x.p(x)\,;\,\lambda s.|s| \geq 2})$

CCGBank analyses many determiners as adjectives (i.e. N/N rather than $NP^{\uparrow}$/N), which is inconvenient for the semantics given here. To deal with this problem, any NP node that starts with a determiner (as defined by our lexicon) is automatically converted to the correct analysis.

Chapter 6 extends this lexicon with an account of implicative verbs.

It is worth pointing out that the semantics of many function words is highly context-dependent, and the simple approach described here is insufficient to capture this. For example, the number of individuals quantified by *many* and *few* is highly dependent on the speaker's prior expectations, and does not seem to have truth-conditional boundaries. Universal quantifiers rarely quantify over all individuals, but over some pragmatically relevant subset. A long tradition argues that the semantics of *only* depends on access to a set of alternatives (and is also dependent on focus). Clearly much work remains to be done here—annotated corpora would be particularly useful. However, despite the challenges, formal semantics remains the most successful approach for modelling function words.

3.4.3 Post-processing Syntax Trees

The current output from CCG syntactic parsers is not always in the best form for semantic interpretation. Consequently, the system makes a number of automatic post-processing steps to the syntax before building the semantics.

3.4.3.1 Lexicalizing Type Raising

Existing parsers implement type-raising as a unary rule. As explained in Section 3.2.5, I implement type-raising in the lexicon. A first step towards is to eliminate type-raising unary rules, by pushing them up to the lexical level. For example, the system makes the following conversion:

$$
\begin{array}{ccccl}
\text{Some} & \text{man} & \text{wrote} & \text{Macbeth} & \\
NP/N & N & (S\backslash NP)/NP & NP & \\
\cline{1-2}
\multicolumn{2}{c}{NP} & & & > \\
\cline{1-2}
\multicolumn{2}{c}{S/(S\backslash NP)} & & & >\mathbf{T} \\
\cline{1-3}
\multicolumn{3}{c}{S/NP} & & >\mathbf{B} \\
\cline{1-4}
\multicolumn{4}{c}{S} & >
\end{array}
\quad \rightarrow \quad
\begin{array}{ccccl}
\text{Some} & \text{man} & \text{wrote} & \text{Macbeth} & \\
(S/(S\backslash NP))/N & N & (S\backslash NP)/NP & NP & \\
\cline{1-2}
\multicolumn{2}{c}{S/(S\backslash NP)} & & & > \\
\cline{1-3}
\multicolumn{3}{c}{S/NP} & & >\mathbf{B} \\
\cline{1-4}
\multicolumn{4}{c}{S} & >
\end{array}
$$

3.4.3.2 Type Raising all NP, PP and PR nodes

All categories of the form *NP*\$, *PP*\$ and *PR*\$ categories are replaced with type-raised equivalents (where \$ schematizes over possible arguments). To do this, the system finds function application nodes with arguments of type $X \in \{NP, PP, PR\}$, and updates them with the equivalent type-raised form[5].

For forward application:

$$
\begin{array}{ccl}
Y/X & X & \\
\hline
\multicolumn{2}{c}{Y} & >
\end{array}
\quad \rightarrow \quad
\begin{array}{ccl}
Y/X & Y\backslash(Y/X) & \\
\hline
\multicolumn{2}{c}{Y} & <
\end{array}
$$

For backward application:

$$
\begin{array}{ccl}
X & Y\backslash X & \\
\hline
\multicolumn{2}{c}{Y} & <
\end{array}
\quad \rightarrow \quad
\begin{array}{ccl}
Y/(Y\backslash X) & Y\backslash X & \\
\hline
\multicolumn{2}{c}{Y} & >
\end{array}
$$

Once the category has been updated, the rest of the tree is then updated to account for the change, using *inverse combinators* similarly to Thomforde and Steedman [2011].

3.4.3.3 Named Entities

Named entities can be merged into a single node, by collapsing consecutive words that have the same NER tag. For example, *Barack Obama* is collapsed into *Barack_Obama*. The system makes no other attempt to model compound nouns, although there has been interesting work on modelling their semantics [Tratz and Hovy, 2010].

3.4.3.4 Subcategorize PP and PR categories

All PP and PR categories are automatically subcategorized with the corresponding head preposition. This is useful for our analysis where the preposition is a semanti-

[5] I assume that *NP*, *PP* and *PR* categories will eventually be arguments of a function application combinator, as type-raising blocks composition operators. This causes occasional problems, for example in *preposition stranding* constructions, such as *I live in and like Edinburgh*, where if *in* is type-raised the composition is not possible. Following Steedman and Baldridge [2011], I use the category *PP*/*NP* for the preposition here.

cally transparent case-marker, but causes their predicates to subcategorize for different frames.

Prepositions are treated as being semantically transparent case-markers on nouns—i.e. the semantics of a preposition is the identity function.

For example, *He ran from home* and *He ran to home*, the following lexical entries are used for *ran*, with distinct categories and interpretations[6].

run $\vdash (S\backslash NP)/PP_{\text{from}} : \lambda y \lambda x \lambda e.run(e) \wedge arg0(x,e) \wedge from(y,e)$

run $\vdash (S\backslash NP)/PP_{\text{to}} : \lambda y \lambda x \lambda e.run(e) \wedge arg0(x,e) \wedge to(y,e)$

3.4.3.5 Correcting NP conjunctions

The Rebanked version of CCGBank [Honnibal et al., 2010] contains an error in which *NP* conjunctions are treated as *N* conjunctions. Noun conjunctions are rare, so I automatically correct all such cases to be NP conjunctions. For example, the interpretation of the uncorrected version below would be a single individual who is both *John* and *Mary*:

$$
\begin{array}{cccc}
\text{John} & \text{and} & \text{Mary} & \text{walked}\\
N & conj & N & S\backslash NP\\
\hline
& \multicolumn{2}{c}{N\backslash N \;(>)} & \\
\multicolumn{3}{c}{N \;(<)} & \\
\multicolumn{3}{c}{NP \;(lex)} & \\
\multicolumn{4}{c}{S \;(<)}
\end{array}
\quad \rightarrow \quad
\begin{array}{cccc}
\text{John} & \text{and} & \text{Mary} & \text{walked}\\
N & conj & N & S\backslash NP\\
\hline
NP \;(lex) & & NP \;(lex) & \\
& \multicolumn{2}{c}{NP\backslash NP \;(>)} & \\
\multicolumn{3}{c}{NP \;(<)} & \\
\multicolumn{4}{c}{S \;(<)}
\end{array}
$$

3.5 Conversion to First-Order Logic

The logic used in Steedman [2012] supports inference directly. However, rather than create a new theorem prover for this formalism, I chose to instead convert it to standard first-order logic (for which there are already numerous highly-optimised theorem provers). The conversion process involves replacing the Generalized Skolem Terms with standard first-order quantifiers. It must ensure that the quantifiers are instantiated in the correct scope with respect to negation and other quantifiers.

The conversion closely follows the definition of the model theory for the Skolemized language. Note that some Skolemized logical forms cannot be interpreted by the model theory, and have no translation in first order logic. The conversion algorithm re-

[6]This leads to occasional problems where the original parse coordinates nodes headed by different prepositions, as in *I climbed up the mountain and down the other side.*

lies on the semantic parser to not produce such forms. Therefore, the algorithm given here is not intended to translate arbitrary sentences. In Section 3.6.1 I show empirically that the conversion algorithm interprets almost all sentences produced by the semantic parser.

Before giving the main algorithm, I define a function s, which searches the sub-sentence for Skolem terms with the correct scope, and q, which replaces a given Skolem term in a sentence with a standard first order quantifier.

3.5.1 Finding Skolem terms with a given scope

The function s searches a logical form for instances of Skolem terms with a specified scope and polarity. For example $s(S,\{x,y\},+)$ returns the set of positively polarized Skolem terms in sentence S that are exactly in the scope of x and y (not any superset). $s(S,\emptyset,-)$ returns all the negatively polarized Skolem terms in S.

3.5.2 Replacing Skolem terms with Quantifiers

3.5.2.1 Replacing Singular Skolem Terms

Singular Skolem terms represent entities. They are straightforward to replace, by existentially quantifying a new variable that meets the restrictor condition, and substituting all instances of the Skolem term with the variable:

$q(sk_{n;\lambda x.r(x)},S) = \exists x[r(x) \wedge subst(S,sk_n,x)]$

$subst(S,sk_n,x)$ replaces all Skolem terms with identifier n in S with the variable x.

I also define the corresponding function q' that uses a universal quantifier:

$q'(sk_{n;\lambda x.r(x)},S) = \forall x[r(x) \Rightarrow subst(S,sk_n,x)]$

3.5.2.2 Replacing Plural Skolem Terms

Plural Skolem terms represent sets, and can be translated in a similar way. I focus on simple cardinality conditions of the form $|s| = k$ or $|s| \geq k$ (upper bounds are expressed by negating lower bounds). While the current lexicon ensures all cardinality conditions are of this form, it ignores the difficulties of translating quantifiers such as *most*, fractions, ranges, comparatives, etc.

$q(S,sk_{\lambda x.r(x);\lambda s.|s|\geq k}) = \exists y_0 \ldots \exists y_{k-1}[unique(y_0,\ldots y_{k-1}) \wedge \forall x[(x=y_0 \vee \cdots \vee x=y_{k-1}) \Rightarrow (r(x) \wedge subst(S,sk_n,x))]]$

Where $unique(x_0, \ldots x_{k-1})$ ensures the list of variables contains no duplicates (otherwise, all the existentially quantified variables could refer to the same object).

This function replaces a Skolem term with minimum cardinality k with k unique existentially quantified variables, each of which satisfy both the restrictor condition of the Skolem term and the predicate applied to it.

For example, the following represents q replacing the Skolem term in the interpretation of *At least 2 dogs bark*:
$q(sk_{n;\lambda x.dog(x);\lambda s.|s|\geq 2}, bark(sk_{n;\lambda x.dog(x);\lambda s.|s|\geq 2})) = \exists y_0 \exists y_1 [y_0 \neq y_1 \wedge \forall x[(x = y_0 \vee x = y_1) \Rightarrow (dog(x) \wedge bark(x))]$

The case where the cardinality condition contains equality is similar, except that it uses a biconditional rather than an implication. This implements Steedman [2012]'s *maximal participants condition*, which states that no superset of the Skolem term should both satisfy the restrictor condition and be an argument of the predicate. For example, *Shakespeare wrote 37 plays* is interpreted as meaning that *Shakespeare wrote exactly 37 plays*, and is false in models where he wrote 38.
$q(S, sk_{\lambda x.r(x);\lambda s.|s|=k})$
$\exists y_0 \ldots \exists y_{k-1} [unique(y_0, \ldots y_{k-1}) \wedge \forall x[(x = y_0 \vee \cdots \vee x = y_{k-1}) \iff (r(x) \wedge subst(S, sk_n, x))]]$

In practice, this conversion can lead to logical forms which are intractable for theorem proving. For simplicity, for $k \geq 5$ I use the singular translation with an additional predicate $mod(x) = k$. Of course, we could go further and add arithmetic axioms to the theorem prover, but that is beyond the scope of this thesis.

3.5.3 Main Translation Algorithm

Next, I define the main translation function τ. The algorithm recursively visits sub-sentences of the logical form, maintaining a set X of all universal quantifier variables enclosing the current sub-sentence. At each sub-sentence, it searches for a Skolem term that can be replaced with a first-order quantifier there in the correct scope. If it exists, all instances of the Skolem term are substituted with a new variable, and τ is called recursively on the new sentence. If no Skolem terms can be replaced at that point, τ is called recursively on its sub-sentences.

3.5.3.1 Atomic Sentences

The simplest case is atomic sentences. Either an atomic sentence contains a Skolem term in the current scope that can be translated, or the sentence is returned unchanged:

$$\tau(S,X) = \begin{cases} S, & Y = \{\} \\ \tau(q(k \in Y,S),X), & Y \neq \{\} \end{cases} \textbf{ where } Y = s(S,X,+)$$

For example, to translate the interpretation of *Some man loves Mary* we have:
$\tau(love(+sk_{\lambda x.man(x)},mary)) = \exists x[man(x) \wedge love(x,mary)]$

If there are multiple saturated Skolem terms in the sentence, an arbitrary one is chosen, and the function is recursively called on the result.

3.5.3.2 Negation

At a negated sentence, any *negatively* polarized Skolem terms can be quantified, so that the existential quantifer is in the scope of negation. Note that negatively polarized Skolem terms cannot be bound by universal quantifiers. As all negatively polarized Skolem terms must be in the scope of a negation operator, all other cases only need to translate positive Skolem terms.

$$\tau(\neg S,X) = \begin{cases} \neg\tau(S,X), & Y = \{\} \\ \tau(\neg q(k \in Y,S),X), & Y \neq \{\} \end{cases} \textbf{ where } Y = s(S,X,-)$$

For example, the interpretation of *No man loves Mary* can be translated:
$\tau(\neg love(-sk_{\lambda x.man(x)},mary)) = \neg\exists x[man(x) \wedge love(x,mary)]$

3.5.3.3 Conjunction and Disjunction

In cases of sentences joined by a connective, Skolem terms are only replaced if they are positively polarized and appear on *both* sides of the connective.

$$\tau(S \wedge T,X) = \begin{cases} \tau(S,X) \wedge \tau(T,X), & Y = \{\} \\ \tau(q(k \in Y,S),S \wedge T), & Y \neq \{\} \end{cases} \textbf{ where } Y = s(S,X,+) \cap s(T,X,+)$$

$$\tau(S \vee T,X) = \begin{cases} \tau(S,X) \vee \tau(T,X), & Y = \{\} \\ \tau(q(k \in Y,S),S \vee T), & Y \neq \{\} \end{cases} \textbf{ where } Y = s(S,X,+) \cap s(T,X,+)$$

For example, the interpretation of *Some man loves Jane and Mary* can be translated:
$\tau(love(+sk_{53:\lambda x.man(x)},jane)) \wedge love(+sk_{53:\lambda x.man(x)},mary))$
$= \exists x[man(x) \wedge love(x,jane) \wedge love(x,mary)]$

3.5.3.4 Implicatives

Implicatives act slightly differently, in that if a Skolem term appears on both sides then it is universally quantified.

$$\tau(S \Rightarrow T, X) = \begin{cases} \tau(S,X) \Rightarrow \tau(T,X), & Y = \{\} \\ \tau(q'(k \in Y, S \Rightarrow T), X), & Y \neq \{\} \end{cases} \textbf{ where } Y = s(S,X,+) \cap s(T,X,+)$$

For example, the interpretation of *If a man loves Jane then he loves Mary* can be translated:

$\tau(love(+sk_{53:\lambda x.man(x)}, jane)) \Rightarrow love(+sk_{53:\lambda x.man(x)}, mary))$
$= \forall x[(man(x) \wedge (love(x, jane))) \Rightarrow love(x, mary)]$

3.5.3.5 Quantifiers

Universal quantifiers change the current scope—so any Skolem term inside a quantifier that is not bound by it must be quantified outside the universal:

$$\tau(\forall x[S], X) = \begin{cases} \forall x[\tau(S, X \cup \{x\})], & Y = \{\} \\ \tau(q(k \in Y, \forall x[S]), X), & Y \neq \{\} \end{cases} \textbf{ where } Y = s(S,X,+)$$

For example, the wide-scope interpretation of *Every man loves a woman* can be translated:

$\tau(\forall x[man(x) \Rightarrow love(x, sk^{()}_{\lambda y.woman(y)})], \emptyset) = \exists y[woman(y) \wedge \forall x[man(x) \Rightarrow love(x,y)]]$

However, in the narrow-scope reading, the s function will not return this Skolem term, as it does not match the current scope. x is added to the current scope, and the function is called recursively.

$\tau(\forall x[man(x) \Rightarrow love(x, sk^{(x)}_{\lambda y.woman(y)})], \emptyset)$
$= \forall x[\tau(man(x) \Rightarrow love(x, sk^{(x)}_{\lambda y.woman(y)}), \{x\})]$
$= \forall x[\tau(man(x), \{x\}) \Rightarrow \tau(love(x, sk^{(x)}_{\lambda y.woman(y)}), \{x\})]$
$= \forall x[man(x) \Rightarrow \exists y[woman(y) \wedge love(x,y)]]$

3.5.4 Other

For convenience, I represent the complete set of entities satisfying a predicate p with: $all_{\lambda x.p(x)}$ (this could equivalently be expressed with a Skolem term). Such *all* functions must also be replaced with standard first order quantifers.

The atomic sentence $q(\ldots, all_{\lambda x.p(X)}, \ldots)$ can be replaced with: $\forall x[p(x) \Rightarrow q(\ldots, x, \ldots)]$

3.6 Experiments

In the evaluation, I show that the system allows fast wide-coverage language interpretation, that it is capable of correctly analysing a variety of complex constructions, that it successfully implements packed logical forms for scope ambiguities, and that the logical forms support complex inference.

3.6.1 Coverage

First, I investigate the percentage of sentences for which the system is able to produce logical forms. The C&C parser [Clark and Curran, 2004] is used, with settings disabling 'noisy rules' and 'extra rules'. Parses whose top level category is not S are ignored—the interpretations of sentences with other categories will contain free variables, so do not support inference. Parsed sentences are first converted to the natural semantics representation. Then, the resulting packed logical forms are unpacked, and each interpretation is converted to standard first-order logic. The first-order logical forms are checked to ensure they contain no free variables, that all interpretations contain no Skolem terms or free-variables, and that there are no duplicate logical forms produced. The semantics pipeline is successful for 99.6% of sentences in Section 23 of CCGBank (after development on Sections 02-22).

Excluding the conversion to FOL, the system produces semantic interpretations for 135 sentences per second on a single core on Wall Street Journal text. This compares to 27 sentences per second for syntactic parsing on the same system. This result shows that the semantic interpretation is efficient enough to not be the bottle-neck in processing large corpora.

3.6.2 Qualitative Evaluation

Figures 3.1 to 3.10 show examples of actual system output. These examples demonstrate that the system can build packed logical forms representing scope ambigities, and can model predicate-argument structure across a variety of complex linguistic expressions. Where interesting, I also show the result of the conversion to first-order logic. To save space in the derivations, the semantics of verbs is automatically simplified, so $verb(e) \wedge arg0(x,e) \wedge arg1(y,e)$ is compressed to to $verb(x,y,e)$.

3.6.2.1 Syntactic Variation

$$
\begin{array}{cccc}
\text{A vote} & \text{was} & \text{won} & \text{by Obama} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/(S_{pss}\backslash NP) & (S_{pss}\backslash NP)/PP_{by} & (S_{pss}\backslash NP)\backslash((S_{pss}\backslash NP)/PP_{by}) \\
\lambda p.p(sk_{\lambda x.vote(x)}) & \lambda p\lambda x\lambda e.p(x,e) & \lambda x\lambda y\lambda e.win(x,y,e) & \lambda p.p(Obama) \\
\cline{3-4}
 & & \multicolumn{2}{c}{S_{pss}\backslash NP} \\
 & & \multicolumn{2}{c}{\lambda x\lambda e.win(Obama,x,e)} \\
\cline{2-4}
 & \multicolumn{3}{c}{S_{dcl}\backslash NP} \\
 & \multicolumn{3}{c}{\lambda x\lambda e.win(Obama,x,e)} \\
\hline
\multicolumn{4}{c}{S_{dcl}} \\
\multicolumn{4}{c}{\lambda e.win(Obama,sk_{\lambda x.vote(x)},e)}
\end{array}
$$

Figure 3.1: System output for a passive sentence, showing how it can derive the same logical form as the equivalent active-voice sentence.

$$
\begin{array}{cccc}
\text{man} & \text{who} & \text{wrote} & \text{Macbeth} \\
\hline
N & (N\backslash N)/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/NP & (S_{dcl}\backslash NP)\backslash((S_{dcl}\backslash NP)/NP) \\
\lambda x.man(x) & \lambda p\lambda q\lambda x.q(x)\wedge\exists e[p(x,e)] & \lambda x\lambda y\lambda e.write(y,x,e) & \lambda p.p(Macbeth) \\
\cline{3-4}
 & & \multicolumn{2}{c}{S_{dcl}\backslash NP} \\
 & & \multicolumn{2}{c}{\lambda x\lambda e.write(x,Macbeth,e)} \\
\cline{2-4}
 & \multicolumn{3}{c}{N\backslash N} \\
 & \multicolumn{3}{c}{\lambda p\lambda x.p(x)\wedge\exists e[write(x,Macbeth,e)]} \\
\hline
\multicolumn{4}{c}{N} \\
\multicolumn{4}{c}{\lambda x.man(x)\wedge\exists e[write(x,Macbeth,e)]}
\end{array}
$$

Figure 3.2: System output showing object extraction from a relative clause.

$$
\begin{array}{cccc}
\text{play} & \text{which} & \text{Shakespeare} & \text{wrote} \\
\hline
N & (N\backslash N)/(S_{dcl}/NP) & S_X/(S_X\backslash NP) & (S_{dcl}\backslash NP)/NP \\
\lambda x.play(x) & \lambda p\lambda q\lambda x.q(x)\wedge\exists e[p(x,e)] & \lambda p.p(Shakespeare) & \lambda x\lambda y\lambda e.write(y,x,e) \\
\cline{3-4}
 & & \multicolumn{2}{c}{S_{dcl}/NP \quad {>}\mathbf{B}} \\
 & & \multicolumn{2}{c}{\lambda x\lambda e.write(Shakespeare,x,e)} \\
\cline{2-4}
 & \multicolumn{3}{c}{N\backslash N \quad >} \\
 & \multicolumn{3}{c}{\lambda p\lambda x.p(x)\wedge\exists e[write(Shakespeare,x,e)]} \\
\hline
\multicolumn{4}{c}{N \quad <} \\
\multicolumn{4}{c}{\lambda x.play(x)\wedge\exists e[write(Shakespeare,x,e)]}
\end{array}
$$

Figure 3.3: System output showing subject extraction from a relative clause.

$$
\begin{array}{cccc}
\text{I} & \text{did n't} & \text{pass} & \text{an exam} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/(S_b\backslash NP) \ {>}\mathbf{B} & (S_b\backslash NP)/NP & (S_b\backslash NP)\backslash((S_b\backslash NP)/NP) \ > \\
\lambda p.p(i) & \lambda p\lambda x\lambda e.\neg p(x,e) & \lambda x\lambda y\lambda e.pass(y,x,e) & \lambda p.p(\circ sk_{\lambda x.exam(x)}) \\
\cline{3-4}
 & & \multicolumn{2}{c}{S_b\backslash NP \quad <} \\
 & & \multicolumn{2}{c}{\lambda x\lambda e.pass(x,\circ sk_{\lambda y.exam(y)},e)} \\
\cline{2-4}
 & \multicolumn{3}{c}{S_{dcl}\backslash NP \quad >} \\
 & \multicolumn{3}{c}{\lambda x\lambda e.\neg pass(x,-sk_{\lambda y.exam(y)},e)} \\
\hline
\multicolumn{4}{c}{S_{dcl} \quad >} \\
\multicolumn{4}{c}{\lambda e.\neg pass(i,-sk_{\lambda x.exam(x)},e)} \\
\multicolumn{4}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\
\multicolumn{4}{c}{\neg\exists x[exam(x)\wedge\exists e[pass(i,x,e)]]}
\end{array}
$$

Figure 3.4: *I didn't pass an exam* Example showing how the polarity of a noun can change during the derivation, when used with an unpolarized determiner such as *an*. Our analysis does not capture the positive-polarity interpretation, but in principle it could be extended to build a packed logical form expressing the fact that either polarity is possible.

$$
\begin{array}{cccc}
\text{I} & \text{did n't} & \text{pass} & \text{some exam} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/(S_b\backslash NP) \; {>}\mathbf{B} & (S_b\backslash NP)/NP & (S_b\backslash NP)\backslash((S_b\backslash NP)/NP) \; {>} \\
\lambda p.p(i) & \lambda p\lambda x\lambda e.\neg p(x,e) & \lambda x\lambda y\lambda e.pass(y,x,e) & \lambda p.p(+sk_{\lambda x.exam(x)})
\end{array}
$$

$$
\frac{(S_b\backslash NP)/NP \quad (S_b\backslash NP)\backslash((S_b\backslash NP)/NP)}{\begin{array}{c} S_b\backslash NP \\ \lambda x\lambda e.pass(x,+sk_{\lambda y.exam(y)},e) \end{array}} <
$$

$$
\frac{(S_{dcl}\backslash NP)/(S_b\backslash NP) \quad S_b\backslash NP}{\begin{array}{c} S_{dcl}\backslash NP \\ \lambda x\lambda e.\neg pass(x,+sk_{\lambda y.exam(y)},e) \end{array}} >
$$

$$
\frac{S_{dcl}/(S_{dcl}\backslash NP) \quad S_{dcl}\backslash NP}{\begin{array}{c} S_{dcl} \\ \lambda e.\neg pass(i,+sk_{\lambda x.exam(x)},e) \end{array}} >
$$

$$
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots
$$

$$
\exists x[exam(x) \land \neg\exists e[pass(i,x,e)]]
$$

Figure 3.5: *I didn't pass some exam* This derivation shows how the positively polarized determiner *some* protects its noun from the scope of negation.

$$
\begin{array}{ccc}
\text{At least 2 students} & \text{passed} & \text{an exam} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/NP & (S_{dcl}\backslash NP)\backslash((S_{dcl}\backslash NP)/NP) \; {>} \\
\lambda p.p(+sk_{\lambda x.student(x);\lambda y.|y|>=2}) & \lambda x\lambda y\lambda e.pass(y,x,e) & \lambda p.p(\circ sk_{\lambda x.exam(x)})
\end{array}
$$

$$
\frac{(S_{dcl}\backslash NP)/NP \quad (S_{dcl}\backslash NP)\backslash((S_{dcl}\backslash NP)/NP)}{\begin{array}{c} S_{dcl}\backslash NP \\ \lambda x\lambda e.pass(x,\circ sk_{\lambda y.exam(y)},e) \end{array}} <
$$

$$
\frac{S_{dcl}/(S_{dcl}\backslash NP) \quad S_{dcl}\backslash NP}{\begin{array}{c} S_{dcl} \\ \lambda e.pass(+sk_{\lambda x.student(x);\lambda y.|y|>=2},\circ sk_{\lambda z.exam(z)},e) \end{array}} >
$$

$$
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots
$$

$$
\exists e[\exists x[\exists y[\neg y = x \land \exists z[exam(z) \land \forall u[x = u \lor y = u \implies student(u) \land pass(u,z,e)]]]]]
$$

Figure 3.6: *At least 2 students passed an exam* The system correctly builds a logical form for the wide-scope reading, in which the two students may have passed different exams. However, it fails to predict the reading where all the students passed different exams. To cope with this, the system would need to be extended to mark plurality in the syntax, and then have a separate 'distributive' category for verbs that introduces a universal quantifier (as in Steedman [2012]).

$$
\begin{array}{ccc}
\text{At most 1 student} & \text{passed} & \text{an exam} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/NP & (S_{dcl}\backslash NP)\backslash((S_{dcl}\backslash NP)/NP) \\
\lambda p.\neg p(-sk_{\lambda x.student(x);\lambda y.|y|>=2}) & \lambda x\lambda y\lambda e.pass(y,x,e) & \lambda p.p(\circ sk_{\lambda x.exam(x)}) \\
\cline{2-3}
 & \multicolumn{2}{c}{S_{dcl}\backslash NP} \\
 & \multicolumn{2}{c}{\lambda x\lambda e.pass(x,\circ sk_{\lambda y.exam(y)},e)} \\
\hline
\multicolumn{3}{c}{S_{dcl}} \\
\multicolumn{3}{c}{\lambda e.\neg pass(-sk_{\lambda x.student(x);\lambda y.|y|>=2},-sk_{\lambda z.exam(z)},e)} \\
\hdashline
\multicolumn{3}{c}{\neg\exists x[exam(x)\wedge\exists y[\exists z[\neg z=y\wedge\forall u[y=u\vee z=u \implies student(u)\wedge\exists e[pass(u,x,e)]]]]]}
\end{array}
$$

Figure 3.7: *At least 1 student passed an exam* This example shows how the system deals with upper bounds. The interpretation denies the existence of two different students who both passed the same exam.

$$
\begin{array}{ccc}
\text{A man} & \text{loves} & \text{every woman} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/NP & (S_{dcl}\backslash NP)\backslash((S_{dcl}\backslash NP)/NP) \\
\lambda p.p(sk_{\lambda x.man(x)}) & \lambda x\lambda y\lambda e.love(y,x,e) & \lambda p.\forall x[woman(x)\Rightarrow p(x)] \\
\cline{2-3}
 & \multicolumn{2}{c}{S_{dcl}\backslash NP} \\
 & \multicolumn{2}{c}{\lambda x\lambda e.\forall y[woman(y)\Rightarrow love(x,y,e)]} \\
\hline
\multicolumn{3}{c}{S_{dcl}} \\
\multicolumn{3}{c}{\lambda e.\forall x[woman(x)\Rightarrow love(\left\{\begin{array}{c} sk^{()} \\ sk^{(x)} \end{array}\right\}_{\lambda y.man(y)},x,e)]}
\end{array}
$$

Figure 3.8: Output for a simple sentence contained scope ambiguities and scope inverion. It shows how the Skolem term becomes ambiguous when it becomes enclosed by the universal quantifier.

$$\begin{array}{cccccc}
\text{Every boy} & \text{admires} & \text{and} & \text{every girl} & \text{detests} & \text{some saxophonist} \\
\hline
S_X/(S_X\backslash NP) & (S_{dcl}\backslash NP)/NP & ((S_{dcl}/NP)\backslash(S_{dcl}/NP))/(S_{dcl}/NP) & S_X/(S_X\backslash NP) & (S_{dcl}\backslash NP)/NP & S_{dcl}\backslash(S_{dcl}/NP) \\
\lambda p.\forall x[boy(x) \implies p(x)] & \lambda x\lambda y\lambda e.admire(y,x,e) & \lambda p\lambda q\lambda x\lambda e.q(x,e)\wedge p(x,e) & \lambda p.\forall x[girl(x) \implies p(x)] & \lambda x\lambda y\lambda e.detest(y,x,e) & \lambda p.p(+sk_{\lambda x.saxophonist(x)})
\end{array}$$

$$\xrightarrow{>\mathbf{B}}\quad S_{dcl}/NP \qquad \lambda x\lambda e.\forall y[boy(y) \implies admire(y,x,e)]$$

$$\xrightarrow{>\mathbf{B}}\quad S_{dcl}/NP \qquad \lambda x\lambda e.\forall y[girl(y) \implies detest(y,x,e)]$$

$$\xrightarrow{>}\quad (S_{dcl}/NP)\backslash(S_{dcl}/NP) \qquad \lambda p\lambda x\lambda e.p(x,e)\wedge\forall y[girl(y) \implies detest(y,x,e)]$$

$$\xrightarrow{<}\quad S_{dcl}/NP \qquad \lambda x\lambda e.\forall y[boy(y) \implies admire(y,x,e)]\wedge\forall z[girl(z) \implies detest(z,x,e)]$$

$$\xrightarrow{<}\quad S_{dcl}$$

$$\lambda e.\forall x[boy(x) \implies admire(x,\left\{\begin{array}{c}+sk_{35}^{()}\\+sk_{35}^{(x)}\end{array}\right\}_{\lambda y.saxophonist(y)},e)]\wedge\forall z[girl(z) \implies detest(z,\left\{\begin{array}{c}+sk_{35}^{()}\\+sk_{35}^{(z)}\end{array}\right\}_{\lambda y.saxophonist(y)},e)]$$

$$\exists e[\forall x[boy(x) \implies \exists y[saxophonist(y)\wedge admire(x,y,e)]]\wedge\forall z[girl(z) \implies \exists u[saxophonist(u)\wedge detest(z,u,e)]]]$$

$$\exists e[\exists x[saxophonist(x)\wedge\forall y[boy(y) \implies admire(y,x,e)]\wedge\forall z[girl(z) \implies detest(z,x,e)]]]$$

Figure 3.9: *Every boy admires and every girl detests some saxophonist* System output for the Geach sentence. The system correctly handles the right-node-raising construction by composition and coordination, to build a logical form that captures the relations between boys, girls and saxophonists. As explained in Section 3.3.1.4, the logical form can be unpacked to reveal the correct two interpretations (e.g. it does not predict interpretations where saxophonists are wide-scope with respect to boys but narrow-scope with respect to girls). Unfortunately, to get obtain the correct syntactic analysis I had to manually set the supertag for *admires*, which the parsing model assigns as a noun. Such mistakes are frequent, and highlight the fact that the semantic analysis is highly reliant on the syntax.

$$
\begin{array}{ccccc}
\text{I} & \text{asked} & \text{her} & \text{to} & \text{talk} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & ((S_{dcl}\backslash NP)/(S_{to}\backslash NP))/NP & ((S_{dcl}\backslash NP)/(S_{to}\backslash NP))\backslash(((S_{dcl}\backslash NP)/(S_{to}\backslash NP))/NP) & (S_{to}\backslash NP)/(S_b\backslash NP) & S_b\backslash NP \\
\lambda p.p(i) & \lambda x\lambda p\lambda y\lambda e.ask(e)\wedge arg(y,e)\wedge\exists e'[p(x,e')] & \lambda p.p(her) & \lambda p\lambda x\lambda e.p(x,e) & \lambda x\lambda e.talk(e)\wedge arg(x,e)
\end{array}
$$

$$\text{asked her} \xrightarrow{<} (S_{dcl}\backslash NP)/(S_{to}\backslash NP) : \lambda p\lambda x\lambda e.ask(e)\wedge arg(x,e)\wedge\exists e'[p(her,e')]$$

$$\text{to talk} \xrightarrow{>} S_{to}\backslash NP : \lambda x\lambda e.talk(e)\wedge arg(x,e)$$

$$\text{asked her to talk} \xrightarrow{>} S_{dcl}\backslash NP : \lambda x\lambda e.ask(e)\wedge arg(x,e)\wedge\exists e'[talk(e')\wedge arg(her,e')]$$

$$\text{I asked her to talk} \xrightarrow{>} S_{dcl} : \lambda e.ask(e)\wedge arg(i,e)\wedge\exists e'[talk(e')\wedge arg(her,e')]$$

Figure 3.10: System output for an object control verb. In the logical form, the *her* is doing the talking. CCGBank has the same categories for subject and object control verb, so subject-control verbs (like *promise*) would need hand-built lexical entries.

3.6.3 Comparison with Boxer

An obvious comparison for the system is with Boxer. The systems are very similar in principle, as both approaches are rule-based conversions of CCG parses onto logical forms—and both then convert their output to first-order logic for inference. One difference, which is useful for the work described later in this thesis, is a different analysis of prepositions. In first-order logic, Boxer analyses the expression *the author of Macbeth* as $\exists x[author(x) \wedge of(x, macbeth)]$. However, the preposition *of* has little meaning in itself, and can only be interpreted with respect to the noun. For example, it is difficult to see how to create good inference rules of the form: $of(x,y) \rightarrow p(x,y)$. Instead, we analyse the expression as $\exists x[author_{be,of}(x, macbeth)]$, so that all the predicates have a clear meaning.

3.6.4 Comparison with Dependency Syntax

Syntactic dependeyncy representations, such as Standford dependencies, and currently widely used in NLP applications. While there is a lack of empirical work comparing these representations, there are a number of reasons for preferring CCG in this thesis. The logical forms derived from our CCG parses can abstract over many syntactic variations on the same meaning, which have different dependency parses. For example, the dependency parses for the following sentences would all express a different relationship between *John* and *cake*, but would all receive the same interpretation in a logical form:

- *John baked a cake,*
- *A cake was baked by John*
- *John baked a cookie and a cake*
- *A cake that John baked*
- *John, who baked a cake*
- *John baked and ate a cake.*
- *John baked Mary a cookie and Sue a cake*

This variation means that additional learning is required on top of the dependency parse to judge whether they are equivalent. Some of these examples can be improved

Premises:	Every European has the right to live in Europe. Every European is a person. Every person who has the right to live in Europe can travel freely within Europe.
Hypothesis:	Every European can travel freely within Europe
Solution:	Yes

Premises:	Few committee members are from Portugal. All committee members are people. All people who are from Portugal are from southern Europe.
Hypothesis:	There are few committee members from southern Europe.
Solution:	Unknown

Premises:	One of the leading tenors is Pavarotti. Neither leading tenor comes cheap.
Hypothesis:	Pavarotti is a leading tenor who comes cheap.
Solution:	No

Figure 3.11: Example problems from the FraCaS suite.

by post-processing the dependency parse. However, I am not aware of any adequate way of representing argument cluster coordination (as in *John baked Mary a cookie and Sue a cake*) in a dependency parse, whereas CCG has an elegant account of this construction [Steedman, 2012].

3.6.5 Experiments on the FraCaS Suite

The FraCaS suite [Cooper et al., 1996][7] contains a hand-built set of entailment problems designed to be challenging in terms of formal semantics. Section 1 is used, which contains 74 problems requiring an understanding of quantifiers[8]. They do not require any knowledge of lexical semantics, meaning that the evaluation focusses purely on the understanding of quantifiers and composition. Figure 3.11 gives several example problems.

The only previous work I am aware of on this dataset is by MacCartney and Man-

[7]Using the version converted to machine readable format by MacCartney and Manning [2007]
[8]Excluding 6 problems without a defined solution.

ning [2007]. Their Natural Logic[9] approach is supplied with a small handbuilt lexicon of function words, which marks how the polarity of the word affects the polarity of its children. It uses this to transform a sentence into a polarity annotated string. The system then aims to transform the premise string into a hypothesis. Positively polarized words can be replaced with less specific ones (e.g. by deleting adjuncts), whereas negatively polarized words can be replaced with more specific ones (e.g. by adding adjuncts). Whilst this approach is high-precision and often useful, this logic is unable to perform inferences with multiple premise sentences (in contrast to the first-order logic used by the CCG system).

Development consists of adding entries to the lexicon for quantifiers. For simplicity, multi-word quantifiers like *at least a few* are treated as being multi-word expressions—although a more compositional analysis may be possible. Following MacCartney and Manning [2007], the evaluation does not use held-out data—each problem is designed to test a different issue, so it is not possible to generalize from one subset of the suite to another. The design of the test-suite is analagous to a 'unit test', where each problem checks a separate edge-case, but with little overlap.

As the aim is to evaluate the semantics, not the parser, gold-standard lexical categories were annotated for sentences with parser errors. A consequence of CCG's close link between syntax and semantics is that any syntactic mistake causes incorrect semantics. Although an N-best parser is used [Ng and Curran, 2012], this does not help in many cases as errors are caused by missing entries in the supertagger and POS-tagger lexicons. For example, *European* is frequently used as a noun in the FraCaS suite examples, as in *Every European is a person.* Unfortunately, *European* is only used as an adjective in CCGBank (as in *European minister*), meaning that the supertagger model is unable to assign the correct category. Without the correct supertag, no parse can deliver the correct dreivation. The problems contain up to 5 sentences, increasing the chance that one will contain a parse error.

Following Bos and Markert [2005], a combination of the Prover9 theorem prover and Mace4 model builder McCune [2005] theorem prover is used for inference, returning *yes* if the premise implies the hypothesis, *no* if it implies the negation of the hypothesis, and *unknown* otherwise[10]. The theorem prover attempts to find a contradiction in the input, while the model builder attempts to prove that the input is consistent

[9]Despite the similar names, Natural Logic is quite different from Natural Semantics

[10]It also returns *unknown* if both the hypothesis and negation of the hypothesis can be proven—which can happen if the premises are inconsistent

System	Single Premise	Multiple Premises
MacCartney&Manning 07	84%	-
MacCartney&Manning 08	98%	-
CCG-Dist (parser syntax)	70%	50%
CCG-Dist (gold syntax)	89%	80%

Table 3.3: Accuracy on Section 1 of the FraCaS suite. Problems are divided into those with one premise sentence (44) and those with multiple premises (30). I do not give an overall number, as the split into problems with single and multiple premise sentences is an arbritraty choice by the authors of the dataset.

by constructing a model that satisfies it. As they can be run in parallel, the combination can take less time than running either to exhaustion.

Results are shown in Table 3.3, and highlight the strengths and weaknesses of our CCG approach compared to Natural Logic. The CCG system improves on previous work by being capable of multi-sentence inferences. Causes of errors include missing a distinct lexical entry for plural *the* (meaning *all*), only taking existential interpretations of bare plurals, failing to interpret mass-noun determiners such as *a lot of*, and not providing a good semantics for non-monotone determiners such as *most*. These problems should be surmountable with further work.

Every error except one is due to incorrectly predicting *unknown*—the system makes just one error on *yes* or *no* predictions (with or without gold syntax). This result suggests that extending downstream applications with first-order logic inferences will not harm precision, and can potentially boost recall.

The system is less robust than MacCartney and Manning [2007] to syntax errors, who achieve excellent performance using a parser of comparable accuracy. One reason is that the CCG semantics is much more closely integrated with the syntax than in Natural Logic. For Natural Logic, the parser simply has to identify the scope of negation, and then the inference can be done at the string level. Conversely, using logical forms for inference allows the CCG system to attempt more of the problems (i.e. those with multi-sentence premises).

3.7 Future Work

Modelling plurality is an obvious area for improvement. For example, it would be useful to distinguish collective and distributive verbs (as in Steedman [2012]), such as *sleeps*, with lexical entries such as:

gather $\vdash S\backslash NP_{pl} : \lambda s \lambda e.gather(s,e)]$

sleep $\vdash S\backslash NP_{pl} : \lambda s \lambda e.\forall x[x \in s \Rightarrow sleep(x,e)]$

The current syntax does not mark plurals, so it is not possible to assign the distributive interpretation. It would be helpful if CCGBank were extended to mark which *NP*s were singular or plural. Of course, plural agreement is also marked syntactically, so may be useful for parsing.

The current approach to modelling scope is believed to capture the set of all possible interpretations, but does not attempt to disambiguate them. Choosing which of these readings is the intended reading would require a probabilistic model, such as that of Srinivasan and Yates [2009], which would allow us to derive a distribution over interpretations (rather than a set). The fact that a probabilistic model would be helpful does not mean that using linguistic constraints is unnecessary—as noted in Section 3.3.1.1, underspecification approaches can generate thousands of spuriously equivalent interpretations of a sentence which are all equally valid. The Natural Semantics approach would mean the model would only have to assign probabilities to the genuine ambiguities.

Current work on modelling quantifiers is limited by the lack of available annotated data. Morante and Blanco [2012] annotated a corpus that marks the scope of negation in text. Similar work on marking quantifier scopes would be extremely useful, for training and evaluating models, and validating linguistic intuitions about available readings. The Gronigen Meaning Bank [Basile et al., 2012] is a useful step in this direction.

Lev et al. [2004] make the intruiging proposal of evaluating models of formal semantics on LSAT problems, which contain natural language logic puzzles. Solving such problems requires little in the way of lexical semantics, but a good understanding of quantifiers, negation, and complex function words such as *same*, *only* and *different*. I experimented with a number of such problems, and while it was possible to solve some examples, the major obstacle was syntactic errors. A consequence of the close link between syntax and semantics in CCG is that, in general, a single syntactic error is enough to cause the inference to fail. As such puzzles typically involve the understanding of a passage of several sentences, even state-of-the-art parsers are unlikely to parse every sentence correctly. There are also many cases where the current CCGBank syntax does not support the correct semantic interpretation, for example in the treatment of comparative constructions. Further refinements to CCGBank, in the spirit of Honnibal et al. [2010], would be useful in these cases.

3.8 Conclusions

This chapter has developed the first wide-coverage computational model of the Natural Semantics theory described in Steedman [2012], discovering and correcting a flaw in the original theory. The model produces packed-logical forms that model scope ambiguities, and captures the underlying predicate-argument structure for a variety of linguistically complex constructions. It also has high coverage of unseen text. I have also shown that the system has the ability to make complex multi-sentence inferences using quantifiers.

On the other hand, the system has a weak model of lexical semantics, and performance would be poor on natural language applications. The rest of this thesis concentrates on addressing this weakness.

CHAPTER 4

Combined Distributional and Logical Semantics

4.1 Introduction

This chapter introduces the main idea of the thesis, which is a new method for combining distributional and logical semantics. The approach closely follows standard CCG semantics, except that the non-logical constants in lexical entries are replaced with distributionally-induced cluster identifiers, allowing distinct content words to express the same semantics. The chapter proceeds as follows:

- Section 4.2 motivates combining formal and distributional semantics, arguing that many practical inferences rely on simultaneously understanding lexical, logical and compositional aspects of semantics.
- Section 4.3 introduces a simple model which captures the desired properties.
- Section 4.4 shows how this model can be extended to model ambiguity. I introduce a novel probabilistic model of ambiguity, and show how to incorporate it into CCG derivations.

Input Sentence

Shakespeare wrote Macbeth

$\Downarrow$

Initial semantic analysis

$write_{arg0,arg1}$*(shakespeare, macbeth)*

$\Downarrow$

Entity Typing

$write_{arg0:PER,arg1:BOOK}$*(shakespeare:PER, macbeth:BOOK)*

$\Downarrow$

Distributional semantic analysis

relation37(shakespeare:PER, macbeth:BOOK)

Figure 4.1: Layers used in the model.

- In Section 4.5, The model is then evaluated on a question answering task, showing good performance.
- Section 4.6, discusses how this model relates to a variety of existing approaches.

.

The work in this chapter has previously been published as Lewis and Steedman [2013a].

Figure 4.1 gives an overview of the model developed in this chapter.

4.2 Motivating Combining Distributional and Logical Semantics

There has been much recent progress in unsupervised distributional semantics, in which the meaning of a word is induced based on its usage in large corpora. This approach is useful for a range of key applications including question answering and relation extraction [Lin and Pantel, 2001, Poon and Domingos, 2009, Yao et al., 2011]. Because such a semantics can be automatically induced, it escapes the limitation of depending on relations from hand-built training data, knowledge bases or ontologies, which have proved of limited use in capturing the huge variety of meanings that can be expressed in language. See Section 2.4 for a more detailed discussion of distributional semantics.

However, distributional semantics has largely developed in isolation from the formal semantics literature. Whilst distributional semantics has been effective in modelling the meanings of content words such as nouns and verbs, it is less clear that it can be applied to the meanings of function words. Semantic operators, such as determiners, negation, conjunctions, modals, tense, mood, aspect, and plurals are ubiquitous in natural language, and are crucial for high performance on many practical applications—but current distributional models struggle to capture even simple examples. Conversely, computational models of formal semantics have shown low recall on practical applications, stemming from their reliance on ontologies such as WordNet [Miller, 1995] to model the meanings of content words [Bobrow et al., 2007, Bos and Markert, 2005].

For example, consider what is needed to answer a question like *Did Google buy YouTube?* from the following sentences:

1. Google purchased YouTube
2. Google's acquisition of YouTube
3. Google acquired every company
4. YouTube may be sold to Google
5. Google will buy YouTube or Microsoft
6. Google didn't takeover YouTube

The examples require knowledge of lexical semantics (e.g. that *buy* and *purchase* are synonyms), but some also need interpretation of quantifiers, negatives, modals and disjunction. It seems unlikely that either distributional or formal approaches can accomplish the task alone.

4.3 A Simple Model for Combining Distributional and Logical Semantics

The approach to combining distributional and logical semantics is to attempt to learn a CCG lexicon which maps semantically equivalent words onto the same logical form—for example learning entries such as:

$$
\begin{array}{ccc}
\text{Shakespeare} & \text{wrote} & \text{Macbeth} \\
\hline
NP & (S\backslash NP)/NP & NP \\
shakespeare' & \lambda y \lambda x.relation37(x,y) & macbeth'(y) \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP \quad >} \\
 & \multicolumn{2}{c}{\lambda x.relation37(x, macbeth')} \\
\hline
\multicolumn{3}{c}{S \quad <} \\
\multicolumn{3}{c}{relation37(shakespeare', macbeth')}
\end{array}
$$

Figure 4.2: A CCG derivation for *Shakespeare wrote Macbeth* using clusters, in which the predicate $write_{arg0,arg1}$ has been mapped to cluster 37. Figure 4.3 shows how the same logical form can be derived for the sentence *Shakespeare is the author of Macbeth*

author $\vdash \mathrm{N/PP_{of}} : \lambda x \lambda y.relation37(x,y)$
write $\vdash \mathrm{(S\backslash NP)/NP} : \lambda x \lambda y.relation37(x,y)$

Intuitively, these lexical entries encapsulate the idea that if two words express the same meaning, they should have the same lexical semantics.

The only change to the standard CCG derivation is that the symbols used in the logical form are arbitrary relation identifiers. These symbols are learnt by first mapping to a deterministic logical form (using predicates such as $author_{be,of}$ and $write_{arg0,arg1}$), using the process developed in Chapter 3, and then clustering predicates (both verbal and nominal) based on their arguments. This lexicon can then be used to parse new sentences, and integrates seamlessly with CCG theories of formal semantics.

4.3.1 Initial Semantic Analysis

The starting point for the method is a standard formal-semantic analysis, using the system described in Chapter 3, which maps CCG syntax trees onto logical forms. Lexical entries for content words are generated automatically based on the words and its CCG category, and are supplemented with a small manual lexicon of function words (such as *not* and *every*). A number of small changes are made to the semantic parser from Chapter 3, as explained beneath.

4.3.1.1 Make Adjuncts Core Arguments

Many semantic theories distinguish *core arguments* of predicates and *adjuncts*. The version of CCGBank used in this thesis [Honnibal et al., 2010] makes the distinction

Shakespeare	is	the	author	of	Macbeth
$NP^{\uparrow}$	$(S\backslash NP)/NP$	$NP^{\uparrow}/N$	N/PP_{of}	$PP^{\uparrow}_{of}/NP$	$NP^{\uparrow}$
$\lambda p.p(shakespeare')$	$\lambda y\lambda x.x=y$	$\lambda p\lambda q.q(sk_{\lambda x.p(x)})$	$\lambda y\lambda x.rel37(x,y)$	$\lambda x.x$	$\lambda p.p(macbeth')$

$$\xrightarrow{\quad\quad} > \quad PP^{\uparrow}_{of} \quad \lambda p.p(macbeth')$$

$$\xrightarrow{\quad\quad} > \quad N \quad \lambda x.rel37(x,macbeth')$$

$$\xrightarrow{\quad\quad} > \quad NP^{\uparrow} \quad \lambda q.q(sk_{\lambda x.rel37(x,macbeth')})$$

$$\xrightarrow{\quad\quad} < \quad S\backslash NP \quad \lambda y.y = sk_{\lambda x.rel37(x,macbeth')}$$

$$\xrightarrow{\quad\quad} > \quad S \quad shakespeare' = sk_{\lambda x.rel37(x,macbeth')}$$

$$\cdots\cdots\cdots \quad rel37(shakespeare',macbeth')$$

Figure 4.3: A CCG derivation of *Shakespeare is the author of Macbeth*, where clustering maps $author_{be,of}$ to cluster 37. Figure 4.2 shows how the same logical form can be derived for the sentence *Shakespeare wrote Macbeth*

based on the Propbank annotations [Kingsbury and Palmer, 2002]. Unfortunately, such distinctions are very difficult for parsers to make, and can lead to inconsistent analyses of sentences. I found it useful in practice to assume all adjuncts were arguments, which leads to greater consistency.

The conversion is made by searching for adverbial $((S\backslash NP)\backslash(S\backslash NP))/NP$ and adnomial $(N\backslash N)/NP$ prepositions, and replacing them with the core-argument category PP/NP. The change is then propagated through the tree, so that the noun or verb that was originally modified gains an extra PP argument. In cases where the main verb is modified by auxiliary or secondary verbs, care must be taken to attach the PP to the main verb (e.g. in *Shakespeare was born in Stratford*, the PP, should attach to *born*, not *was*).

For example, the parser may output an analysis such as the following:

$$
\begin{array}{ccccc}
\text{Shakespeare} & \text{was} & \text{born} & \text{in} & \text{Stratford} \\
\hline
NP^{\uparrow} & (S_{dcl}\backslash NP)/(S_b\backslash NP) & S_b\backslash NP & ((S\backslash NP)\backslash(S\backslash NP))/NP & NP^{\uparrow} \\
 & \multicolumn{2}{c}{\xrightarrow{} >\mathbf{B}} & \multicolumn{2}{c}{\xrightarrow{} <} \\
 & \multicolumn{2}{c}{S_{dcl}\backslash NP} & \multicolumn{2}{c}{(S\backslash NP)\backslash(S\backslash NP)} \\
 & \multicolumn{4}{c}{\xrightarrow{} <} \\
 & \multicolumn{4}{c}{S\backslash NP} \\
\multicolumn{5}{c}{\xrightarrow{} >} \\
\multicolumn{5}{c}{S}
\end{array}
$$

The previous derivation is automatically converted to:

$$
\begin{array}{ccccc}
\text{Shakespeare} & \text{was} & \text{born} & \text{in} & \text{Stratford} \\
\hline
NP^{\uparrow} & (S_{dcl}\backslash NP)/(S_b\backslash NP) & (S_b\backslash NP)/PP_{in} & PP^{\uparrow}_{in}/NP & NP^{\uparrow} \\
 & \multicolumn{2}{c}{\xrightarrow{} >\mathbf{B}} & \multicolumn{2}{c}{\xrightarrow{} <} \\
 & \multicolumn{2}{c}{(S_{dcl}\backslash NP)/PP_{in}} & \multicolumn{2}{c}{PP^{\uparrow}_{in}} \\
 & \multicolumn{4}{c}{\xrightarrow{} <} \\
 & \multicolumn{4}{c}{S\backslash NP} \\
\multicolumn{5}{c}{\xrightarrow{} >} \\
\multicolumn{5}{c}{S}
\end{array}
$$

4.3.1.2 Binarizing Predicates

The focus of this thesis is on inducing binary relations. Many existing approaches have shown how to produce good clusterings of (non-event) nouns [Brown et al., 1992], any of which could be simply integrated into the framework developed here. However, relation clustering remains an open problem (see Section 2.4.2). Relatively little work has attempted to cluster predicates with variable numbers of arguments—USP [Poon and Domingos, 2009] is one exception.

I take a simple (but novel) approach to circumventing clustering relations with more than 2 arguments. Higher order relations are binarized, by creating a binary relation between each pair of arguments. For example, in the sentence *Russia sold Alaska to the United States* the ditransitive verb *sell* would have the following lexical entry:

$sell \vdash (S\backslash NP)/PP_{to}/NP : \lambda x \lambda y \lambda z.sell_{arg0,arg1}(z,y) \wedge sell_{arg0,to}(z,x) \wedge sell_{arg1,to}(y,x)$

The three binary relations roughly correspond to *sellToSomeone(Russia, Alaska), buyFromSomeone(US, Alaska), sellSomethingTo(Russia, US).*

This transformation does not exactly preserve meaning, but captures the most important relations. For example, the system can infer that *Russia sold Alaska to the United States→Russia sold Alaska*, as both express the *sellToSomeone(Russia, Alaska)* relation.

Expressing the predicates in this way allows the system to compare semantic relations across different syntactic types—for example, both transitive verbs and argument-taking nouns can be seen as expressing binary semantic relations between entities.

Section 6.7.3 suggests how the system could be improved to give a better handling of n-ary relations.

4.3.2 Distributional Relation Clustering

Building the distributional lexicon requires clustering predicates that are semantically equivalent. Following the distributional hypothesis, it is assumed that predicates with similar usage in a large corpus will be semantically similar. First, the CCG semantics from Section 4.3.1 is used to gather statistics on predicates. Then, predicates are clustered based on these statistics.

4.3.2.1 Corpus Statistics

Predicates are clustered based on their usage in a large corpus. The standard CCG approach described in Section 4.3.1 is run over sentences. Then, the arguments of binary predicates are extracted from the logical form.

For each predicate, a vector is built containing the count of each proper-noun argument pair. Alternative approaches have used statistics based on individual arguments (e.g. [Lin and Pantel, 2001]). The corpus used here is larger than most previous work, reducing issues of sparsity, and proper-noun argument pairs may be more discriminative. For example, the predicates *X was born in Y* and *X lives in Y* will have very similar vectors of individual arguments—for both, the *X* could be filled by any person, and the *Y* slot could be any place. However, taking argument pairs may give more dis-

	$born_{arg1,in}$	$birthplace_{poss,be}$
(Shakespeare, Stratford)	16	5
(Obama, Hawaii)	37	8
(Obama, 1961)	42	0
(Jesus, Bethlehem)	106	25
(Napoleon, Corsica)	7	2
(Shakespeare, 1564)	28	0

Table 4.1: Some example vectors for two similar predicates. The similarity of these vectors is evidence they can be clustered.

criminative features, as there will be fewer semantic relations that hold between pairs like *(Obama,Hawaii)* and *(Jesus,Bethlehem)*. These vectors can be viewed as samples from the denotations of the predicates. If two predicates have the similar vectors with a large sample size, this is evidence that they are samples from the same denotation, and are therefore semantically equivalent.

Table 4.1 shows example vectors for two semantically related predicates which should be clustered.

4.3.2.2 Clustering

Many algorithms have been proposed for clustering predicates based on their arguments [Poon and Domingos, 2009, Yao et al., 2012]. The number of relations in the corpus is unbounded, so the clustering algorithm should be non-parametric. It is also important that it remains tractable for very large numbers of predicates and arguments, in order to give a greater coverage of language than can be achieved by hand-built ontologies.

Predicates are clustered using the Chinese Whispers algorithm [Biemann, 2006], a simple graph clustering algorithm summarized in Algorithm 1. Although somewhat ad-hoc, it is both non-parametric and highly scalable[1]. This algorithm has previously

[1] I also experimented with a Dirichlet Process Mixture Model [Neal, 2000], which is a more principled Bayesian approach to non-parametric flat clustering. Using the 'Chinese Restaurant' analogy [Aldous, 1985] each of the 'tables' corresponds to an underlying semantic relation, and the 'dishes' served are the entity-pairs observed for that relation. Predicates represent 'customers', and are likely to choose the same tables as other predicates with similar arguments. However, even with the efficient A* search algorithms introduced by Daumé III [2007], the cost of inference was found to be prohibitively high when run at large scale. The quality of the clustering was also highly dependent on the choice of hyper parameters.

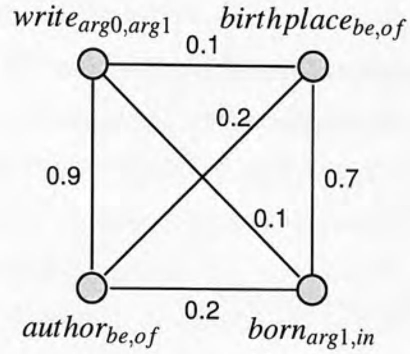


Figure 4.4: Example input graph for the Chinese Whispers clustering, in which nodes a represent predicates and edge-weights are distributional similarities. The algorithm will partition the graph into two clusters.

been used for noun clustering by Fountain and Lapata [2011], who argue it is a cognitively plausible model for language acquisition.

The collection of predicates and arguments is converted into a graph with one node per predicate, where edge weights represent the similarity between predicates. An example input graph is shown in Figure 4.4. Predicates with different types have zero-similarity, and otherwise similarity is computed as the cosine-similarity of the tf-idf vectors of argument-pairs. As cosine-similarity is symmetric, the graph is undirected. The system prunes nodes occurring fewer than 20 times, edges with weights less than 0.002, and a short list of stop predicates[2]. Removing low-weight edges is important, as it allows predicates which only have very low similarities to any other predicate to be assigned to their own singleton cluster, and prevents overly general clusters from forming. The value was chosen based on empirical observation of the clustering.

4.4 Modelling Ambiguity

The model developed in Section 4.3 has many limitations. Two of the most serious are that it cannot model ambiguous words, and that the clustering problem is intractable with large numbers of predicates. In this section, I show how adding types to entities and predicates can address both of these weaknesses.

[2]Mostly verbs which are frequently light, such as *make*, *take* or *give*, where the real predicate is often the object rather than the verb. Section 6.7.5 discusses better ways of modelling such predicates.

Data: Set of predicates P

Result: A cluster assignment r_p for all $p \in P$

$\forall p \in P : r_p \longleftarrow$ unique cluster identifier;

while *not converged* **do**

  randomize order of P

  for $p \in P$ **do**

    $r_p \longleftarrow \arg\max_r \sum_{p'} \mathbb{1}_{r=r_{p'}} sim(p, p')$

  end

end

Algorithm 1: Chinese Whispers algorithm, used for predicate clustering. $sim(p, p')$ is the distributional similarity between p and p', and $\mathbb{1}_{r=r'}$ is 1 iff r=r' and 0 otherwise. The algorithm is not guaranteed to terminate in pathological cases [Biemann, 2006], but this problem can be avoided by bounding the number of iterations. In practice, it converged in all experiments.

4.4.1 Entity Typing

Typing predicates—for example, determining that *writing* is a relation between people and books—has become standard in relation clustering [Schoenmackers et al., 2010, Berant et al., 2011, Yao et al., 2012]. Section 4.4.3 demonstrates how to build a typing model into the CCG derivation, by subcategorizing all terms representing entities in the logical form with a more detailed type. These types are also induced from text[3], as explained in Section 4.4.2, but for convenience they are described here with human-readable labels, such as *PER*, *LOC* and *BOOK*.

A key advantage of typing is that it allows the system to model ambiguous predicates. Following Berant et al. [2011], different type signatures of the same predicate are assumed to have different meanings, but given a type signature a predicate is unambiguous. For example a different lexical entry for the verb *born* is used in the contexts *Obama was born in Hawaii* and *Obama was born in 1961*, reflecting a distinction in the semantics that is not obvious in the syntax[4].

Typing also greatly improves the efficiency of clustering, as the system only needs

[3] An alternative would have been to use WordNet for typing. However, this approach would introduce additional difficulties. For example, many named-entities are not present in WordNet, and disambiguating to WordNet senses is a hard problem with low inter-annotator agreement [Hovy et al., 2006]

[4] Whilst this assumption is very useful, it does not always hold—for example, the genitive in *Shakespeare's book* is ambiguous between *ownership* and *authorship* relations even given the types of the arguments.

to compare predicates with the same type during clustering (for example, the system does not have to consider clustering a predicate between people and places with predicates between people and dates). Almost all clustering algorithms are superlinear, so there is an advantage in decomposing the clustering problem into many smaller ones. Even the simple and scalable Chinese Whispers algorithm used in Section 4.3.2.2 requires a quadratic number of cosine-similarity calculations, which cannot be scaled to vocabularies with tens of thousands of predicates. In Chapter 6, a much more accurate but more expensive clustering algorithm is used, which is made possible by this development.

4.4.2 Topic Model

The entity-typing model assigns types to nouns, which is useful for disambiguating polysemous predicates. The approach is similar to O'Seaghdha [2010] in that it aims to cluster entities based on the noun and unary predicates applied to them (it is simple to convert from the binary predicates to unary predicates). For example, the pair *(born$_{in}$, 1961)* should map to a DAT type, and *(born$_{in}$, Hawaii)* should map to a LOC type. This is non-trivial, as both the predicates and arguments can be ambiguous between multiple types—but topic models offer a good solution (described below).

The type of each argument of a predicate is assumed to depend only on the predicate and argument, although Ritter et al. [2010] demonstrate an advantage to modelling the joint probability of the types of multiple arguments of the same predicate, and Yao et al. [2012] shows the importance of document level features. The standard Latent Dirichlet Allocation model [Blei et al., 2003] is used, which performs comparably to more complex models proposed in O'Seaghdha [2010].

In topic-modelling terminology, a 'document' is constructed for each unary predicate (e.g. *born$_{in}$*), based on its set of argument entities ('words'). The model assumes that these arguments are drawn from a small number of types ('topics'), such as PER, DAT or LOC[5]. Example documents are shown in Table 4.2. Each type j has a multinomial distribution ϕ_j over arguments (for example, a LOC type is more likely to generate *Hawaii* than *1961*). Each unary predicate i has a multinomial distribution θ_i over topics, so the *born$_{in}$* predicate will normally generate a DAT or LOC type. Sparse Dirichlet priors α and β on the multinomials bias the distributions to be peaky. The parameters are estimated by Gibbs sampling, using the Mallet implementation [Mc-

[5]Types are induced from the text, but I give human-readable labels here for convenience.

Unary Predicate	Arguments
$born_{in}$	Hawaii, Bethlehem, 1961, Stratford, 1564, 1985, ...
$year_{of}$	2001, 1963, 1961, 2014, 1564, 1845, ...
$live_{in}$	Hawaii, Bethlehem, London, Paris, Edinburgh, ...
die_{in}	Dallas, Edinburgh, 1963, Paris, 1918, 1985, ...
$travel_{to}$	Edinburgh, Hawaii, London, Paris, Sydney, ...

Table 4.2: Hypothetical pseudo-documents for predicats, which are used as input to the LDA topic model. From this data, the model is likely to learn that arguments are a mixture of 'place' and 'date' topics, and the selectional preferences on the predicates.

Callum, 2002].

The generative story to create the data is:

For every type k:

  Draw the $p(arg|k)$ distribution ϕ_k from $Dir(\beta)$

For every unary predicate i:

  Draw the $p(type|i)$ distribution θ_i from $Dir(\alpha)$

  For every argument j:

    Draw a type z_{ij} from $Mult(\theta_i)$

    Draw an argument w_{ij} from $Mult(\phi_{\theta_i})$

Following O'Seaghdha [2010], Ritter et al. [2010], a small number of very frequent, highly ambiguous predicates that have very weak selectional preferences are excluded from the clustering. This was found to both improve the speed and performance of the clustering.

4.4.3 Typing in Logical Form

In the logical form, all constants and variables representing entities x can be assigned a distribution over types $p_x(t)$ using the type model. An initial type distribution is applied in the lexicon, using the ϕ distributions for the types of nouns, and the θ_i distributions for the type of arguments of binary predicates (inverted using Bayes' rule). Then at each β-reduction in the derivation, the type probabilities are updated to be the product of the type distributions of the terms being reduced. If two terms x and

$$
\dfrac{\dfrac{\text{file}}{(S\backslash NP)/NP \atop \lambda y{:}\left\{\begin{array}{c} DOC=0.5 \\ LEGAL=0.4 \\ CLOTHES=0.01 \\ \ldots \end{array}\right\} \lambda x{:}\left\{\begin{array}{c} PER=0.7 \\ ORG=0.2 \\ \ldots \end{array}\right\} .file_{arg0,arg1}(x,y)} \quad \dfrac{\text{a suit}}{NP^{\uparrow} \atop \lambda p.\exists y{:}\left\{\begin{array}{c} CLOTHES=0.6 \\ LEGAL=0.3 \\ DOC=0.001 \\ \ldots \end{array}\right\} [suit'(y) \wedge p(y)]}}{S\backslash NP \atop \lambda x{:}\left\{\begin{array}{c} PER=0.7 \\ ORG=0.2 \\ \ldots \end{array}\right\} \exists y{:}\left\{\begin{array}{c} LEGAL=0.94 \\ CLOTHES=0.05 \\ DOC=0.004 \\ \ldots \end{array}\right\})[suit'(y) \wedge file_{arg0,arg1}(x,y)]} <
$$

Figure 4.5: Using the type model for disambiguation in the derivation of *file a suit*. Type distributions are shown after the variable declarations. Both *suit* and the object of *file* are lexically ambiguous between different types, but after the β-reduction only one interpretation is likely. If the verb were *wear*, a different interpretation would be preferred.

y combine to a term z:

$$p_z(t) = \frac{p_x(t)p_y(t)}{\sum_{t'} p_x(t')p_y(t')}$$

For example, in *wore a suit* and *file a suit*, the variable representing *suit* may be lexically ambiguous between CLOTHES and LEGAL types, but the variables representing the objects of *wear* and *file* will have preferences that allow the system to choose the correct type when the terms combine. Figure 4.5 shows an example derivation using the type model for disambiguation.

4.4.4 Distributional Relation Clustering

The typed binary predicates can be grouped into clusters, each of which represents a distinct semantic relation. Note that because typed predicates are clustered, $born_{arg0:PER,in:LOC}$ and $born_{arg0:PER,in:DAT}$ can be clustered separately.

4.4.4.1 Corpus Statistics

Typed binary predicates are clustered based on the expected number of times they hold between each argument-pair in the corpus. This means there is a single vector of argument-pair counts for each predicate (not a separate vector for each argument). For example, the vector for the typed predicate $write_{arg0:PER,arg1:BOOK}$ may contain non-zero counts for entity-pairs such as *(Shakespeare, Macbeth)*, *(Dickens, Oliver Twist)* and *(Rowling, Harry Potter)*. The entity-pair counts for $author_{arg0:PER,of:BOOK}$ may be similar, on the assumption that both are samples from the same underlying semantic relation.

To find the expected number of occurrences of argument-pairs for typed binary predicates in a corpus, the type-model is applied to the derivation of each sentence, as described in Section 4.4.3. This outputs untyped binary predicates, with distributions over the types of their arguments. The type of the predicate must match the type of its arguments, so the type distribution of a binary predicate is simply the joint distribution of the two argument type distributions.

For example, if the arguments in a $born_{arg0,in}(obama, hawaii)$ derivation have the respective type distributions *(PER=0.9, LOC=0.1)* and *(LOC=0.7, DAT=0.3)*, the distribution over binary typed predicates is: ($born_{arg0:PER,in:LOC}$=0.63, $born_{arg0:PER,in:DAT}$=0.27, etc.) The expected counts for $(obama, hawaii)$ in the vectors for $born_{arg0:PER,in:LOC}$ and $born_{arg0:PER,in:DAT}$ are then incremented by these probabilities.

	$born_{arg1:PER,in:LOC}$	$birthplace_{poss:PER,be:LOC}$	$born_{arg1:PER,in:DAT}$	$birthdate_{poss:PER,be:DAT}$
(Shakespeare, Stratford)	13.5	4.6	0.6	0.0
(Obama, Hawaii)	29.1	7.3	1.6	0.0
(Jesus, Bethlehem)	87.9	23.5	2.4	0.0
(Napoleon, Corsica)	5.6	1.8	1.1	0.0
(Obama, 1961)	1.2	0.0	37.5	10.1
(Shakespeare, 1564)	0.8	0.0	25.9	8.0

Table 4.3: Hypothetical example vectors for typed predicates (corresponding untyped predicates are shown in Table 4.1). The two senses of $born_{arg1,in}$ are disentangled by splitting the predicate into two typed predicates—one of which is similar to $birthplace_{poss:PER,be:LOC}$, and another which is related to $birthdate_{poss:PER,be:DAT}$

Table 4.3 shows example vectors for several typed predicates, demonstrating how adding types to predicates can help resolve ambiguity.

4.4.4.2 Clustering

As in Section 4.3.2.2, predicates are clustered using the Chinese Whispers algorithm. However, the predicates are now typed, meaning that the graph of predicates can first be decomposed into subgraphs of predicates with the same type. For example, there may be a subgraph for predicates with the (PER, LOC) type (containing predicates such as $live_{arg0,in}$ and $fly_{arg0,to}$) and one for predicates with the (PER, DAT) type. This greatly improves efficiency, as the edge weights only need to be computed for nodes in the same subgraph. It also means the algorithm can easily be parallelized, by having one thread per subgraph.

4.4.5 Semantic Parsing Using Relation Clusters

The final phase is to use the relation clusters in the lexical entries of the CCG semantic derivation. This is slightly complicated by the fact that the predicates are lexically ambiguous between all the possible types they could take, and hence the relations they could express. For example, the system cannot tell whether $born_{arg1,in}$ is expressing a *birthplace* or *birthdate* relation until later in the derivation, when it combines with its arguments. However, all the possible logical forms are identical except for the symbols used, which means the system can output a packed logical form capturing the full distribution over logical forms. To create the packed logical form, the predicates used are functions from argument types to relations.

For each word, the system first finds the lexical semantic definition produced by the algorithm in Section 4.3.1. For binary predicates in this definition (which will be untyped), the system performs a deterministic lookup in the cluster model learnt in Section 4.3.2.2, using all possible corresponding typed predicates. As multiple relations symbols are found for a single untyped predicate, the predicate can be represented as a *packed predicate*: a function from argument types to relations.

For example, if the clustering maps $born_{arg0:PER,in:LOC}$ to *rel49* ("birthplace") and $born_{arg0:PER,in:DAT}$ to *rel53* ("birthdate"), the lexicon contains the following packed lexical entry (type-distributions on the variables are suppressed for brevity):

$$born \vdash (S\backslash NP)/PP[in] : \lambda y \lambda x. \left\{ \begin{array}{l} (x\!: PER, y\!: LOC) \Rightarrow rel49 \\ (x\!: PER, y\!: DAT) \Rightarrow rel53 \end{array} \right\} (x,y)$$

The distributions over argument types then imply a distribution over relations. For example, if the packed-predicate for $born_{arg0,in}$ is applied to arguments *Obama* and *Hawaii*, with respective type distributions *(PER=0.9, LOC=0.1)* and *(LOC=0.7, DAT=0.3)*[6], the distribution over relations will be (*rel49*=0.63, *rel53*=0.27, etc.).

If *1961* has a type-distribution *(LOC=0.1, DAT=0.9)*, the output packed-logical form for *Obama was born in Hawaii in 1961* will be:

$$\left\{ \begin{array}{c} rel49{=}0.63 \\ rel53{=}0.27 \\ \ldots \end{array} \right\} (obama, hawaii) \wedge \left\{ \begin{array}{c} rel49{=}0.09 \\ rel53{=}0.81 \\ \ldots \end{array} \right\} (obama, 1961)$$

The probability of a given logical form can be read from this packed logical form.

4.5 Experiments

The model is evaluated on a question-answering task. Results in Chapter 3 show that the system offers a strong model of formal semantics, capable of sophisticated multi-

[6] These distributions are composed from the type-distributions for both the predicate and argument, as explained in Section 4.4

Type	Top Words
1	suspect, assailant, fugitive, accomplice
2	author, singer, actress, actor, dad
5	city, area, country, region, town, capital
8	subsidiary, automaker, airline, Co., GM
10	musical, thriller, sequel, special

Table 4.4: Most probable terms in some clusters induced by the Type Model.

sentence inferences, and these results are unaffected by the use of clustering. To evaluate the work in this chapter, an evaluation is used that focuses on lexical semantics.

4.5.1 Experimental Setup

The system is trained on Gigaword [Graff et al., 2003], which contains around 4 billion words of Newswire. The corpus was preprocessed to reduce noise.

- Only text occurring within *¡P¿* or *¡TEXT¿* tags was used, and only documents whose type is *story*.
- To filter text such as *(END OPTIONAL TRIM)*, lines containing no lower case letters were ignored.
- Parts of the corpus appear to contain errors where underscores are used instead of commas - these were automatically replaced.
- Some articles start with meta-information, such as: *Y2K-MAIN _ WASHINGTON _*. These are filtered with a regular expression.

The type-model is trained using 15 types[7], and 5,000 iterations of Gibbs sampling (using the distributions from the final sample). Table 4.4 shows some example types. The relation clustering uses only proper nouns, to improve precision (sparsity problems are partly offset by the large input corpus). Aside from parsing, the pipeline takes around a day to run using 12 cores.

[7]This number was chosen by examination of models trained with different numbers of types. The algorithm produces semantically coherent clusters for much larger numbers of types, but many of these are fine-grained categories of people, which introduces sparsity in the relation clustering.

4.5.2 Question Answering Experiments

As yet, there is no standard way of evaluating lexical semantics. Existing tasks like Recognising Textual Entailment [Dagan et al., 2006] rely heavily on background knowledge and coreference resolution, which is beyond the scope of this work. Intrinsic evaluations of entailment relations have low inter-annotator agreement [Szpektor et al., 2007], due to the difficulty of evaluating relations out of context.

The evaluation is based on that performed by Poon and Domingos [2009]. A set of questions is automatically constructed by sampling from text. The evaluation then tests how many correct answers can be found in a different corpus.

From dependency-parsed Newswire (using the MaltParser Nivre et al. [2007]), simple binary relations are sampled using the following patterns: $X \overset{nsubj}{\leftarrow} verb \overset{dobj}{\rightarrow} Y$, $X \overset{nsubj}{\leftarrow} verb \overset{pobj}{\rightarrow} Y$ or $X \overset{nsubj}{\leftarrow} be \overset{dobj}{\rightarrow} noun \overset{pobj}{\rightarrow} Y$ patterns, where X and Y are proper nouns and the verb is not on a list of stop verbs. These patterns are deterministically converted to questions in the present tense. For example, from *Google bought YouTube*, the questions *What does Google buy?* and *What buys YouTube?* are created. To improve the quality of the questions, patterns are only extracted from predicates with a single object, which avoids generating questions like *What gives Michelle?* from *Obama gave Michelle a present.* I also automatically excluded questions where the main predicate is a frequently light verb, or the preposition is *as*, *than* or *like*. While these filters do prune some valid questions, they were found to greatly improve the overall quality of the question set.

The task is to find proper-noun answers to these questions in a different corpus, which are then evaluated by human annotators based on the sentence the answer was retrieved from[8]. Systems can return multiple answers to the same question (e.g. *What did Google buy?* may have many valid answers), and all of these contribute to the result. As none of the systems model tense or temporal semantics, annotators were instructed to annotate answers as correct if they were true at any time. This approach means that relations are evaluated in proportion to corpus frequency. 1000 questions were sampled from the New York Times subset of Gigaword from 2010, and the New York Times from 2009 was used for evaluation. A 50% sample of the output was annotated for the CCG systems.

[8]Common nouns are filtered automatically. To focus on evaluating the semantics, annotators ignored garbled sentences due to errors pre-processing the corpus (these are excluded from the results). Weekday and month answers were also automatically filtered, which are overwhelmingly syntax errors for all systems—e.g. treating *Tuesday* as an object in *Obama announced Tuesday that...*

The following comparison systems were used for evaluation:

- **CCG-Baseline** The logical form produced by the standard CCG derivation, without using clustering.
- **CCG-WordNet** The CCG logical form, plus WordNet as a model of lexical semantics.
- **CCG-Distributional** The logical form including the type model and clusters.
- **Relational LDA** An LDA based model for clustering dependency paths [Yao et al., 2011]. The model was trained on New York Times subset of Gigaword[9], using their setup of 50 iterations with 100 relation types.
- **Reverb** A sophisticated Open Information Extraction system [Fader et al., 2011].

Unsupervised Semantic Parsing [Poon and Domingos, 2009, 2010, Titov and Klementiev, 2011] would be another obvious baseline. However, memory requirements mean it is not possible to run at this scale (the CCG-Distributional system is trained on 4 orders of magnitude more data than the USP evaluation). Yao et al. [2011] found it had comparable performance to Relational LDA.

For the CCG models, rather than performing full first-order inference on a large corpus, the system simply tests whether the question predicate subsumes a candidate answer predicate, and whether the arguments match.This approach is much more efficient than full first-order theorem-proving. Theorem-proving would allow the system to make additional inferences, such as answering *What did Google buy?* from *Google bought the largest video website* and *YouTube is the largest video website.*. The system is able to use the scope of negation, so will not answer the question based on *Google did not buy Apple*.

In the case of CCG-Distributional, the probability is calculated that the two packed-predicates are in the same cluster, marginalizing over their argument types. For example, say the system considers answering the question *What is Obama's birthplace?* from the sentence *Obama was born in Hawaii.* The predicates $born_{arg1,in}$ and $birthplace_{poss,be}$ may map to the same relation cluster with some types (e.g. *(PER,LOC)*), but not with other types such as *(PER,DAT)*. The probability that the inference holds is then the probability that they both have a type where the inference holds, i.e:

[9] This is around 35% of Gigaword, and was the largest scale possible with available resources.

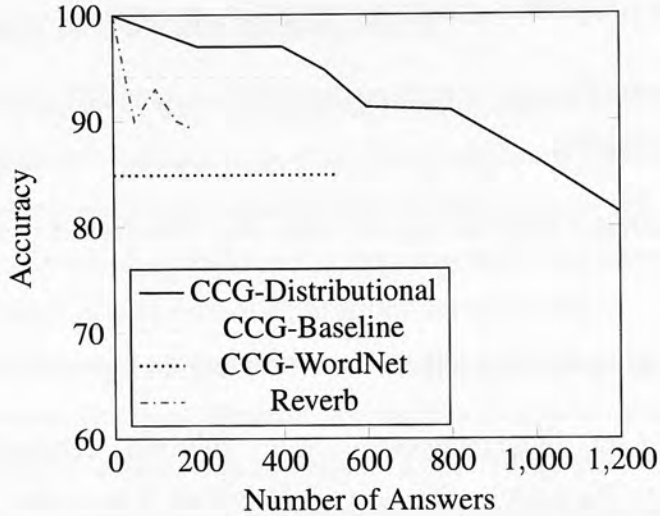


Figure 4.6: Precision at Rank curves for a wide-coverage Question Answering task. It is not possible to give a recall figure, as the total number of correct answers in the corpus is unknown. Relational-LDA is not shown, but found 7046 answers with 11.6% precision.

$$\frac{\sum_{t1,t2} \mathbb{1}_{r(born_{arg1:t1,in:t2})=r(birthplace_{poss:t1,be:t2})} p(t1,t2|context)}{\sum_{t1,t2} p(t1,t2|context)}$$

Where r maps predicates to cluster identifiers, and:

$$\begin{aligned} p(t1,t2|context) = {} & p(t1|obama)p(t1|born_{arg1})p(t1|birthplace_{poss}) \\ & \cdot p(t2|hawaii)p(t2|born_{in}p(t2|birthplace_{be}) \end{aligned}$$

Question/Answer pairs are ranked by this probability. For CCG-WordNet, the system checks if the question predicate is a hypernym or synonym of the candidate answer predicate (using any WordNet sense of either term).

Results are shown in Figure 4.6.

- Relational-LDA induces many meaningful clusters, but all predicates in the corpus must be assigned to one of 100 relation types. As there are far more than 100 different kinds of relations in a large English corpus, the model learns a number of very large, noisy clusters, which dominate the results. It is not possible to take the N-best answers as the cluster assignments do not have a confidence score. Yao et al. [2011] show that these clusters can still be useful in a supervised framework, as a supervised classifier can learn which clusters are useful. However, I believe this result is an argument for using non-parametric clustering.

Question	Answer	Sentence
What presides over House?	Nancy Pelosi	The Democratically-controlled House led by House Speaker Nancy Pelosi , D-Calif. . .
What is a columnist for Denver Post?	Chuck Plunkett	Chuck Plunkett is a runner and an editorial writer for The Denver Post
What is a columnist for Denver Post?	William Porter	William Porter is a reporter for The Denver Post
What lives in India?	Briton	Slym, a Briton who worked for the company in Poland . . . before coming to India . . .
What do Democrates win in?	South Dakota	My Republican parents were shocked that a Democrat could get elected in South Dakota
What reports from Washington	Eric Schmitt	Eric Schmitt contributed from Washington
What moves from Boston?	Ramirez	The Dodgers sold 30 000 tickets the day after Ramirez arrived from Boston in a trade
What does Dalai Lama arrive in?	Taiwan	Ma reluctantly allowed the Dalai Lama, the spiritual leader of the Tibetans, to visit Taiwan
What meets with Obama?	Gates	Obama who has already discussed military spending with Gates. . .
What serves in Senate?	Burris	Burris . . . hopes to begin working in the Senate the next day
What meets with Obama?	Henry Gates	Obama has invited Harvard Professor Henry Gates . . . to the White House
What leaves for Harvard?	Hatch	Hatch went back to Harvard
What arrives in Washington?	Price	Lewis and Price flew to Washington to meet with officials from the Federal Reserve
What runs for Congress	Coffin	Coffin was elected to Congress two years later and served two terms
What serves in House	Jeff Flake	. . . said Rep. Jeff Flake who has sponsored bipartisan immigration reform in the House
What speaks with Hu Jintao?	Geithner	Geithner also was scheduled to meet Tuesday with President Hu Jintao
What speaks with Hillary Clinton?	Karzai	Hillary Clinton bluntly told Karzai that running with Fahim would damage his standing with the United States
What returns from China	Nixon	. . . the ping-pong diplomacy preceding President Nixon 's historic visit to China in 1972

Table 4.5: Example questions answered by CCG-Distributional that could not be answered by the baseline CCG system. Some of these are rely on erroneous clustering—for example, equating *running for Congress* with being *elected to Congress*.

- The two deterministic systems, CCG-Baseline and Reverb systems both achieve good precision, with the CCG system able to improve recall based on long-range dependencies and coordination constructions. The CCG-Baseline errors are mainly caused by parser errors, or relations in the scope of non-factive operators. For example, it believed *Mexico is struggling to qualify for the 2010 World Cup→Mexico qualifies for the World Cup*, because it did not understand that *struggling* is non-factive. This issue is addressed further in Chapter 6. There were also a number of errors due to misidentifying named-entities.

- CCG-WordNet adds relatively few correct answers to CCG-Baseline, reflecting the limitations of hand-built ontologies. It loses precision compared to CCG-Baseline, mostly due to us not modelling word-senses. For example, it inferred *Randolph saw combat in Vietnam → Randolph met in Vietnam* , because one WordNet sense of *meet* is in the same synset as *see*. We could in principle use word-sense disambiguation to resolve this problem, but resolving WordNet senses is notoriously difficult [Hovy et al., 2006]. In any case, recall is still far lower than for the distributional approach.

- CCG-Distributional substantially improves recall over other approaches whilst retaining good precision, demonstrating that that our system has learnt a powerful model of lexical semantics. Table 4.5 shows some correctly answered questions. The system improves over the baseline by mapping expressions such as *merge with* and *acquisition of* to the same relation cluster. Many of the errors (and, in fact, successes) are caused by conflating predicates where the entailment only holds in one direction, such as *was elected to* with *ran for*—meaning that many of the clusters do not truly correspond to underlying semantic concepts. Chapter 6 introduces a much more sophisticated approach to clustering to solve this problem.

4.5.3 Qualitative Evaluation

The answers returned by the system were inspected to determine the contribution of different parts of the model. The use of formal semantics helps the CCG-based models to improve both precision and recall over alternative approaches. However, a weak model non-factive verbs is a major source of errors.

Using formal semantics helps the CCG models to easily return answers that would be difficult for the other systems, by capturing non-local arguments and normalizing syntactic variations. The most frequent examples are in relative clauses and coordination. For example, the CCG models correctly answer a question by making the inference *Kuklo, who is currently an associate professor of medicine at Washington University→Kuklo is a professor at Washington University*. Answering the question would require additional inference for an approach based on syntactic dependencies, as the relative clause changes the dependencies between *Kuklo* and *professor*. Similarly, for the question *What does GM focus on?*, the CCG models return all four answers from the sentence *GM ... is focusing on Chevrolet, Cadillac, Buick and GMC*—because the use of logical form allows the same relationship between all the conjuncts and the predicate (as explained in Chapter 3). Syntactic dependency trees give a less elegant account of coordination, as only one of the conjuncts can be the head—meaning that the other conjuncts have a different relation to the predicate. These examples show how the use of CCG can improve recall over using syntactic dependencies, by abstracting over different syntactic ways of expressing the same meaning.

The use of formal semantics allows the CCG models to identify negated predicates, which helps improve precision over other approaches. For example, Reverb incorrectly answers the question *Who testifies before Congress?* with *Obama* based on the sentence *Susan Sher ... emphasized that Obama would not testify before Congress*, because the relatively simple pattern matching ignores adverbs (including *not*). However, the CCG models avoid that mistake by detecting that the verb *testify* is within the scope of negation. This shows how formal semantics can boost the precision of question answering systems.

We find many errors to do with failing to model non-factive predicates. For example, the CCG approaches license inferences such as *Mexico is struggling to qualify for the 2010 World Cup→Mexico qualifies for the World Cup*, *Burris plans to arrive in Washington→Burris arrives in Washington* and *GM is expected to focus on China→GM focuses on China*, leading to precision errors. A more detailed consideration of implicative verbs is given in Chapter 6, which partially addresses these issues.

4.6 Comparison with Related Work

This section compares the approach outlined in this chapter with a range of recent work in computational semantics. More detailed descriptions of alternative approaches can be found in Chapter 2.

4.6.1 Unsupervised Semantic Parsing

Unsupervised Semantic Parsing [Poon and Domingos, 2009, 2010, Titov and Klementiev, 2011] is a closely related approach to the Distributional CCG model, in that it builds a logical form using cluster identifiers as symbols. It has both strengths and weakness compared to the model developed here.

USP clusters a greater range of predicates than the system developed in this thesis. Whereas my work has concentrated on binary relations, USP also clusters other kinds of predicates, such as adverbs. However, the evaluation of the system only focuses on nouns and binary predicates, so it is unclear how effective their methods are on other kinds of relations. USP gives a Davidsonian analysis of multi-argument relations, rather than the simple alternative here of binarizing predicates with more than 2 arguments. USP's approach has advantages when comparing a binary relation to a unary one, for example *I walked to work*→*I walked*. In contrast, our current ambiguity model would not work in this setting (USP assumes all words are unambiguous). Binarizing also makes it easier to capture relations between equivalent expressions with different syntactic frames, such as in the equivalence between *X bought Y* and *Y was sold to X*. USP is likely to assign the verbs *buy* and *sell* to unrelated clusters, as it assumes argument keys are conditionally independent given a cluster.

On significant limitation of USP is that it has extremely high memory requirements, meaning it can only be applied to small corpora. Yao et al. [2011] found it required 45GB of RAM to run on just 1000 news articles. Reassigning a predicate to a new cluster affects the probabilities of all sentences containing that predicate. The CCG system is much more computationally efficient, as rather than trying to find a clustering that maximizes the probability of all the sentences in the corpus, every predicate is transformed into a vector based on local context, and then clustered based on these vectors.

4.6.2 Compositional Vector Space Models

Section 2.4.1 discussed several challenges for current compositional vector space models—including representing logical operators, composing word meanings for expressions longer than a few words, representation of factual knowledge, and dealing with complex syntactic constructions such as coordination. The CCG model developed in Chapter 3 offers good solutions to these issues, and this chapter developed it by including distributional representations of content words. However, vector space models do have some advantages over the current work, which I discuss here.

Both the present approach and compositional vector space models start from vectors representing non-compositional units. Vector space approaches typically use context words as dimensions [Baroni et al., 2013], whereas the CCG model uses argument entities. It is reasonable to expect that the context word approach is less sparse, though the statistics are also potentially less informative. Of course, which gives the better performance is an empirical question. The model of developed in Chapter 6 takes advantage of both.

Baroni et al. [2013] make an extended case for compositional vector space models. They view them as complementary to logical models, but argue they are superior in a number of ways. Most of their criticisms of logical models—including the size of the lexicon and modelling ambiguity—are handled by the distributional CCG approach (by non-parametric clustering and a probabilistic model of ambiguity).

One advantage of compositional vector space models over the current proposal is in their ability to handle compositionality involving multiple content words—for example, capturing that *dog house=kennel*. The predicates clustered in the CCG model all contain a single content word. The most simple solution to this in the CCG model would be to treat such items as multi-word expressions, and cluster them based on the composed vectors for individual words. Section 6.7.4 discusses this point in more detail, and gives an alternative solution.

Baroni et al. [2013] suggest that another advantage of vector space models is the ability to capture *near paraphrases*—expressions which are strongly related, but not truth-conditionally equivalent. For example: *The workers are stressed≈The workers are busy*[10]. The current approach would deny that the sentences imply each other, whereas a vector-space approach is likely to find correlations between them. The distributional CCG system could be extended to capture such inferences, by using a soft-clustering model such as a Hierarchical Dirichlet Process [Teh et al., 2006] (instead of the hard clustering given by the Chinese Whispers algorithm). This model would allow the semantics of typed predicates to express a distribution over clusters, which could capture that the *stressed* utterance was generated by the *busy* cluster with non-zero probability. It is somewhat unclear what the applications of such reasoning are—near-paraphrasing is probably insufficiently high precision for question-answering applications, although it may be useful for information retrieval.

[10]The example was suggested by Chris Manning

4.6.3 Natural Logic

Natural Logic [MacCartney and Manning, 2007] circumvents many of the problems associated with building full logical forms for sentences, by simply annotating the scope of negation in text. MacCartney and Manning [2007] point out that there are many obstacles to building full semantic interpretation, including *idioms, ellipsis, paraphrase, ambiguity, vagueness, lexical semantics, the impact of pragmatics, and so on.* Whilst the present work attempts to address some of these issues, there are clearly many outstanding problems. The simpler approach of natural logic aims to provide more robust inference than full semantic interpretation, whilst retaining the ability to reason about phenomena such as negation and quantifiers.

On the other hand, the insertion and deletion rules used by Natural Logic for inference limits the potential for certain kinds of entailment, as words must appear in the same order in the premise and hypothesis. For example, it would not be able to infer: *Google bought YouTube* $\Longrightarrow$ *YouTube was bought by Google.* Traditional formal semantics approaches can avoid this problem, as mapping to a traditional logical form abstracts away from the original word order. Natural Logic also cannot make inferences involving multiple premises. MacCartney and Manning [2007] relies on WordNet for a model of lexical semantics, which is likely to limit the inferences it can make. It may be possible to create a *Distributional Natural Logic* using similar techniques to those developed in this thesis.

4.6.4 Distributional Inference Rules

Garrette et al. [2011], Beltagy et al. [2013] introduced an approach in which the output of a CCG system is augmented with distributionally-induced inference rules (see Section 2.4.3.2 for a more detailed introduction). This approach is the most closely related to ours in both aims and methods. However, there are several important differences.

The most important distinction is in how the lexical semantics is expressed. Garrette et al. [2011]'s model needs to create an axiom between each pair of content words in the corpus (though low-probability rules are pruned), and the number of axioms required will grow quadratically in the size of the corpus. In contrast, the system developed here grounds the meaning of words in cluster identifiers, so the size of the logical form grows linearly in the size of the corpus. For example, if a corpus contains four words with high distributional similarity, such as *buy*, *purchase*, *acquire* and *take-over*, Garrette et al. [2011] would create 9 axioms (*buy*$\rightarrow$purchase, *buy*$\rightarrow$acquire, *purchase*→acquire, etc.). The clustering approach would simply have all terms express the same cluster identifier. Using clustering may also reduce sparsity; rarer terms such as *take-over* may have low distributional similarity with some of their synonyms, but clustering may still allow the relation to be identified.

The distributional inference rule approach is based on the idea that lexical semantics is inherently fuzzy, and should be dealt with probabilistically. A consequence of the clustering approach is that words are treated as being synonyms or unrelated (Chapter 6 generalizes this approach to handle entailments that only hold in one direction, but the approach is still discrete). The use of a probabilistic typing model means that our system is probabilistic, but conceptually words are ambiguous between discrete meanings, rather than themselves having fuzzy interpretations.

Our approach could equivalently be formulated using probabilistic inference rules, by making each word express a unique semantic primitive (so each word instance has a unique meaning, based on its context), add adding probabilistic inference rules expressing a distribution over cluster identifier symbols. For example:

$p(buy456(google, youtube) \leftrightarrow cluster47(google, youtube) = 0.9$
$p(buy456(google, youtube) \leftrightarrow cluster186(google, youtube) = 0.01$
$p(purchase423(google, youtube) \leftrightarrow cluster47(google, youtube) = 0.85$
$p(acquire768(google, youtube) \leftrightarrow cluster47(google, youtube) = 0.95$
$p(take\_over867(google, youtube) \leftrightarrow cluster47(google, youtube) = 0.93$

Because words express distributions over cluster identifiers, rather than distributions over other words, this approach asymptotically reduces the number of inference rules required. Ignoring pruning, the number of inference rules added for understanding a corpus is the product of the number of content words and the number of types (rather than being quadratic in the number of words). This approach may be useful for efficiency, as probabilistic theorem proving is expensive.

Of course, these approaches are not mutually exclusive—one way to hybridize them would be to represent synonyms using clusters, but also add axioms to represent fuzzy inference rules between clusters, which would capture the advantages of both.

4.7 Conclusions

This chapter has introduced a new model which combines many of the advantages of formal and distributional semantics, by modelling the meanings of content words with distributionally induced cluster identifiers. I have demonstrated that the combined model is capable of both making complex logical inference involving quantifiers, and answering questions that require knowledge of lexical semantics. However, there is much potential for enhancing the current model, and it will be greatly improved in Chapter 6.

CHAPTER 5

Unsupervised Induction of Cross-lingual Semantic Relations

5.1 Introduction

This chapter attempts to generalise the work of Chapter 4 to multiple languages. It explores the strongest hypothesis presented in this thesis, which speculates that the interpretation of all languages can be expressed using the same set of predicates, and that these predicates can learnt from unlabelled text. Chapter 4 suggested that the clusters represent abstract concepts—are they abstract enough to be shared between languages? To investigate this, I build clusters containing both French and English predicates, using the fact that named-entities are similar between languages to guide the clustering. The work inevitably remains somewhat preliminary, as there is little existing work in a potentially large field, but encouraging results suggests that the direction is worth pursuing. The work in this chapter was previously published as Lewis and Steedman (2013b).

The rest of the chapter proceeds as follows:

- Section 5.2 discusses the motivations for attempting to induce cross-lingual semantic clusters.

- Section 5.3 gives an overview of the proposed method.
- Sections 5.4 and 5.5.1 discuss how predicates and entities are represented to allow the cross-lingual clustering.
- Sections 5.7 and 5.8 evaluate the method, showing good results on question-answering and translation reranking experiments.
- Section 5.9 describes how this work is related to other recently proposed methods in machine translation and cross-lingual semantics.

5.2 Motivation

Identifying a language-independent semantics is a major long term goal of computational linguistics, and is interesting both theoretically and for practical applications. Interlingual machine translation [Dorr et al., 2004] assumes that semantically equivalent sentences in any language can be mapped onto a common meaning representation. Such a representation would be of great utility for tasks such as translation, relation extraction, summarization, question answering, and information retrieval. Regardless of whether it is even possible to create such a semantics, I show that even an incomplete version can be useful for downstream tasks.

Semantic machine translation aims to map a source language to a language-independent meaning representation, and then generate the target language translation from this. It is hoped this would alleviate the difficulties of simpler models when translating between languages with very different word ordering and syntax [Vauquois, 1968]. It would also avoid the problem of needing parallel text in every pair of languages to be translated, as is required by current approaches. Instead, each language only requires a single mapping into and out of the interlingual representation.

Despite many attempts to create interlingual representations [Mitamura et al., 1991, Beale et al., 1995, Banarescu et al., 2013], state-of-the-art machine translation still uses phrase-based models [Koehn et al., 2007]. The major obstacle to defining interlinguas has been devising a meaning representation that is language-independent, but capable of expressing the limitless number of meanings that natural languages can express [Dorr et al., 2004].

I introduce an approach that avoids this problem, by utilising the methods of distributional semantics. The work presented in Chapter 3, and several other recent papers

[Poon and Domingos, 2009, Yao et al., 2011], has shown that paraphrases of expressions can be learnt by clustering those with similar arguments —for example learning that *X wrote Y* and *X is the author of Y* are equivalent if they appear in a corpus with similar *(X, Y)* argument-pairs such as $\{$*(Shakespeare, Macbeth), (Dickens, Oliver Twist)*$\}$. In this chapter, I extend this to the multilingual case, aiming to also map the French equivalents *X a écrit Y* and *Y est un roman de X* on to the same cluster as the English paraphrases. Conceptually, I treat a foreign expression as a paraphrase of an English expression. The cluster identifier can be used as a predicate in a logical form, suggesting that the fundamental predicates of an interlingua can be learnt in an unsupervised manner via clustering.

In this chapter I focus on learning binary relations between named entities. This problem is much simpler than attempting complete interlingual semantic interpretation, but in Section 5.10 I suggest how it could be generalized. This class of expressions has proved extremely useful in the monolingual case, with direct applications for question answering and relation extraction [Poon and Domingos, 2009, Mintz et al., 2009], and I demonstrate how to use them to improve machine translation. It is important to be able to extract knowledge across languages, as many facts will not be expressed in all languages—either due to less-complete encyclopedias being available in some languages, or facts being most relevant to a single country.

In contrast to most previous work on machine translation and cross-lingual clustering, the proposed method requires no parallel text (see Section 5.9 for discussion of some exceptions). It instead exploits an alignment between named-entities in different languages. The limited size of parallel corpora is a significant bottleneck for machine translation [Resnik and Smith, 2003], whereas the clustering approach can be used on much larger monolingual corpora. This means it is potentially useful for language-pairs where little parallel text is available, for domain adaptation, or for semi-supervised approaches.

The other motivation for this work is to explore whether cross-lingual clustering can induce better mono-lingual clusters than simply running on one language alone. The intuition here is that seeing the same concept expressed multiple languages provides stronger evidence that it is really an underlying relation. Another possibility is that using multiple languages may allow the use of parallel text as a way to supervise clustering—effectively treating a parallel corpus as a huge paraphrase corpus. In practice, I did not find that adding multiple languages helped improve the clustering in individual languages, but in Section 5.10 I discuss possibilities for overcoming this obstacle.

5.3 Overview of Approach

This chapter builds on clustering-based approaches to monolingual distributional semantics, such as that of Chapter 4, aiming to create clusters of semantically equivalent predicates based on their arguments in a corpus. In each language, each sentence in a large monolingual corpus is first deterministically mapped onto a simple logical form, by extracting binary predicates between named entities. Then, predicates with similar arguments are clustered both within and between languages.

When parsing a new sentence, instead of using the monolingual predicate, the cluster identifier is used as a language-independent semantic relation, as shown in Figure 5.1. The resulting logical form can be used for inference in question answering.

Unlike traditional approaches to translation, this method does not require parallel text—but it does impose some additional constraints on language resources. The approach requires:

- A large amount of factual text, as the approach relies on the same facts being expressed in different languages. I use Wikipedia, which contains articles in 250 languages, including 121 with at least 10,000 articles.[1] Other domains, such as Newswire, may also be effective. However, the method would probably not be successful for works of fiction.

- A method for extracting binary relations from sentences. This is straightforward from dependency parses, which are available for many languages. It is also possible without a parser, with some language-specific work [Fader et al., 2011]. The approach is described in Section 5.4.

- A method for linking entities in the training data to some canonical representation. McNamee et al. [2011] report good results on this task in 21 languages. A simple method for entity linking is described in Section 5.5.1.

5.4 Predicate Extraction

The proposed method relies on extracting binary predicates between entities from sentences. Various representations have been suggested for binary predicates, such as Re-

[1] As of June 2013.

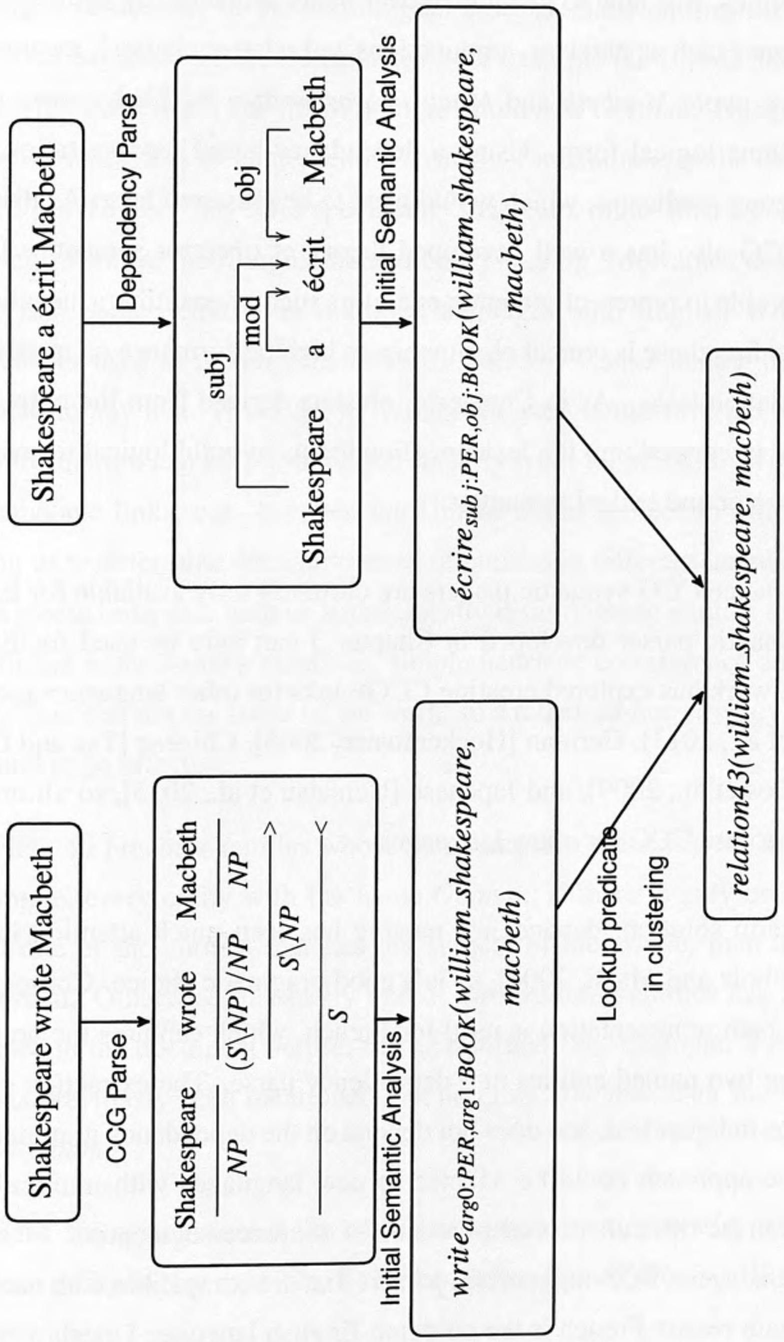


Figure 5.1: Example showing how the system can map sentences in different languages to the same meaning representation, assuming that $write_{arg0:PER,arg1:BOOK}$ and $écrire_{subj:PER,obj:BOOK}$ have been clustered.

verb patterns [Fader et al., 2011], dependency paths [Lin and Pantel, 2001, Yao et al., 2011], and binarized predicate-argument relations derived from a CCG-parse [Lewis and Steedman, 2013a]. The approach suggested here is formalism-independent, and is compatible with any method of expressing binary predicates.

For English, the CCG parser developed in Chapter 3 is used. It outputs a logical form derived automatically from the CCG-parse, containing predicates such as: $write_{arg0,arg1}(shakespeare,macbeth)$. By using the close relationship between the CCG syntax and semantics, it is able to generalize over many semantically equivalent syntactic constructions (such as passives, conjunctions and relative clauses), meaning that both *Shakespeare wrote Macbeth* and *Macbeth was written by Shakespeare* can be mapped to the same logical form. Using a dependency-based representation, these would have different predicates, which would need to be clustered later. As discussed in Chapter 3, CCG also has a well developed theory of operator semantics [Steedman, 2012], so is able to represent semantic operators such as quantifiers, negation and tense—understanding these is crucial phenomena to high performance on question answering or translation tasks. As in Chapter 4, clusters derived from the output from the parser can be integrated into the lexicon, allowing us to build logical forms which capture both operator and lexical semantics.

High performance CCG syntactic parsers are currently only available for English, meaning the semantic parser developed in Chapter 3 can only be used for English. However, recent work has explored creating CCGBanks for other languages including Hindi [Ambati et al., 2013], German [Hockenmaier, 2006], Chinese [Tse and Curran, 2010], Italian [Bos et al., 2009], and Japanese [Uematsu et al., 2013], so ultimately it may be possible to use CCG for many languages.

As a short term solution, dependency parsing has seen much attention in many languages [Buchholz and Marsi, 2006], so is a good pragmatic choice. Consequently, the dependency path representation is used for French, which captures the nodes and edges connecting two named entities in a dependency parse. The extraction of these paths is language-independent, and does not depend on the dependency grammar used, which means the approach could be adapted to new languages with minimal work. In practice, it can be difficult to find pipelines of sentence segmenters, tokenizers, morphological analyzers, POS-taggers and parsers that are compatible with each other - which is the main reason French is the only non-English language I used.

5.5 Entity Semantics

5.5.1 Entity Linking

As discussed in Section 5.3, the approach assumes that semantically similar predicates will have similar argument entities. This requires it to be able to identify coreferring entities across languages during training. In the monolingual case, it suffices to represent entities by the string used in the sentence, as was done in Chapter 4. String matching is inadequate in the multilingual case, as many entities may be referred to by different names in different languages—for example the *United States* translates as *les États-Unis* in French and *die Vereinigte Staaten* in German. This problem is worsened by the ambiguity of named-entity strings—for example, in the context of a sports article, *United States* may refer specifically to a team, rather than a country.

Recent work on multilingual named-entity linking [McNamee et al., 2011] shows how to link named entities in multiple languages onto English Wikipedia articles, which can be used as unique identifiers for entities. Consequently, the method could be applied to any text. However, as Wikipedia itself is used for the training corpora, entity information can be bootstrapped directly from its markup. Wikipedia contains cross-language links, e.g. between the United States articles in different languages, allowing us to determine the equivalence of entities in different languages.

Wikipedia links also help us automatically disambiguate entities to a given article. For unlinked named-entity mentions, simple heuristic co-reference and entity linking is used. This was not the focus of the work, so a rather ad-hoc strategy was used, but it was found to be effective:

- First, all Freebase entities whose name matches the string are returned—for example, every entity with the name *Obama*. If there is only one such entity, or if one of the entities matches the subject of the article, then this entity is returned. Otherwise, if exactly one of the matching entities has previously been used in the document before, that is returned. For example, if *Michelle Obama* has previously been mentioned, but no other *Obama*s, then she will be matched by *Obama*.

- Next, the system sees if the name has non-zero word overlap with a previously seen entity — for example, *Barack* matches *Barack Obama*. If so, that entity is returned.

- The system then checks if the entity name defaults to a particular Wikipedia article (and then returns the corresponding Freebase entity). For example, while there are articles about several different *Obamas* on Wikipedia, the title *Obama* redirects to the *Barack Obama* article. This method captures the intuition that even though most names are ambiguous, there is often a sensible 'default' interpretation.
- Finally, if the entity begins with a title (e.g. *Mrs*, *President* etc.), it tries removing it, and running the coreference algorithm recursivel on the result.

We also ignore entities matching the Freebase 'language' type, which were problematic for the above algorithm (which would incorrectly identify *English* as a language in *I am English*).

While Wikipedia information is useful for co-reference and linking, it of course does not mean that our approach is only applicable to the Wikipedia corpus.

5.5.2 Entity Typing

It has become standard in clustering approaches to distributional semantics to assign types to predicates before clustering, and only cluster predicates with the same type [Schoenmackers et al., 2010, Berant et al., 2011, Yao et al., 2012]. Typing predicates is useful for resolving ambiguity—for example the phrase *born in* may express a place-of-birth or date-of-birth relation depending on whether its second argument has a *LOC* or *DAT* type. Doing this is particularly imporant when clustering cross-lingually, as ambiguous expressions may translate differently in other languages. For example, the two interpretations of *was born in* translate in French as *est né à* and *est né en* respectively. The type of a predicate is determined by the type of its arguments, and predicates with different types are treated as distinct.

In Chapter 3, an unsupervised model of entity types was induced using Latent Dirichlet Allocation [Blei et al., 2003], based on selectional preferences of verbs and argument-taking nouns. When applied cross-linguistically, I found this technique tended to create language-specific topics. As a simple alternative to inducing types, I use an existing type-schema. I exploit the fact that many Wikipedia entities are linked to the Freebase database, which has a detailed manually-built type-schema. This means that the system can look up the Freebase types of many Wikipedia entities.[2] The simplified type-set of 112 types created by Ling and Weld [2012] is used, as it cleaner and

[2]Named entities not present in Freebase are ignored during training.

contains less duplication than the full Freebase schema. Where entities have multiple types (for example, Shakespeare is both an *author* and a *person*), a separate relation is created for each type.

5.6 Relation Clustering

Predicates are clustered into those which are semantically equivalent, based on their argument-pairs in a corpus. The initial semantic analysis is run over the corpora, and for each predicate a vector is built containing counts for each of its argument-pairs (these counts are divided by the overall frequency of an argument-pair in the corpus, so that rarer argument-pairs are more significant). These vectors are used to compute similarity between predicates.

First, the clustering algorithm is run on each language independently, and then the clusters are aligned. Duc et al. [2011] and Täckström et al. [2012] use similar two-step approaches. Running the clustering on both languages simultaneously was found to produce many clusters only containing predicates from a single language. This appears to be because even if predicates in two different languages are truth-conditionally equivalent, the language biases the sample of entity-pairs found in a corpus. For example, the French verb *écrire* may contain more French author/book pairs than the English equivalent *write*. This difference can make the verbs appear to represent different predicates to the clustering algorithm. The two-step approach also means that advances in monolingual clustering should directly lead to improved cross-lingual clusters.

5.6.1 Monolingual Clustering

As in Chapter 4 the Chinese Whispers algorithm [Biemann, 2006] is used for monolingual clustering. As before, the advantages are that the algorithm is simple, non-parametric (meaning that the number of relation clusters is induced from the data), and highly scalable. A separate graph is created for each type of predicate in each language—for example, predicates between types *AUTHOR* and *BOOK* in French (so only predicates with the same type will be clustered). One node is created per predicate in the graph, and edges represent the distributional similarity between the predicates.

The distributional similarity between a pair of predicates is calculated as the cosine-similarity of their argument pair vectors in the corpus. Many more sophisticated approaches to determining similarity have been proposed [Kotlerman et al., 2010, Weisman et al., 2012], and future work should explore these. To reduced noise, the system prunes nodes with less than 25 occurrences, edges of weight less than 0.05, and a short list of stop predicates. These parameters prune considerably more predicates and edges than those used in Chapter 4, and reflect the difficulty of building good cross-lingual clusters. Many of the French dependency paths do not have a clear semantic interpretation (dependency paths appear to be noisier than CCG predicates), so the additional requirements are added that dependency paths contain at least one content word, contain at most 5 edges, and that one of the dependencies connected to the root is subject, object or the French preposition *de*.

5.6.2 Cross-lingual Cluster Alignment

A simple greedy procedure is used to find an alignment between the monolingual clusters in different languages. First, the entity-pair vectors for each predicate in a relation cluster are merged, creating a single 'super-predicate' subsuming all the predicates in the monolingual cluster. Then, the cosine similarity between entity-pair vectors for clusters in different languages is calculated—based only on argument-pairs that occur in both languages, to reduce the potential bias of some entities being more relevant to one language. Clusters are then greedily aligned, in order of their similarity, as in Algorithm 2 (pruning similarities less than 0.01). This means that clusters are aligned with their most similar foreign cluster. Only clusters with the same types are considered for alignment.

Data: Sets of monolingual relation clusters R_{L1} and R_{L2}
Result: An alignment between the monolingual clusters A
$A \longleftarrow \{\}$;
while $R_{L1} \neq \{\} \wedge R_{L2} \neq \{\}$ **do**
  $(r1, r2) \longleftarrow \underset{(r_1,r_2)\in R_{L1}\times R_{L2}}{\arg\max} sim(r1, r2)$;
  $A \longleftarrow A \cup \{(r1, r2)\}$;
  $R_{L1} \longleftarrow R_{L1}/\{r1\}$;
  $R_{L2} \longleftarrow R_{L2}/\{r2\}$;
end

Algorithm 2: Cluster alignment algorithm

An efficient implementation of this algorithm is possible using a priority queue. First, the similarity matrix of relations is pre-computed, and then converted into a priority queue of $(r1, r2)$ pairs. In each iteration of the loop, the highest scoring $(r1, r2)$ pair is removed from the queue. If neither $r1$ nor $r2$ have yet been aligned (i.e. they are still in the sets R_{L1} and R_{L2} respectively), then $(r1, r2)$ is added to A, and R_{L1} and R_{L2} are updated. The complexity of building the priority queue is $\mathcal{O}(n \log n)$, where $n = |R_{L1}| \times |R_{L2}|$.

5.7 Cross-lingual Question Answering Experiments

The system is evalutated on English and French, using Wikipedia for corpora. The English corpus is POS-tagged and CCG-parsed with the C&C tools [Clark and Curran, 2004]. The French corpus is tagged with MElt [Denis et al., 2009] and parsed with MaltParser [Nivre et al., 2007], trained on the French Treebank [Candito et al., 2010]. Wikipedia markup is filtered using Wikiprep [Gabrilovich and Markovitch, 2007]—replacing internal links with the name of their target article, to help entity linking. Some example clusters learnt by the model are shown in Table 5.1. The cross-lingual clusters typically contain more French expressions than English. One explanation is that the English corpus is substantially larger than the French, so the predicates have more observations, making large English clusters appear dissimilar to the French simply because they cover a wider range of arguments. Using more sophisticated similarity metrics than cosine may help address this limitation. Adjusting the parameters in Section 5.6 results in larger clusters, but introduces noise. Despite these weaknesses, the clusters have learnt to identify a wide range of concepts across languages with no supervision.

5.7.1 Experimental Setup

The system is evaluated on a cross-lingual question answering task, similar to monolingual QA evaluations by Poon and Domingos [2009] and in Chapter 4. A question is asked in language *L*, and is answered by the system from a corpus of language *L'*. Human annotators are shown the question, answer entity, and the sentence that provided the answer, and are then asked whether the answer is a reasonable conclusion based on the sentence. Whilst this task is much easier than full translation, it is both a practical application for the approach, and a reasonably direct extrinsic evaluation for the cross-lingual clusters.

As in Chapter 4 and Poon and Domingos [2009], the question dataset is automat-

English	**French**
X invades Y	X envahit Y invasion de Y par X
X orbits Y	X est un satellite de Y X est une lune de Y
X is a skyscraper in Y	X est un gratte-ciel de Y
X is a novel by Y	X est un roman de Y
X joins Y X is a member of Y	X adhère à Y X entre dans Y X rejoint Y

Table 5.1: Some example cross-lingual clusters. Predicates are given in a human-readable form, and predicate types are suppressed.

ically generated from the corpus. This approach has the advantage of evaluating on expressions in proportion to their corpus frequency, so understanding frequent expressions is more important than rare ones. Then 1000 questions are sampled for each language, by extracting binary relations matching certain patterns ($X \stackrel{nsubj}{\leftarrow} verb \stackrel{dobj}{\rightarrow} Y$, $X \stackrel{nsubj}{\leftarrow} verb \stackrel{pobj}{\rightarrow} Y$ or $X \stackrel{nsubj}{\leftarrow} be \stackrel{dobj}{\rightarrow} noun \stackrel{pobj}{\rightarrow} Y$), and removing one of the arguments. For example, from the sentence *Obama lives in Washington* the questions *X lives in Washington?* and *Obama lives in X?* are created.[3] Answers are judged by fluent bilingual humans, and do not have to match the entity that originally instantiated *X*. Multiple answers can be returned for the same question.

The implementation attempts this task by mapping both the question and candidate answer sentences (which will be in a different language to the question) on to a logical form using the clusters, and determining whether they express the same relation. This tests the ability of the approach to cluster expressions into those which are semantically equivalent between languages. It is possible for entities to have multiple types (see Section 5.5.2), and answers are ranked by the number of types in which the entailment relation is predicted to hold.

[3]Questions are given in a declarative form, to make the tasks simpler for the machine translation baseline. The machine translation performed poorly on questions such as *What is Obama the president of?*, as inverted word-orders and long-range dependencies are difficult to handle with re-ordering models and language models (though are straightforward to handle for a CCG system [Clark et al., 2004]). The machine translation was found to perform much better on declarative equivalents, such as: *Obama is the president of X.*

5.7.2 Baseline

The baseline makes use of the Moses machine translation system [Koehn et al., 2007], and is similar to previous approaches to cross-lingual question answering such as Ahn et al. [2004]. I trained a Moses model on the Europarl corpus [Koehn, 2005]. First, the question is translated from language L to L', taking the 50-best translations. As the questions are typically shorter than corpus sentences, this is substantially easier for the machine-translation than translating the corpus. These are then parsed, and patterns are extracted (as in Section 5.4). To avoid penalizing the translation system for failing to translate named-entities that have not been seen in its training data, the Freebase named-entity translation is automatically supplied. These patterns are then used to find answers to the questions. Answers are ranked by the score of the best translation that produced the pattern. Figure 5.2 illustrates this pipeline.

The choice of languages is very favourable to the machine-translation system; English and French have similar word-order, and there is a large amount of parallel text available [Koehn and Monz, 2006]. The clustering system is insensitive to word-order, and does not require parallel text for training, so it is reasonable to expect better performance relative to machine-translation on other language pairs[4]. Future work will experiment with more diverse languages. The sentences to be translated are also very short, reducing the potential for error. On the other hand, Wikipedia text is out-of-domain for the machine translation system.

5.7.3 Results

Results are shown in Table 5.3. Accuracy for each system is based on a sample of 100 answers from its output. Unsurprisingly, the machine-translation has high accuracy on this task, given the choice of languages and the short queries. Pleasingly, the clusters achieve similar accuracy to machine-translation, with much greater recall, with no usage of parallel text.

On examining the results, I found that the distribution of answers is highly skewed for all systems, with many answers to a smaller number of questions (multiple answers can be returned to the same question). This is due to the Zipfian nature of language, the difficulty of the task (which is far from solved in the monolingual case), and the possibility that questions may have no answers in the foreign corpus. This is particularly

[4]On the other hand, my system does rely on large corpora and reliable NLP tools, which are not available for all languages.

Question	**Answer**
X dies in Moscow	**Sergueï Guerassimov** meurt d'une crise cardiaque le mardi 26 novembre 1985 à Moscou
Germany invades X	... depuis l'invasion de la **Pologne** par l'Allemagne et l'URSS
X wins the FA Cup	**Portsmouth FC** remporte la FA Challenge Cup en s'imposant en finale face à Wolverhampton Wanderers FC
X is a band from Finland	**Yearning** est un groupe Finlande de doom metal atmosphérique
X travels to Paris	Après un court séjour en Australie avec son épouse, **Kerensky** revient à Paris en 1949
X leaves London	**Bulwer** quitte Londres en 1839 et retourne dans le Norfolk
X is a municipality in Brazil	**Divinolândia** est une municipalité Brésil de la Microrégion de São João da Boa Vista
X vit en France	**Dewi Sukarno** ... has lived in different countries including Switzerland, France and the United States
X bat Kurt Angle	**Anderson** defeated Kurt Angle and Abyss to advance to the finals
X est une ville de Kirghizistan	**Il'chibay** is a village in the Issyk Kul Province of Kyrgyzstan
X envahit Lebanon	In June 1982 **Israel** invaded Lebanon
X quitte Londres	In the autumn of 1997 **Nasar** left London for Afghanistan operating initially as a lecturer
Australie gagne X	Australia went on to win the Third Fourth and Fifth **Tests** and retain the Ashes
X devance Vettel	**Button** attempted to overtake Vettel at the start of the race, yet was pressured towards the grass by Vettel

Table 5.2: Example questions answered using the clusters, with the answer entity highlighted in bold. All are correct apart from the final entry, which would be correct without the non-factive verb *attempt*.

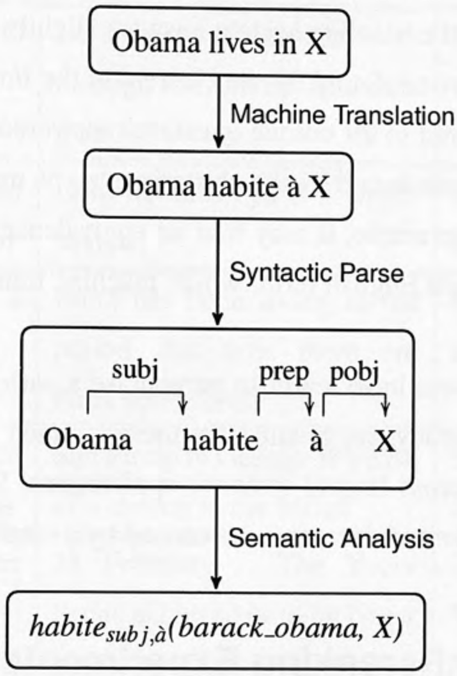


Figure 5.2: Pipeline used by baseline system for answering French questions. The pattern extracted from the translated sentence is used to search for answers in an English corpus.

English→ French	**Answers**	**Correct**
Baseline	269	86%
Clusters (best 270)	270	100%
Clusters (all)	1032	72%
French→ English	**Answers**	**Correct**
Baseline	274	85%
Clusters (all)	401	93%

Table 5.3: Results on wide-coverage Question Answering task. Best-N results are shown to illustrate the accuracy of the cluster-based system at the same rank as the baseline. It is not possible to give a recall figure, as the total number of correct answers in the corpus is unknown. *English→French* results are from the full French Wikipedia corpus, whereas *French→English* results are from a 10% sample.

true for the clustering approach—although the clustering system finds more answers with the English corpus, the baseline system answers slightly more unique questions (57 vs 66). The 1032 answers found by the clusters in the French corpus came from just 56 questions (compared to 29 unique questions answered by the baseline). This suggests that the translations found by the clustering can be more useful than those of Moses on this task—for example, it may find an equivalence between a rare French term and a common related English term, where machine translation may only find a more literal translation.

Despite this, the clusters have learnt to paraphrase a variety of relations between languages with high accuracy, suggesting that there is much potential for the use of unsupervised clusters in cross-lingual semantic applications. Some examples answers are given in Table 5.2. Most of the errors are caused by a small number of questions.

5.8 Translation Reranking Experiments

Ultimately, I would like to be able to translate using semantic parsing with cross-lingual clusters. However, the current representation is far too weak to support translation. As a step towards this goal, I investigated whether the clusters could be used to rerank the output of a machine translation system, on the basis of whether the semantic parse of the source sentence is consistent with that of candidate translations.

French sentences are sampled where the system can produce a semantic parse (i.e. it can extract a predicate between named entities that maps to a cross-lingual cluster). These sentences are translated to English using Moses, taking the 50-best list, and semantic parses are produced for each of these. If the semantic parse for the 1-best translation does not match the source semantic parse, the system searches for the parse from the 50-best list that most closely matches it—otherwise the sentence is discarded from the evaluation, as the cluster-based semantics agrees with the machine-translation.

To ensure that the evaluation focuses on the clusters, I excluded several other factors that might affect the results. The coverage of the CCG parsing and semantic analysis drops significantly on noisy translated sentences, and potentially acts as a language model by failing to produce any semantic parse on ungrammatical output sentences. Therefore only sentences which the system could produce a semantic parse for the 1-best machine translation output were considered. To avoid penalizing the machine-translation system for failing to translate named entities correctly, the system did not attempt to rerank sentences where the entities from the source sentence are not present

Source	Machine translation 1-best	Reranked translation
Le Princess Elizabeth arrive à Dunkerque le 3 août 1999	Le Princess Elizabeth is to manage to Dunkirk on 3 August 1999	The Princess Elizabeth arrives at Dunkirk on 3 August 1999
Esau Mwamwaya est un chanteur du Malawi	Esau Mwamwaya is a singer Malawi	Esau Mwamwaya is a singer from Malawi
Baltz vit maintenant à Paris et Venise	Baltz has been living in the period that now there are Paris and Vienna	Baltz now live in Paris and Venice
San Pietro in Gessate est une église de Milan	San Pietro in Gessate is a case of a church to the Milan	San Pietro in Gessate is a church in Milan
8 février : Le Yuder Pacha atteint le Niger	28 February : The Yuder Pacha achieved both by Niger	28 February : The Yuder Pacha reached Niger

Table 5.4: Example sentence that is reranked by the cluster-based reranking system. Human evaluators were asked which translation best preserved the meaning between the named entities.

in the 1-best translation.

Human annotators were shown the source sentence, the 1-best translation, and the translation chosen by the reranker (the translations were shown in a random order). To focus the evaluation on the semantic relations being modelled, annotators were asked which sentence best preserves the meaning between the named entities that have different relations in the semantic parse. This avoids the reranker being penalised for choosing a translation that is worse in aspects other than the relations it is modelling. An example is shown in Table 5.4. The data was annotated jointly by two fluent bilingual speakers, who reported high agreement on this task.

Results are shown in Table 5.5, with the original Moses output being preferred to the reranked translation in only 5% of cases where the model makes a positive prediction. The results also provide further evidence that the clustering has been able to accurately learn a range of semantic relations.

On inspecting the results, it was found that many of the cases where the annotators had no preference were caused by syntactic parse errors. For example, if the 1-best

	Percentage of translations preferred
1-best Moses translation	5%
Cluster-based Reranker	39%
No preference	56%

Table 5.5: Human preference judgements for the translation reranking experiment, based on a sample of 87 sentences. Results show the percentage of sentences for which the annotators preferred the original translation, the reranked translation, or neither. As discussed in the text, results where annotators had no preference were typically due to syntactic parse errors.

translation is correct, but a prepositional phrase is incorrectly attached by the parser, it will appear to have an incorrect semantics. A similar translation in the 50-best list may be correctly parsed, and consequently selected by the reranker. However, a human will have no preference between these translations. Incorporating K-Best parsing into the pipeline may help mitigate against such cases.

This preliminary experiment suggests that there is potential for future improvements in machine translation using cross-lingual distributional semantics. The system only attempts to rerank a very small proportion of sentences, but the coverage could be greatly improved by including relations between common nouns (rather than just named-entities)—future work should explore this.

5.9 Related Work

This chapter builds on Chapter 4, and other recent progress in monolingual distributional semantics [Poon and Domingos, 2009, Yao et al., 2011], by clustering typed predicates into those which are semantically equivalent. I have also shown how to bootstrap semantic information about entities from the Wikipedia markup, and I believe that this makes Wikipedia an interesting corpus for future work on monolingual distributional semantics. Other work on distributional semantics has represented named entities as strings, but linking them to a knowledge base reduces sparsity and should improve the quality of the clustering.

Cross-language Latent Relational Analysis [Duc et al., 2011] is perhaps the most similar previous work to this chapter, which moves the work of Turney [2005] into a multilingual setting. Duc et al. [2011] aim to compute, for example, that the 'latent relation' between *(Obama, US)* in an English corpus is similar to that between *(Cameron, UK)* in a foreign corpus. This is solved by finding all textual patterns between the two entity-pairs, and computing their overall similarity. Like us, they compute similarity between expressions in different languages based on named-entity arguments and clustering (unlike us, they also rely on machine translation for computing similarity). A key difference is that their system aims to understand the overall relation between an entity-pair based on many observations, whereas the approach developed here attempts to understand each sentence individually (as is required for tasks such as translation).

Various recent papers have explored the relationship between translation and monolingual paraphrases —for example Bannard and Callison-Burch [2005] create paraphrases by pivoting through a foreign translation, and Callison-Burch et al. [2006] show that including monolingual paraphrases improves the quality of translation by reducing sparsity. The success of these approaches depends on the many-to-many relationship between equivalent expressions in different languages. My approach aims to model this relationship explicitly by clustering all equivalent paraphrases in different languages.

Current state-of-the-art machine translation systems circumvent the problem of full semantic interpretation, by using phrase-based models learnt from large parallel corpora [Brown et al., 1993]. Although this approach has been very successful, it has significant limitations—for example, when translating between languages with very different word-orders [Birch et al., 2009], or with little parallel text.

Semantic machine translation aims to map the source language to an interlingual semantic representation, and then generate the target language sentence from this. Jones et al. [2012] show how this can be done on a small dataset using hyperedge replacement grammars. A major obstacle to this is designing a suitable meaning representation, which involves choosing a set of primitive concepts which are abstract enough to be capable of expressing meaning in any language [Dorr et al., 2004]. A recent proposal for this is the Abstract Meaning Representation [Banarescu et al., 2013], which uses English verbs as a set of predicates. This is a less abstract form of semantic interpretation than the clustering approach, as semantically equivalent paraphrases may be given a different representation. Such an approach also relies on annotating large amounts of text with the semantic representation—whereas the clustering approach offers a way to build such an interlingua using only a method for extracting predicates from sentences.

Whilst almost all recent work on machine-translation has relied on parallel text, there have been several interesting approaches that do not. Rapp [1999] learn to translate words based on small seed bilingual dictionary. Klementiev et al. [2012a] exploit a variety of interesting indirect sources of information to learn a lexicon—for example assuming that equivalent Wikipedia articles in different languages will use semantically similar words. The Polylingual Topic Model [Mimno et al., 2009] makes use of similar intuitions. Whilst the present work exploits equivalent Wikipedia articles for entity linking, it does not require aligned articles. Incorporating such techniques into the model would be a natural next step, allowing it to learn a more complete lexicon. To my knowledge, this chapter introduces the first approach to learn to translate semantic relations, rather than words and phrases.

Several other recent papers have learnt cross-lingual word clusters, and used these to improve cross-lingual tasks such as document-classification [Klementiev et al., 2012b], parsing [Täckström et al., 2012] and semantic role labelling [Kozhevnikov and Titov, 2013] in resource-poor languages. Cross-lingual word clusters are learnt by aligning monolingual clusters on the basis of parallel text—in language-pairs where parallel text is available, this offers an interesting complement to the proposed method of clustering based on named entities.

Mikolov et al. [2013b] recently introduced an unsupervised method for translating words. A recurrent neural network language model is trained on each individual language, which learns vector space embeddings for each word. Then, they assume that a mapping can be learnt from the vector space for one language to that of another. In contrast to my method, learning this mapping requires supervision (Google Translate is used). They also do not attempt to model ambiguity, and note that this problem significantly worsens results in some languages. However, if such techniques generalise well then they solve an important problem by improving the lexicons of machine translation systems using unlabelled text.

5.10 Future Work

An obvious extension is to try to cluster more languages, particularly more diverse ones. As discussed in Section 5.2, the major advantages of interlingua-based translation are to handle languages with diverse word orders, and those with little parallel text. This chapter only explored clustering English and French, which are closely related languages with a similar word order, and large amounts of parallel text available. It would be interesting to explore whether similar results could be obtained with languages such as Chinese. This would also provide a more thorough test of the hypothesis that meanings in any languages can be mapped to the same set of language-independent predicates.

Chapter 6 shows how a small amount of supervision can be used to greatly improve the quality of a monolingual clustering. It would be interesting to explore whether parallel text can provide this supervision in the multilingual case. Given two aligned sentences in different languages, we know that their underlying semantics is the same, which provides a large amount of information that can be used to guide the clustering. This process can be used as a complement to the named-entity based technique, which allows the system to exploit large amounts of non-parallel text. Chapter 6 introduces a semi-supervised technique for learning entailment graph structures over monolingual predicates, and parallel text would be a natural way of providing the necessary supervision in the cross-lingual case.

One of the biggest limitations of the current technique is that can only be applied to entities in Freebase, as Freebase is relied on for typing the entities. As discussed in Section 5.5.2, the LDA typing model of Chapter 4 was found to produce language-specific topics. It may be possible to avoid this limitation by building a variant of LDA with language-specific topic-document distributions drawn from the same Dirichlet with sparse priors. This constraint would encourage the model to have similar topic-document distributions in each language, to avoid creating language-specific topics.

5.11 Conclusions

In this chapter, I have shown that it is reasonably straightforward to extend the work of Chapter 4 to a multilingual setting, by exploiting the fact that equivalent predicates in different languages may have similar named entity arguments. The technique required had to be adapted by linking entities to an existing knowledge base, and using the knowledge base type schema. The best clustering was obtained by clustering predicates in each language independently and then aligning the clusters. Results show that clusters can be built with high precision for a variety of relations. Tentatively, I suggest that this is evidence that a set of interlingual semantic relations could be learnt for expressing the semantics of multiple languages. I believe this work opens a new and exciting direction, and I have suggested several interesting avenues for future work.

related languages with a similar word order, as a large amounts of parallel text is available. It would be interesting to explore whether similar results could be obtained with languages such as Chinese. This would also help to answer the question of the hypothesis that semantics of any languages can be mapped to the same set of language-independent predicates.

Chapter 6 shows how a small amount of bitext training can be used to greatly improve the quality of monolingual clustering. It would be interesting to explore whether parallel text can provide a good signal in the relation clustering. Given two aligned sentences in different languages, we know that their underlying semantics is the same, which provides a large amount of information that can be used to guide the clustering. This approach may be used as a complement to the unsupervised entity based technique, which allows the system to exploit large amounts of non-parallel text. Chapter 6 introduces a semi-supervised technique for learning equivalent graph structures over monolingual predicates, and parallel text would be a natural way of providing the necessary supervision to the semi-supervised approach.

One of the biggest limitations of the current technique is that it can only be applied to entities in Freebase, as knowledge is not available for many other entities. As discussed in Section 5.4.2, the LDA typing model of Chapter 4 was found to produce language-specific topics. It may be possible to avoid this limitation by building on [illegible] et al. with language-specific topic-document distributions drawn from the same Dirichlet with sparse priors. This would encourage the model to learn similar topic-document distributions in each language, to avoid creating language-specific topics.

5.11 Conclusions

In this chapter I have shown that it is relatively straightforward to extend the work of Chapter 4 to a multilingual setting, by exploiting the fact that equivalent predicates in different languages may have similar named entity arguments. The labelled data required had to be adapted by linking entities to an existing knowledge base, and using the knowledge base type scheme. The best clustering was obtained by clustering predicates in each language independently and then aligning the clusters. Results show that clusters can be built with high precision for a variety of relations. Tentatively, I suggest that this is evidence that a set of interlingual semantic relations could be learnt for expressing the semantics of multiple languages. I believe this work opens up a new and exciting direction, and I have suggested several interesting avenues for future work.

CHAPTER 6

Directional Inference for Combined Distributional and Logical Semantics

6.1 Introduction

This chapter extends both the formal and lexical semantics of the system described in Chapter 4. As a motivating example, consider a question answering system attempting to answer *Did Columbus sail to India?* from the sentence *Columbus failed to reach India*. To correctly answer *no*, the system must both understand that *sails to→reaches*, and that *fail* negates its complement. Conversely, the system should be able to answer *yes* to *Did Columbus try to reach India?*

Existing work struggles to model such complex interactions between the lexical and compositional semantics. Approaches based on non-compositional inference rules [Lin and Pantel, 2001, Berant et al., 2011] suffer from sparsity when dealing with complex expressions like *try to reach*. In contrast, standard formal-semantic approaches [Bos, 2008, Bobrow et al., 2007] cannot handle the relation between *sails to* and *reaches*, while modelling negation and monotone inference is problematic in vector-space models [Hermann et al., 2013]. The example questions also expose limitations of the model proposed in Chapter 4 (as discussed below), but I show how to extend it by incorporating ideas from both the distributional semantics and linguistics literatures.

The model developed in Chapter 4 use a flat clustering to model the meaning of content words. The flat clustering enables the system to model synonymy relations between words, but not relations where the entailment only holds in one direction—for example, *sails to*→*reaches*, but not vice-versa. I address this problem using the *entailment graph* framework introduced by Berant et al. [2011], which learns an ontology of entailment relations. I build entailment graphs over binarized predicates extracted from CCG parses, and show how to convert the graph into a CCG lexicon. I also show how the performance of these graphs can be improved with novel and linguistically motivated morpho-syntactic features.

Another limitation of the model from Chapter 4 is that it has a weak model of implicative verbs, such as *try* or *fail*. Such verbs are common, but semantically complex—for example, *X tried to reach Y* entails neither *X reached Y* nor *X did not reach Y*, but does entail *X attempted to reach Y*. I extend the existing work by including a lexicon of implicative verbs—using modal logic operators to help capture their semantics.

Both these developments are integrated into the system from Chapter 4, and I show that they lead to substantial improvements on an entailment task over the original model and a range of existing approaches.

6.2 Global Learning of Entailment Graphs

To solve the problem of directional lexical inference, I use the *global entailment graph* framework developed in [Berant et al., 2010, 2011, 2012].

Previous work had shown how to estimate a probability that one predicate entails another; for example that *conquers*→*invades*. I call a function that estimates this likelihood a *local classifier*. Berant et al. [2010] shows how a local classifier can be used to construct a *global entailment graph*, that contains an edge between all predicates that are predicted to entail each other. An example is shown in Figure 6.1. The graph is learned by maximizing the product of the probabilities of all the pair-wise edge decisions, based on the local classifier. The key observation is that because entailment is a transitive relation, the graph structure must be restricted to be closed under transitivity—which gives two important advantages to using the graph over simply using the output of the local classifier:

- It can reduce sparsity, by predicting edges for which there is no direct evidence. If the classifier gives high probabilities to *conquer* $\rightarrow$ *invade* and *invade* $\rightarrow$

Local Classifier Probabilities

$$p(conquer_{arg0,arg1} \rightarrow invade_{arg0,arg1}) = 0.9$$
$$p(invade_{arg0,arg1} \rightarrow attack_{arg0,arg1}) = 0.8$$
$$p(conquer_{arg0,arg1} \rightarrow attack_{arg0,arg1}) = 0.4$$
$$\ldots$$

Optimal Global Entailment Graph

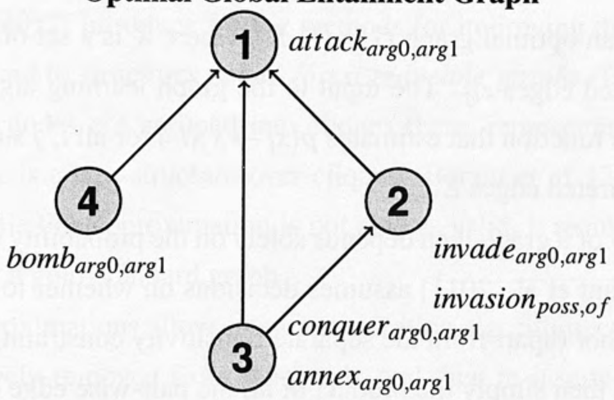


Figure 6.1: An entailment graph for relations between countries, built from the local classifier. Cliques have been collapsed into nodes representing synonyms, and edges represent entailments. The transitivity constraint means the global graph predicts $conquer_{arg0,arg1} \rightarrow attack_{arg0,arg1}$ (unlike the local classifier). Section 6.3.3 shows how to build a CCG lexicon from such graphs.

attack, this is evidence that *conquer* $\rightarrow$ *attack*, even if there is weak direct evidence for this.

- A local classifier can be inconsistent, as it only makes pair-wise decisions. For example, a system based on a local classifier may claim *conquer* $\rightarrow$ *invade* and *invade* $\rightarrow$ *attack* with high probability, but *conquer* $\rightarrow$ *attack* with low probability—which may affect performance in downstream tasks.

Transitivity can be unsound when predicates are ambiguous. Berant et al. [2011] add types to predicates, assuming typed predicates are unambiguous, and then construct separate graphs for predicates with the same types.

6.2.1 Learning Entailment Graphs

The aim is to find an optimal graph $G = (X, E)$, where X is a set of predicates x_i and E is a set of directed edges e_{ij}. The input to the graph learning algorithm is a set of predicates X, and a function that estimates $p(x_i \rightarrow x_j|F)$ for all i, j such that $i \neq j$. The output is a set of direted edges E.

The probability of a graph then depends solely on the probability of the set of edges in the graph. [Berant et al., 2011] assumes decisions on whether to include are independent of each other (apart from the separate transitivity constraint). The probability of a given graph is then simply the product of all the pair-wise edge decisions:

$$p(G) = \prod_{e_{ij} \in E} p(x_i \rightarrow x_j) \prod_{e_{ij} \notin E} (1 - p(x_i \rightarrow x_j))$$

To find the most probable graph:

$$\hat{G} = \arg\max_G \prod_{e_{ij} \in E} p(x_i \rightarrow x_j) \prod_{e_{ij} \notin E} (1 - p(x_i \rightarrow x_j))$$

Equivalently optimizing for log probability gives:

$$\hat{G} = \arg\max_G \sum_{e_{ij} \in E} logp(x_i \rightarrow x_j) + \sum_{e_{ij} \notin E} (1 - p(x_i \rightarrow x_j))$$

Introducing an indicator function I on whether an edge is in the graph:

$$\hat{G} = \arg\max_G \sum_{i \neq j} [I_{e_{ij} \in E} logp(x_i \rightarrow x_j) + (1 - I_{e_{ij} \in E}) log(1 - p(x_i \rightarrow x_j))]$$

$$\hat{G} = \arg\max_G \sum_{i \neq j} [I_{e_{ij} \in E} logp(x_i \rightarrow x_j) + log(1 - p(x_i \rightarrow x_j)) - I_{e_{ij} \in E} log(1 - p(x_i \rightarrow x_j))]$$

Dropping the term $log(1 - p(x_i \rightarrow x_j))$, which is independent of the graph:

$$\hat{G} = \arg\max_G \sum_{i \neq j} [I_{e_{ij} \in E} logp(x_i \rightarrow x_j) - I_{e_{ij} \in E} log(1 - p(x_i \rightarrow x_j))]$$

$$\hat{G} = \arg\max_G \sum_{i \neq j} log \frac{p(x_i \rightarrow x_j)}{1 - p(x_i \rightarrow x_j)} I_{e_{ij} \in E}$$

The graph is restricted to be closed under transitivity, so additional constraints are added:
$\forall i \forall j \forall k[(e_{ij} \in \mathbf{e} \wedge e_{jk} \in \mathbf{e}) \rightarrow e_{ik} \in \mathbf{e}]$

The transitivity constraint means that the problem is constrained optimization. As the objective is a linear function with binary variables, it can be solved optimally using Integer Linear Programming (ILP) solvers. The number of transitivity constraints is cubic in the number of predicates.

ILP is an NP-hard problem, and does not scale to large numbers of predicates. Berant et al. [2011] propose improving the efficiency by first decomposing the graphs into subgraphs G_n such that for $\forall x_i \in G_m, \forall x_j \in G_m$ $p(x_i \rightarrow x_j) < 0.5$. The intuition here is that the classifier estimates that there are no likely edges between two sets of predicates, then the optimal solution can never contain an edge between these sets of predicates, and they can effectively be treated as separate optimization problems.

Berant et al. [2012] introduce further methods for improving the efficiency. First, graphs are restricted to structures called *forest reducible graphs* (FRGs). An FRG is a graph in which nodes are grouped into cliques (here, representing clusters of synonyms), and there is a tree-structure over cliques. Berant et al. [2012] show experimentally that, while this approximation is not always valid, it requires removing only 5% of edges from a gold-standard graph.

The FRG approximations allow a new optimization algorithm, called *tree-node-fix*. Nodes are iteratively removed from the graph, and then re-inserted at the point that most improves the graph's objective function. This process is repeated to convergence. Bounding the maximum number of iterations means that the algorithm is quadratic in the number of edges in the graph. The graph is initialized using ILP, multiplying the local classifier probability by a prior low enough that an optimal solution can be found quickly. This initialization aims to avoid local minima, by building an optimal sparser graph that captures the main structure.

6.3 Entailment Graphs for CCG

The present section shows entailment graphs of typed predicates can be built and converted into a CCG lexicon. This lexicon allows the CCG system to make directional lexical inferences, which are not possible with Chapter 4's flat clustering.

In Section 6.3.1, I describe how entailing/non-entailing pairs of predicates are extracted from a small annotated entailment dataset, and are used for training the local classifier.

Section 6.3.2 then defines the features used for training the local classifier, which include distributional statistics (extracted from a large unlabelled corpus), and novel morpho-syntactic features.

The features and training data are used to train a local classifier, which is used to construct entailment graphs with Berant et al. [2012]'s method. The training method is semi-supervised, as it uses both a small annotated dataset and a large unlabelled corpus.

Section 6.3.3 shows entailment graphs can be converted into a CCG lexicon, which can then easily be incorporated into the CCG inference system.

6.3.1 Training Data

The local classifier is trained using a small amount of labelled data (as well as distributional statistics). Training data is automatically extracted from an entailment dataset, which contains pairs of sentences annotated with whether the first entails the second. First, the system makes a CCG semantic parse of each sentence, and then determines if changing a single binary predicate in the premise and hypothesis logical-forms is sufficient to make an inference hold. For example, if the entailment problem is *Rome conquered Carthage→Rome invaded Carthage* then the inference only holds if $conquer_{arg0:T1,arg1:T2} \rightarrow invade_{arg0:T1,arg1:T2}$ (where T1 and T2 are the types of the predicate). Training instances are created for all type combinations *(T1,T2)*, and the instances are weighted by the probability of that type combination according to the type model (filtering instances with weight less than 10^{-3}).

If the predicates are negated, the direction of the training instances are reversed, because $(\neg p \rightarrow \neg q) \iff (q \rightarrow p)$. For example, in *Britain didn't invade Rome→ Britain didn't conquer Rome*, the following training instance is added: $conquer_{arg0:T1,arg1:T2} \rightarrow invade_{arg0:T1,arg1:T2}$.

6.3.2 Features

The training data is used to train a *local classifier*, which determines the probability of one predicate entailing another. In Chapter 4, predicate similarity is determined based on a single statistic—cosine similarity of vectors of expected counts of named-entity argument-pairs. This statistic is often sparse, is symmetric (so does not capture similarity that only holds in one direction) and ignores other potentially useful sources of information. In contrast, the present method can use multiple features to determine similarity, because it uses a supervised classifier.

The classifier is trained using a variety of features, many of which are novel. The most important features are still distributional similarity statistics—but I also experiment with adding ontological and morpoh-syntactic features.

Distributional Features Following Berant et al. [2010], I incorporate a range of distributional features, which are based on the expected argument counts of the predicates in a large corpus. I use the Weeds precision [Weeds and Weir, 2003] similarity measure which is asymmetric (unlike the commonly used cosine similarity), allowing the system to capture entailments that hold in only one direction. The similarity of two predicates P_1 and P_2, with arguments x occurring with frequency $f_i(x)$, is calculated as:

$sim(P_1,P_2) = \frac{\sum_x min(f_1(x),f_2(x))}{\sum_x f_1(x)}$

This metric is calculated for the argument-pairs of the typed predicate, the average of individual arguments of the typed-predicate, and argument-pairs of the corresponding untyped-predicate. This limits the potential for errors caused by the type-model to make equivalent predicates appear to have different distributions. I also add a feature for the cosine similarity of the word embeddings released by Mikolov et al. [2013a] (it is not clear how to represent directional similarity in a vector space).

I take the novel step of discretizing all of these real-valued features, splitting them into 10 bins, choosing thresholds that make the bins of as equal width as possible. One bin is reserved for unseen predicates, to distinguish them from predicates which have been seen but are have 0-similarity. Binning features allows their importance to scale non-linearly in their value, and avoids problems with feature scaling.

Morpho-syntactic Features The morphology and syntax of predicates can provide reliable clues about their entailments, but have seen surprisingly little attention. For example, the system could reasonably infer *Google owns YouTube→Google is the owner of YouTube*, even if it has never seen the words *own* or *owner* before. I add a variety of novel features to capture such inferences.

- *Add prefix.* If two predicates are identical, except for one having an additional prefix (of length < 4), this feature is the value of the prefix. This feature learns, for example, that adding the prefix *re* to a word makes an entailment hold in one direction (as in *rewrite→write*).

- *Diathesis alternation* A diathesis alternation occurs where a predicate has multiple subcategorization frames, which is reflected in different argument keys expressing the same semantic relation. Dativization is one example, e.g. in *I gave the present to her↔I gave her the present.* It may learn, for example: $verb_{arg1,to} = verb_{arg2,arg1}$

- *Swap argument key* This feature applies if two predicates are identical except for a single change of argument key. For example, the argument keys provided by the prepositions *at* and *in* are often equivalent.

- *Change Suffix* If two predicates start with the same string, and have the same argument keys, this returns the difference in the suffixes. This learns, for example, that *re* and *er* endings are sometimes interchangeable (e.g. British English *centre* and US English *center*).

- *Change Suffix and Argument Keys* This feature is the same as the *Change Suffix* feature, but also allows the argument keys to change. This feature can learn to map verbs onto deverbal forms, for example the pattern: $verb_{arg0,arg1} = verb\text{–}er_{be,of}$ (as in *X owns Y* and *X is the owner of Y*).

These features prove useful, but currently the technique is limited by their sparsity in labelled data. Semi-supervised techniques such as co-training may be useful in generalizing them.

WordNet Features The lexicalized framework makes it simple to add information from existing word ontologies, such as WordNet [Miller, 1995]. Previous work has found WordNet of limited use for semantic inference [Bos, 2008, Bobrow et al., 2007]. Challenges include limited coverage (for example, no WordNet relation holds between *write* and the noun *author*), and fine-grained sense distinctions which even humans find difficult [Hovy et al., 2006].

The system uses WordNet information as features in the classifier, meaning that it can incorporate information from hand-built ontologies, without being limited by their weaknesses. I add binary features if any of the following WordNet relations holds between any sense of the predicates: *synonym*, *hyponym*, *hypernym*, and *antonym*. The relations *causes* and *entails* were found to be too rare to be useful.

$$\dfrac{\dfrac{\text{Rome}}{\substack{NP\\ rome'}} \quad \dfrac{\dfrac{\text{invaded}}{\substack{(S\backslash NP)/NP\\ \lambda y\lambda x.r1(x,y)\wedge r2(x,y)}} \quad \dfrac{\text{Carthage}}{\substack{NP\\ carthage'(y)}}}{\substack{S\backslash NP\\ \lambda x.r1(x,carthage')\wedge r2(x,carthage')}}>}{\substack{S\\ r1(rome',carthage')\wedge r2(rome',carthage')}}<$$

Figure 6.2: A CCG derivation using the lexicon derived from entailment graphs.

6.3.3 Lexical Entries

The training data and features from Sections 6.3.1 and 6.3.2 (basing distributional features on a large unlabelled corpus) are used to train a local classifier, which is used to build entailment graphs. In this section, I show how to use the entailment graph in a CCG system, allowing it to be combined with the benefits of formal semantics.

The simplest approach would be to create a logical axiom for each pair of predicates in an entailment relation, analogously to how Bos and Markert [2005] compile WordNet into logical axioms. This method would be inefficient, as the number of logical axioms required grows quadratically in the size of the vocabulary. It is also at odds with CCG's lexicalized philosophy, as it would require the meaning of a word to be stored in an ontology, rather than in its lexical entry.

I therefore propose an alternative, lexicalized approach that requires no additional logical axioms. Each node in the entailment graph is assigned a unique arbitrary relation identifier, representing a unique concept. Then, the interpretation of a predicate becomes the conjunction of all reachable relation identifiers in the graph. For example, for the graph in Figure 6.1, the system creates entries such as:

attack $\vdash$ (S\NP)/NP : $\lambda x\lambda y.r1(y,x)$
invade $\vdash$ (S\NP)/NP : $\lambda x\lambda y.r1(y,x)\wedge r2(y,x)$
conquer $\vdash$ (S\NP)/NP : $\lambda x\lambda y.r1(y,x)\wedge r2(y,x)\wedge r3(y,x)$
bomb $\vdash$ (S\NP)/NP : $\lambda x\lambda y.r1(y,x)\wedge r4(y,x)$

It is easy to verify that this lexicon allows inferences such as *conquers→invades* (but not the reverse) and *didn't invade→didn't conquer*.

An example derivation using this lexicon is shown in Figure 6.2.

Verb	Interpretation
try	$\lambda p\lambda x\lambda e.try(e) \wedge agent(x,e) \wedge \diamond\exists e'[p(x,e') \wedge theme(e,e')]$
fail	$\lambda p\lambda x\lambda e.try(e) \wedge agent(x,e) \wedge \diamond\exists e'[p(x,e') \wedge theme(e,e')] \wedge \neg\exists e''[p(x,e'')]$

Table 6.1: Example entries from the lexicon of implicative verbs, with category: $(S_{dcl}\backslash NP)/(S_{to}\backslash NP)$

6.4 Implicative Verbs

Another weakness of the approach of Chapter 4 is that it has a poor model of implicative verbs. Many verbs that take predicative complements do not assert the truth of that complement. For example *Google wants to buy YouTube* does not entail *Google buys YouTube*—however, the Chapter 4 model would licence the inference, as the nested proposition will be present in the logical form.

Lexicons of implicative and factive predicates have previously been used by MacCartney and Manning [2008], Bos [2013] and Lotan et al. [2013]. Following this work, I improve the system by adding a lexicon of implicative verbs. This lexicon requires extending the logic with the modal-logic operator $\diamond$ (propositions scoped by a $\diamond$ operator are hypothetical).

Given the complex semantics of such verbs [Karttunen, 1971], and the relatively small number of common examples, I chose to hand-code the semantics for a small ontology of a number of common examples. Whether the semantics of implicative verbs can be learnt from distributional statistics is an open question. Some examples are shown Table 6.1, which allow the system to capture inferences such as *Google failed to buy Microsoft*→*Google didn't manage to buy Microsoft*. These lexical entries are more detailed than those used by previous work—which only mark whether the nested proposition is entailed or not, so do not capture relations such as those between *try*, *fail*, and *manage*. I hand-code semantics for 22 verbs, split into classes of *wanting*, *trying*, *failing*, *managing*, *needing*, *avoiding* and *expecting*.

Other auxiliary verbs are treated as being semantically transparent. This is necessary to allow inferences such as *Obama lives in Washington*→*Obama continues to live in Washington*—where any non-trvial semantics for *continues to* would prevent the inference from holding. I leave a thorough treatment of temporal semantics to future work—a more detailed proposal is sketched in Section 6.7.2.

While the current ontology is clearly very limited in size, I show empirically that it does help on an entailment task, which motivates a more detailed treatment in the future.

To allow inference with a standard theorem prover, modal operators are removed using a *possible worlds* semantics [Kripke, 1963]. To do this, the system adds an extra 'possible world' argument to each predicate in the logical form. At the top level, this is instantiated by an *actual-world* constant. $\diamond$ operators can be removed by existentially quantifying a new possible-world variable, and using this as an argument to nested terms. For example, it is simple to convert the semantics for *john might sleep* from $\diamond sleep(john)$ to $\exists w[sleep(john, w)]$.

The current approach is unable to deal with negated factive verbs. For example, both *John knew Google bought YouTube* and *John didn't know Google bought YouTube* imply that *Google bought Youtube*—because the inference relies on presupposition rather than entailment. However in the latter, our current approach has no way to mark that the *Google bought YouTube* is not within the scope of negation, and would instead build a logical form that does not entail *Google bought Youtube*, such as:
$\neg\exists e[know(e) \wedge agent(john, e) \wedge \exists e'[buy(e') \wedge arg0(google, e') \wedge arg0(youtube, e')]]$

One way to deal with this in the compositional framework would be to build a separate semantics for presuppositions in parallel with the main semantics during the derivation, analogously to Clausen and Manning [2009]. Another alternative, along the lines of Chapter 3, would be to represent events with Skolem terms, which could be given positive polarity to move them outside the scope of negation. Then lexical entries such as the following could be used, where $+E_p$ refers to a non-negated event satisfying predicate p:
know $\vdash (S_{dcl}\backslash NP)/S : \lambda p \lambda x \lambda e.know(e) \wedge agent(x, e) \wedge arg(+E_p, e)$

6.5 Entailment with Combined Distributional and Logical Semantics

6.5.1 Inference

The logical forms from the system can be used to recognize textual entailment, by performing logical inference with theorem provers. The output of the CCG derivation is a distribution over logical forms, as explained in Chapter 4. The probability of the inference is then the sum of the probabilities of the logical forms for which the entailment holds, allowing the system to marginalize out the ambiguity.

As explain in Chapter 4, the probability of a given logical form for a syntactic parse is conditioned solely on the types of the nouns (which determine the types of the corresponding predicates). For example, there may be a high-probability logical form for *Obama was born in Hawaii* in which *Obama* has a *person* type and *Hawaii* has a *location* type, but there will be some probability mass reserved for other types (such as *Hawaii* being a *date*. I make the assumption that all entities referred to by the same word in the premise and hypothesis have the same type—to do this, I merge their separate type distributions by taking the product and renormalizing. This is similar to the one-sense-per-collocation assumption that has been used in word-sense disambiguation [Yarowsky, 1993], and significantly reduces the search space. I also prune logical forms whose probability is less than 10^{-3}. A more efficient alternative would be to directly use a probabilistic logic, such as Markov logic networks [Richardson and Domingos, 2006].

6.5.2 Missing Predicates

The test data may contain predicates which are too rare in the unlabelled corpus to be included in the entailment graphs[1]. For example, the verb *vanquish* may not satisfy the frequency cutoffs for the graph on relations between countries. As building the graphs is computationally expensive, we cannot include entries for every possible predicate.

For inference, these predicates are temporarily inserted into the graph—for example, *vanquish* should be added to the *conquer* cluster in the graph in Figure 6.1. The local classifier is used to estimate the probability that the new predicate implies each of the other predicates in the graph, and then they are inserted at the point that maximizes the probability of the new graph (according to the probability from Section 6.2. The insertion is restricted so that only edges connecting to the new predicate are modified, so that inferences between predicates already in the graph are unaffected. This restriction is achieved by either inserting the new predicate into an existing synonym cluster, or into a new singleton synonym cluster—which can either be a root, a leaf, or between two already-connected clusters. After performing inference, the graph is restored to its original state, so the graphs cannot grow to an unbounded size.

[1]Previous work on entailment graphs avoids this problem by evaluating on a prespecified list of predicates.

Premise	Hypothesis	Answer
Obama want to boost the defense budget	Obama increase the defense budget	False
The thieves make off with TVs	The thieves manage to steal TVs	True
My son be terrified of him	My son have a fear of him	True

Table 6.2: Examples from the Zeichner et al. [2012] entailment dataset.

6.6 Experiments

6.6.1 Dataset

I perform the evaluation on the entailment dataset produced by Zeichner et al. [2012]. This contains 5556 entailment problems (after excluding those annotated as nonsensical), based on pairs of Reverb extractions from the ClueWeb corpus[2]. Some examples are given in Table 6.2. I chose this dataset as the inferences rely purely on lexical semantics, so it targets the traditional weakness of formal semantics approaches. Chapter 3 has already shown the CCG-approach offers a strong model of logical and compositional semantics, and the work in this chapter addresses lexical semantics. Other entailment datasets, such as RTE [Giampiccolo et al., 2007], involve many forms of inference that are not the current focus, such as coreference resolution and encyclopedic knowledge. I held out a random 10% for testing, and a 10% development set was used.

There are several reasons for preferring an entailment-based evaluation to the question-answering evaluations used in previous chapters and other work.

- There are no gold-standard answers for question answering, as the number of correct answers in the corpus is unknown. Therefore the output from every system has to be evaluated manually, which is time-consuming. A major disadvantage of manual evaluation is that it makes it much harder to develop models, as there is no development set. The model described in Chapter 4 has a number of parameters and design decisions, and these are difficult to tune accurately without an automatic evaluation. In contrast, entailment evaluations have gold-standard annotations.

[2]http://lemurproject.org/clueweb09/

- Automatically generated questions are naturally skewed towards frequent predicates, which may mask weaker performance on rare predicates. Frequent predicates are easier to cluster because they have less sparse distributional statistics.
- Performance on question-answering evaluations could also easily be improved by incorporating other NLP techniques, such as using co-reference resolution to find additional answers. However, this masks the performance of the distributional component of the model, which is the main focus of the evaluation. The simple sentence construction in the Zeichner corpus means that results primarily demonstrate the quality of the lexical semantics.

For these reasons, I chose to only evaluate the new models on the entailment dataset, and suggest that similar datasets should be used in future work.

6.6.2 Experimental Setup

6.6.2.1 Training Corpus

In order to use a large training corpus, I used the recently-released Google Syntactic N-grams [Goldberg and Orwant, 2013]. The corpus contains the frequency of small fragments dependency trees from a parsed version of the Google Books corpus, containing 345 billion words (roughly 2 orders of magnitude larger than the Gigaword corpus used in Chapter 3).

Dependency parses are of course a different representation from the predicates produced by the system in Chapter 4, so I defined a simple mapping for converting common constructions to the format that would have been produced by the CCG system. While there is not a 1-to-1 mapping between the predicates produced by the system in Chapter 4 and the dependency tree fragments, and the dependency parses fail to abstract over constructions such as relative clauses, the large size of the corpus amply compensates. Better training data could be extracted by CCG-parsing the original corpus, but would not be practical on academic resources.

The following mapping was used:

- **Active-voice verbs**: The verb is used as the predicate, and argument keys are mapped as follows: *nsubj→arg0*, *dobj→arg1* and *iobj→arg2*. Prepositions add an argument key of the same name—to reduce the number of predicates, we filter those where both arguments are supplied by prepositions (so from *Obama was born in Hawaii in 1961*, we do not extract a binary relation between *1961* and

Phrase	Dependencies	Predicate
X is president of Y	$X \xleftarrow{nsubj} is \xrightarrow{dobj} president \xrightarrow{pobj\_of} Y$	$president_{be,of}$
X is taller than Y	$X \xrightarrow{nsubj} taller \xrightarrow{pobj\_than} Y$	$taller_{be,than}$
X bought Y	$X \xrightarrow{nsubj} bought \xrightarrow{dobj} Y$	$buy_{arg0,arg1}$
X was bought by Y	$X \xrightarrow{nsubjpass} bought \xrightarrow{pobj\_by} Y$	$buy_{arg0,arg1}$

Table 6.3: Example conversion between dependency parse fragments and the predicates used by the CCG system.

Hawaii). If the verb has a particle argument (identified by the *prt* dependency) then the particle name is appended to the verb name.

- **Passive-voice verbs** are treated as for active voice verbs, with the following exceptions: If the verb has a passive *nsubjpass* dependency, I use *nsubjpass→arg1*, *dobj→arg2* and *pobj_by→ arg1*.

- **Nouns**: If the noun is the *dobj* or *attr* of a copula verb, a *be* argument is added to the subject of the copula. Genitive *poss* dependencies add a *poss* argument key. Arguments supplied by prepositions are handled as for verbs.

- **Predicative Adjectives**: The adjective is used as the predicate, and an argument key is added for the subject: *nsubj→be*. Arguments supplied by prepositions are handled as for verbs.

Some examples are shown in Table 6.3, which may make the process clearer.

A type model was trained using the same methods and data as Chapter 3—but with 25 types instead of 15. A larger number of types was necessary here to ensure there were sufficiently fine-grained word senses for transitivity to hold. For building entailment graphs, I take the most frequent 100 predicates of each type (filtering those occurring less than 100 times).

Test sentences are parsed with the N-best version of the C&C parser [Ng and Curran, 2012], taking the 50-best parses to attempt to mitigate parser errors.

6.6.2.2 Building Entailment Graphs

Entailment graphs are built using the Tree Node Fix algorithm [Berant et al., 2012] with a prior of 0.5 (because the test examples are drawn from the same distribution as the training data). The graphs are initialized using Integer Linear Programming with a prior of p (initially 0.25, but backing off to $p-0.05$ if no solution is found in 60 seconds). Parameters were chosen based on development data. For ILP solving, I use LPSolve [Berkelaar et al., 2004]. Supervised classifiers use the Weka [Hall et al., 2009] implementation of logistic regression. A fragment of an entailment graph learned by the system is shown in Figure 6.3.

6.6.3 Comparison Systems

I compare with the following approaches:

- **Non-compositional**: Various papers have explored learning inference rules between Reverb patterns, based on their arguments [Berant et al., 2011]. I use the distributional features used by the CCG model, and train a logistic regression classifier. This approach is not compositional, which causes sparsity when dealing with expressions such as *try to sail to*, but means it has no extra difficulty with multiword expressions. For training, I used the publicly available corpus of the best 15-million Reverb extractions from ClueWeb.

- **CCG-Baseline**, the model from Chapter 3: a simple CCG semantic parser with no distributional clustering. Performance on this dataset is weak, as the premise and hypothesis are constructed to have different predicates—however, it can still make inferences when the predicates differ in function words, or are the same except for the removal of modifiers.

- **CCG-WordNet**, which extends the previous system with WordNet-derived inference rules.

- **CCG-ChineseWhispers**: Chapter 4's unsupervised model for CCG semantics with predicate clusters derived using Chinese Whispers [Biemann, 2006].

- **CCG-EntailmentGraphs**: CCG with lexical entries derived from entailment graphs.

- **Simple Compositional Semantics (SCS)** Several proposals have been put forward recently for computing the meaning of word combinations in vector spaces (see Baroni et al. [2013] for an overview). I experimented with both the *additive* and *multiplicative* models of Mitchell and Lapata [2008], which have been shown to perform competitively with more sophisticated alternatives [Blacoe and Lapata, 2012]. I use word vectors from Blacoe and Lapata [2012]. I represented each premise and its hypothesis in the dataset by their corresponding compositional vectors, and trained a logistic regression classifier that uses vector entries as features to predict entailment.

- **LATENTLC** A recent model from Abend et al. [2014], which was designed specifically for handling multi-word predicates, such as light verb constructions. Results are quoted from Abend et al. [2014], who uses a different test/train split.

The only other previous work I am aware of on this dataset is by Melamud et al. [2013]. This work only reports results of various subsets of the dataset, so it is difficult to make a direct comparison with their models. Their approach to typed-predicate similarity is related to that used here and in Chapter 4.

Unfortunately, the dataset only contains lemmatized sentences, which is problematic for syntactic and semantic models which rely on morphological information. I attempted to automatically un-lemmatize the corpus, by replacing each pattern with the most frequent phrase that lemmatizes to it, but this process is noisy (for example, *X is taught at Y* and *X is teaching at Y* lemmatize to the same string). This means that the syntactic parser performance on the dataset is weak (as it relies on morphological information), with a consequent effect on the semantics. I used an N-best parser, but this does not mitigate errors by the POS-tagger.

6.6.4 Results

Results are shown in Table 6.4. Results demonstrate that the entailment graph approach outperforms both the baseline CCG and the flat clustering used in Chapter 4 by a wide margin. Using the implicative verb lexicon also improves the results. The non-compositional system only improves slightly over the majority-class baseline, due to the sparsity of its patterns—using a larger corpus may offset this somewhat, but sparsity will always be problematic for non-compositional approaches.

It should be noted that this particular dataset targets the weaknesses of the CCG approach, rather than its strengths—the aim being to expose and address the limitations of computational models of formal semantics. The sentences are relatively simple syntactically (they are extracted by a finite-state model), so it contains few examples of the kinds of relations that require compositional semantics, such as conjunctions, relative clauses and long-range dependencies. Conversely, the dataset contains many

System	Accuracy
Majority Class	56.8%
SCS—Additive	60.6%
LATENTLC	64.6%
Non Compositional	57.4%
CCG Baseline	57.8%
CCG Baseline+WordNet	61.9%
CCG ChineseWhispers	58.0%
CCG Entailment Graphs	64.0%
CCG Entailment Graphs+ Implicative Verb Lexicon	**66.0%**

Table 6.4: Results on the entailment task.

examples of problems that the system is currently unable to model compositionally. A common example is light verb constructions, such as *take a shower*. The current CCGBank syntax gives the same analysis as the 'heavy' usage of *take*, as in *take a book*, whereas ideally the syntax would identify *shower* as being the main predicate. Improvements here should lead to better overall results on this task. Results in Chapter 3 show strong performance on a dataset that emphasises function words, and that result remains independent of the present approach to lexical entailment.

Inspecting the results, I found that the system in fact predicts a relatively small number of answers with high precision (29% recall at 80% precision), and has 0-confidence on others. In fact, it is not possible for the system to make a prediction on 40.1% of problems, with *any* clustering. Cases where the system is unable to make predictions include light verb constructions, and multi-word expressions. I test if an inference is possible by seeing if it holds when all binary predicates are replaced with the same symbol—if not, then no clustering can make the entailment hold. Future work should address improving the coverage.

Because I build deterministic ontologies, many inferences will have 0-probability, even if there is some distributional similarity between the predicates. On the other hand, high precision systems are likely to be useful for applications such as question-answering. This result also suggests that much higher accuracy numbers could be obtained by hybridising with high-recall methods, but I do not explore that here. Ablation

Feature Set	Accuracy
All	66.0%
Without Distributional	65.3%
Only Distributional	63.8%
Without Morpho-syntactic	66.0%
Without Wordnet	63.7%

Table 6.5: Accuracy using different feature sets (using the implicative verb lexicon).

results are given in Table 6.5. Although distributional features are helpful, they have surprisingly little impact. This result is at least partially an artifact of the dataset—which was constructed by choosing examples that already had high distributional similarity, thereby making distributional similarity artificially less effective. If the dataset had been constructed differently based on inferences that held in WordNet, then WordNet features would be found to have little impact. It is crucial to take the methodology used to construct entailment datasets when interpreting the results. However, despite this limitation, purely distributional features do achieve good results.

Results highlight the importance of incorporating WordNet into distributional models—future work should experiment with other lexical resources. The novel morpho-syntactic did not affect results, possibly due to the limited syntactic constructions found in the dataset.

The SCS—Additive model performs surprisingly well, given the simple bag-of-words approach to composition (I was unable to outperform the majority-class baseline with the multiplicative model). This is partly an artifact of the dataset—the premise and hypothesis sentences are identical, except for a small number of consecutive words, meaning that the difference between the premise and hypothesis vectors will be the difference between a small number of word vectors. Consequently the classifier can effectively treat the classification as a simple word-similarity problem, rather than a sentence inference problem. The dataset is much more a test of lexical semantics than compositional semantics (the work has focused on lexical semantics, as it has been the main weakness of logical approaches). Composition is straightforward with the logical CCG approach, so it is reasonable to expect the performance to be unaffected by longer sentences—but they are likely to be much harder for the SCS approach. For example, the SCS—Additive model has the same representation for *Herons eat frogs*

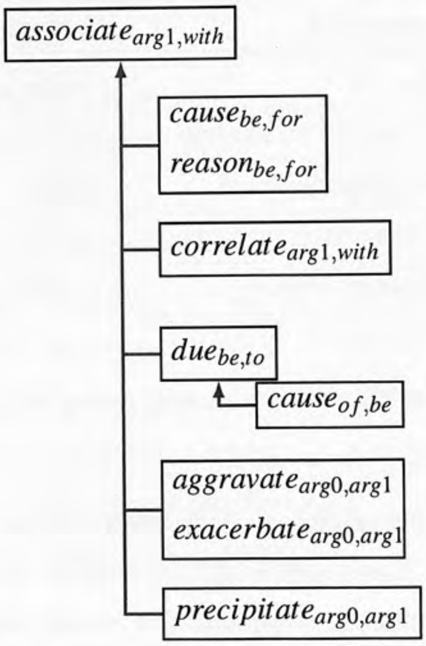


Figure 6.3: Example fragment of an entailment graph learnt by the model.

and *Frogs eat herons*, but this weakness is not exposed as the dataset does not require any knowledge of compositionality.

LATENTLC was designed specifically to handle construction such as MWEs and light verbs, which are not possible for my model. The model is a bag-of-words approach, which means it would need some modifications to scale to full length sentences. Despite the dataset being better suited to LATENTLC, my model achieves slightly higher performance. On the other hand, I make use of larger unlabelled corpora, and the WordNet ontology, so the comparison is not a fair one. Future work should investigate combining the strengths of both approaches.

6.7 Future Work

While the model described in this chapter gives a much more powerful model of semantics than that of Chapter 4, it is still very far from being a complete solution to computational semantics. However, I believe that the current framework could be extended in a number of ways to provide a quite general model of natural language inference. Here, I describe some of the major limitations of the current model, and propose how they could be overcome with future work. Some of these ideas have previously been published in Lewis and Steedman [2014a].

6.7.1 Entity Typing

The models introduced in this thesis have used an LDA topic model for entity typing. However, this approach is clearly sub-optimal. Firstly, it is parametric, meaning that the number of types must be pre-specified. This problem could be addressed using a Hierarchical Dirichlet Process model [Teh et al., 2006], a non-parametric generalisation. Another major drawback is that a 'flat' typing is used. For example, the model learns separate types for different kinds of people, such as politicians and footballers. A better model would build a hierarchy of types, in which politicians and footballers were sub-types of people. Existing topic models such as Pachinko Allocation [Li and McCallum, 2006] attempt to address these problems. I experimented with these models, but found that the Gibbs sampling was highly susceptible to poor solutions in which a bad topic was set at the root.

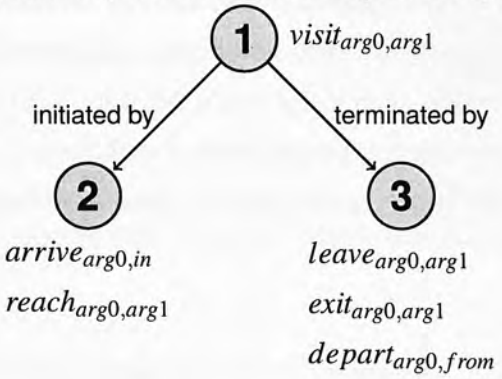
6.7.2 Temporal Semantics

One case where combining formal and distributional semantics may be particularly helpful is in giving a detailed model of temporal semantics. A rich understanding of time would allow the system to understand *when* events took place, or when states were true. Most existing work ignores tense, and would treat the expressions *used to be president* and *is president* either as equivalent or completely unrelated. Failing to model tense would lead to incorrect inferences when answering questions such as *Who is the president of the USA?*

Another motivation for considering a detailed model of temporal semantics is that understanding the time of events should improve the quality of the distributional clustering. It has recently been shown that such information is extremely useful for learning equivalences between predicates, by determining which sentences describe the same events using date-stamped text and simple tense heuristics Zhang and Weld. Such methods escape common problems with traditional approaches to distributional similarity, such as conflating causes with effects, and may prove very useful for building entailment graphs.

Temporal information is conveyed by both by auxiliary verbs such as *will* or *used to*, and in the semantics of content words. For example, the statement *John is visiting Baltimore* licences entailments such as *John has arrived in Baltimore* and *John will* *leave Baltimore*, which can only be understood through both knowledge of tense and lexical semantic relations.

The requisite information about lexical semantics could be represented by labelling edges in the entailment graphs, along the lines of Scaria et al. (2013). Instead of edges simply representing entailment, they should represent different kinds of lexical relations, such as *precondition* or *consequence*. Building such graphs requires training classifiers that predict fine-grained semantic relations between predicates, and defining transitivity properties of the relations (e.g. a precondition of a precondition is a precondition). For example, the system might learn the following graph:



By defining a simple mapping between edge labels and logical forms, this graph can be converted to CCG lexical entries such as:

visit $\vdash (S\backslash NP)/NP : \lambda y \lambda x \lambda e.rel1(x,y,e) \wedge \exists e'[rel2(x,y,e') \wedge before(e,e')] \wedge \exists e''[rel3(x,y,e'') \wedge after(e,e'')]$

arrive $\vdash (S\backslash NP)/PP_{in} : \lambda y \lambda x \lambda e.rel2(x,y,e)$

leave $\vdash (S\backslash NP)/NP : \lambda y \lambda x \lambda e.rel3(x,y,e)$

These lexical entries could be complemented with hand-built interpretations for a small set of common auxiliary verbs:

has $\vdash (S\backslash NP)/(S_b\backslash NP) : \lambda p \lambda x \lambda e.before(r,e) \wedge p(x,e)$

will $\vdash (S\backslash NP)/(S_b\backslash NP) : \lambda p \lambda x \lambda e.after(r,e) \wedge p(x,e)$

is $\vdash (S\backslash NP)/(S_{ng}\backslash NP) : \lambda p \lambda x \lambda e.during(r,e) \wedge p(x,e)$

used $\vdash (S\backslash NP)/(S_{to}\backslash NP) : \lambda p \lambda x \lambda e.before(r,e) \wedge p(x,e) \wedge \neg\exists e'[during(r) \wedge p(x,e')]$

Here, r is the reference time (e.g. the time that the news article was written). It is easy to verify that such a lexicon supports inferences such as *is visiting→will leave*, *has visited→has arrived in*, or *used to be president→is not president*.

The model described here only discusses tense, not aspect—so does not distinguish between *John arrived in Baltimore* and *John has arrived in Baltimore* (the latter says that the consequences of his arrival still hold—i.e. that he is still in Baltimore). Going further, we could implement the much more detailed proposal of Moens and Steedman (1988). Building this model would require distinguishing *states* from *events*—for example, the semantics of *arrive*, *visit* and *leave* could all be expressed in terms of the times that an *is in* state holds.

6.7.3 N-ary Relations

The models presented in this thesis only attempt to cluster binary relations (binarizing higher-order relations). Whilst this is a common approach, it is clearly a simplification, and better modelling of these relations should result in much improved performance. One problem is that binarizing means it is not possible to make inferences such as *Obama was born in Hawaii*→*Obama was born*, because the binary predicate in the premise does not entail the unary predicate in the hypothesis. It also cannot learn that the similarity of $buy_{arg0,arg1}$ and $purchase_{arg0,arg1}$ makes it more likely that $buy_{arg0,from}$ and $purchase_{arg0,from}$ will be equivalent.

A better model would adopt a neo-Davidsonian approach, and aim to learn representations such as the following:

buy $\vdash ((S\backslash NP)/PP_{\mathrm{from}})/NP : \lambda x \lambda y \lambda z \lambda e.rel47(e) \wedge arg0(z,e) \wedge arg1(x,e) \wedge arg2(y,e)$

sell $\vdash ((S\backslash NP)/PP_{\mathrm{to}})/NP : \lambda x \lambda y \lambda z \lambda e.rel47(e) \wedge arg0(y,e) \wedge arg1(x,e) \wedge arg2(z,e)$

As in PropBank, each argument here has non-interpretable predicate-specific labels (as opposed to trying to learn a predicate-independent concept of an *agent* or *instrument*). However, the predicates are induced cluster identifiers. Learning such a representation requires us to both cluster predicates (such as *buy* and *sell*) and align their arguments (for example, the subject of *buy* corresponds to the argument of *sell* supplied by the preposition *to*). Onto-USP [Poon and Domingos, 2010] takes a closely related approach.

6.7.4 Multiword Compositionality

One of the greatest limitations of the work presented in this thesis is that it only models relations between predicates based on a single content word. As such, it cannot capture cluster predicates where a relation is expressed by multiple content words. Some other models avoid this, by finding similarity between longer expressions—for example, Lin and Pantel [2001]'s system learns that *X solved Y* and *X found the solution to Y* are equivalent. Beltagy et al. [2013]'s computes the similarity of phrases based on compositional vector space models of their meaning, and then creates probabilistic inference rules.

There are two ways our approach could handle such compositionality. The simplest approach would be to consider cases such as *found the solution to* to be multiword expressions, and cluster them in the same way as normal predicates. Implementing this would be straightforward, as CCG's generalized notion of constituency means a standard right-branching parse can be re-bracketed so that instead *found the solution to* has a transitive verb-category. This approach is unattractive, as it loses the advantages of compositionality—for example, related cases such as *discovered the answer to* or *X found no solution to Y* would all have to be clustered separately.

A more compositional approach would instead be to decompose the meaning of *solve* into two predicates, so its interpretation literally becomes *finding a solution to.* For example:

solve $\vdash (S\backslash NP)/NP : \lambda x \lambda y \lambda e.find_{arg0,arg1}(y, sk_{\lambda z.solution_{be,to}(z,x)})$

Such decisions could be made based on the non-compositional similarity of *solve* and *find a solution to.*

6.7.5 Light Verb Constructions

Light verbs constructions are frequent in English, and are problematic for the approach described so far. In expressions such as *John took a shower*, *John made a plan* or *John gave a talk*, the main predicate appears to be the noun, and the verb is semantically bleached (or *light*), and contributes little to the meaning. The semantics of the examples could be paraphrased as *John showered*, *John planned*, or *John talked*, and the meanings seem to have little connection to the usual interpretations of *take*, *make* or *give*.

Such cases are difficult, as the syntax and predicate-argument structure appear to be out-of-step, violating one of they key assumptions behind CCG. CCGBank analyses *John took a shower* in exactly the same way as the 'heavy' usage in *John took a book*, and does not capture the idea the key dependency that *John* is an argument of *showering*. As a consequence, there is a danger the system will answer questions like *What did John remove?* with *shower*. My current implementation crudely deals with these cases by treating all instances of common light verbs as being stop predicates, and therefore cannot analyse many frequent relations.

As in Section 6.7.4, there is a simple non-compositional solution approach to dealing with this problem: the system could simply treat instances of *light verb+object* as multiword expressions, and use clustering to infer their meaning. For example, the predicate $take\_a\_shower_{arg0}$ may cluster with $shower_{arg0}$. Such an approach would not generalise well, for example to *took a hot shower* or *took no shower*. Instead, it may be possible to devise a new syntactic analysis for light-verb constructions, in which the noun expects an additional entity argument, and a semantically transparent verb supplies the subject to fill that argument.

6.8 Conclusion

This chapter has greatly developed the model of Chapter 4 by learning complex graph structures over predicates, rather than a simple flat clustering. The major advantages of the new framework are in allowing directional inferences to be represented, and in incorporating information from a diverse range of sources using a discriminative classifier. Both of these improvements contribute to much-improved results over the model of 4. The weak results of the clustering on this task show that relatively few lexical semantic relationships can be adequately captured by clustering, as true synonymy is rare, suggesting that entailment graphs offer a far stronger framework for learning lexical semantics than clustering. The other major advantage of entailment graphs is that they allow a wide variety of features to be incorporated in a principled way—in contrast to clustering, which uses a single similarity statistic.

I have also described how the work could be extended in the future, to build a much richer and more powerful model of semantics. The major limitations of the current model come from a weak model of light verb constructions, only modelling binary relations, not addressing multi-word compositionality, and not modelling temporal semantics. I hope the model described in this thesis will open new avenues in research to find synergies between formal and distributional semantics.

on. As in Section 6.5, I believe a simple more conventional multiword approach to dealing with this problem, the approach could simply treat instances of light verb constructions as multiword expressions, and use clustering to infer their meaning. For example, the predicate *take a shower* ... may cluster with *shower*, ... Such an approach would not generalize well, for example to *took a hot shower* or *took a shower*. Instead, it may be possible to devise a new syntactic analysis for light verb constructions, in which the noun expects an additional entity argument, and a semantically light verb supplies the subject of the argument.

6.8 Conclusion

This chapter has greatly extended the model of Chapter 4 to learn complex semantic structures over predicates, rather than simply relying on clustering. The major advantages of the new framework are in allowing directional inferences to be represented, and in incorporating information from a diverse range of sources using a discriminative classifier. Both of these components contribute to much improved results over the model of 4. The weak results of the clustering on this task show that relatively few lexical semantic relationships can be adequately captured by clustering, as true synonymy is rare, suggesting that entailment graphs offer a far stronger framework for learning lexical semantics than clustering. The other major advantage of entailment graphs is that they allow a wide variety of features to be incorporated in a principled way, in contrast to clustering, which uses a fairly simplistic measure.

I have also described how the work could be extended in the future, to build a much richer and more powerful model of semantics. The major limitations of the current model come from a weak model of light verb constructions, only modelling binary relations, not addressing multi-word compositionality, and not predicting temporal semantics. I hope the model described in this thesis will open new avenues in research to find synergies between formal and distributional semantics.

CHAPTER 7

Conclusions

This thesis has shown that distributional and formal logical semantics do not have to be mutually exclusive, and can be combined in a single model. I have argued that the problem of language interpretation can be divided into the problems of understanding content words, understanding function words, and composing their meanings. Neither formal nor distributional semantics solve all of these problems alone. My major contribution is developing a solution that incorporates the main advantages of each approach. Such a combined model is necessary for high performance on many practical applications, such as question answering. For example, correctly answering the question *Was Obama born in Kenya?* from the sentence *Obama's birthplace isn't Kenya* requires combined distributional and logical semantics.

Compositionality and the semantics of function words already have extensively developed solutions within the linguistics literature, and I have created the first computational implementation of the modern theory described in Steedman [2012]. As a consequence of creating the implementation, I discovered some subtle technical problems in the theory, for which I created solutions. I also developed an algorithm for converting the semantics to standard first-order logic, allowing it to be used with theorem provers. I demonstrated that the implementation can create logical forms with wide-coverage, at a speed which supports web-scale semantic interpretation of text. Example output from the system shows that it can handle linguistically complex constructions, such as right-node raising, and represent scope ambiguities using packed-logical forms. I also showed how it can be used to make complex multi-sentence inferences involving quantifiers, which was not possible for previous work on the dataset.

Neither compositionality nor operator semantics is useful without an understanding of content words. There is little use to being able to compose, negate or quantify words, without a good model of what the words themselves actually mean. Existing work has used hand-built ontologies such as WordNet to model lexical semantics, but despite huge investment in such resources, the problem remains far from solved. I developed a solution in which distributional semantics is used to cluster symbols representing the interpretations of content words, based on similar named-entity arguments. Predicates like *X was born in Y* and *Y is X's birthplace* may have similar named-entity arguments in a large corpus, such as *(Obama, Hawaii), (Napoleon, Corsica) and (Jesus, Bethlehem)*, providing evidence that they express the same concept. Using the cluster identifier as a symbol allows us to generate definitions for *born* and *birthplace* that express the same symbol. Intuitively, the approach aims to assign the same definition to words that have the same meaning. Because the lexical semantics is still symbolic, it integrates seamlessly with compositional and operator semantics. This approach allows lexical semantics to be fully represented in the lexicon, without the need for additional inference rules.

The approach of simply clustering words suffers from a number of limitations. The most serious is that it does not model ambiguity, which is pervasive in natural language. I introduced a new method for modelling ambiguity, by assigning types to predicates and their arguments with a topic model, and assuming that the occurrences of the same predicate with different types are semantically distinct. I gave a novel method that allows these distributions to be combined compositionally during a semantic derivation, by combining distributions at β-reductions, and representing ambiguous terms as 'packed predicates'. The output is therefore a distribution over logical forms, and I introduced a way for succinctly representing the full distribution in a packed logical form. The resulting model shows good performance on a question answering task.

The use of flat clustering does not allow the model to learn lexical relations that only hold in one direction, such as *conquer*$\rightarrow$*invade*. To solve this, I adapted recently proposed methods for building directed graph structures over predicates, in which edges represent entailment. Because entailment is a transitive relation, the graphs are restricted to be closed under transitivity. I introduced a novel method for converting these graphs into an equivalent lexicon, by collapsing cliques into clusters, and making the semantics of a word be the conjunction of the identifiers of all reachable clusters. Instead of relying solely on distributional statistics to determine lexical semantic relations, I showed how to extract supervised training data from an entailment dataset. The use of a supervised classifier means the model can take advantage of a diverse range of features, and not just distributional statistics. I developed a model that uses novel and interesting morphological features, which can learn common diathesis alternations, or how to create deverbal nouns from verbs. These techniques lead to much improved performance on an entailment dataset over the flat clustering model, demonstrating that this approach to learning lexical semantics is far more effective.

I also created a cross-lingual generalisation of this work, which is the first unsupervised model for clustering relations cross-lingually. This work explores the idea that if the clusters correspond to primitive concepts, then they should be language-independent. Cross-lingual clusters can be learnt by aligning named-entities between languages (using Freebase to ground entities), and then clustering predicates with similar arguments. I found a two-stage process was most effective, in which predicates were first clustered mono-lingually, and then the clusters are greedily aligned. Ultimately, I hope that cross-lingual clustering could be used to induce better clusters in each language than monolingual clustering alone, as if multiple languages refer to the same concept, it increases the likelihood that it is a good semantic primitive. However, the current clustering did not achieve this—future word should explore using parallel text for supervision, which may significantly improve the quality of the clustering.

I have also tried to emphasise that the current proposal remains far from a complete solution to computational semantics, and much interesting work remains to be done. Several major challenges are discussed in Chapter 6. The current model can only cluster binary predicates, but I suggest how the model could be extended to handle relations of any arity. It is also important to subcategorize *entailment* into more fine-grained concepts, such as pre-conditions, cause-effect relations, or hypernymy. The current CCGBank grammar does not capture the true predicate-argument structure in some cases, such as light verb constructions, and should be updated to address this. Standard first order theorem provers do not scale to inferences involving large numbers of sentences, and do not allow probabilistic inference, which would prevent the current system from providing inference over a large corpus such as Wikipedia. The recently developed Tractable Markov Logic [Domingos and Webb, 2012] may provide a solution here.

The major conclusion from this thesis is that formal and distributional semantics have much to learn from each other. Distributional semantics can be made more powerful by representing meaning in logic rather than vectors, and formal semantics can benefit greatly from lexical knowledge derived from large unlabelled corpora. The proposed framework, in which logical forms are enhanced with distributional information, offers rich potential for future work.

This thesis has developed a new approach to natural language semantics, which combines the most powerful aspects of solutions from both the linguistics and natural language processing literature. It is the first to incorporate a distributionally-induced lexicon of content words within a wide-coverage implementation of formal semantics, to give a powerful and general model for natural language understanding.

CHAPTER 8

Appendix

8.1 Questions used in Chapter 4

This section contains the question set used in Chapter 4. Answers were annotated by a native English speaker.

What does Marilyn write about?
What does Robert O'Leary run for?
What is Orakzai a stronghold of?
What comes from New England?
What is Provenge a product for?
What ends in Nuevo Laredo?
What collaborates with Boeing?
What is a spokesman for Lucas Bols?
What does FTC reject between?
What is a movement in United States?
What reports from New Orleans?
What is a child of Raj?
What does Timothy M. Dolan arrive in?
What is Safe Kids USA a program of?
What does Ettinger serve in?
What is Colette Bancroft a editor of?
What does Florida run against?
What does O'Hare graduate from?
What do Uruguayans disappear in?
What shows from Hulu?
What works for Disney?
What talks with ESPN?
What looks for Andy?
What is a writer for New Yorker?
What does Terrell Suggs wheel around?
What works with Michelle Obama?
What does Awlaki meet with?
What is Schrade a director of?
What does China leapfrog over?
What does Nugent stand for?

What does White House consult with?
What swoops into Wisconsin?
What is a kind of Robin Hood?
What do Rangers go with?
What is D-N.J. Adler a freshman in?
What arrives in Florence?
What does Nebraska depart for?
What writes in New York Times?
What is a pick from Vanderbilt?
What is a author at St. Petersburg Times Festival?
What is a base for Potash Corp.?
What does CBS talk with?
What does Doyle arrive in?
What serves in Senate?
What is Kennedy a sailor on?
What does Mazorra remain at?
What is a widow of James Clair Jr.?
What is John a home in?
What remains at Guantanamo?
What is a market after Japan?
What agrees with Perricone?
What rides into Boston?
What does BP work with?
What does Vicki Kennedy speak with?
What works for Exxon?
What is Sherman a daughter of?
What is a heart of Switzerland?
What does Timothy F. Geithner arrive in?
What is Olivo a catcher with?
What knows in United States?
What does Rendell land in?
What reports from New York?
What is Chad a coach for?
What is a leader for Patriots?
What is St. Augustine a place in?
What testifys in Portuguese?
What is Sean Daly a critic of?
What files for Social Security?
What is a nation on Earth?
What is a boy of MLS?
What does Red Star contract with?
What meets with Netanyahu?
What testifys before Senate Banking Committee?
What is Fahim a brother of?
What stays on U.S.?
What speaks with Hu Jintao?
What do Puritans land on?
What calls for Congress?
What does Nathan Deal resign from?
What suffers under Hussein?
What does Jeanetta work in?
What is Russia a stakeholder in?
What do Yankees receive from?
What does Beltran meet with?
What does Olofsson come from?
What is Joseph Main a official with?
What is King George V a patron of?
What does Tampa blow by?
What do Mets negotiate with?
What feels about Moss?
What writes Queer Kids?
What does Broadway converge with?
What rules for Big Oil?
What is Steve Persall a critic of?
What works at Treasury?
What is a investor in GM?
What does Charlie Crist converge at?
What lives in Brooklyn?
What is a senator from Manhattan?
What performs in Oslo?
What does Stroughter start alongside?
What is a coach with Baltimore?

What lives in Jamaica?
What does Merkel call for?
What is a teacher at Skinner Middle School?
What presides over House?
What jumps from Jeep Cherokee?
What is Yasar Ozdemir a member of?
What does Lau work for?
What joins with Monterey Institute?
What does Nobel Peace Prize win by?
What wins by Liu Xiaobo?
What does Harry Reid negotiate with?
What is a program of David Horowitz Freedom Center?
What works with Aaron Israel?
What is Montague a officer for?
What is a stakeholder in Kyrgyzstan?
What does CHINA report from?
What is a member of Palm Beach Post Editorial Board?
What does Cathy Connolly arrive in?
What is Morelia a drive from?
What is a youngster in San Diego?
What is Pinera a person in?
What is a son of Pete Muldoon?
What does Lee live in?
What undertakes with Bill?
What is a news for White House?
What is Tom Blackburn a member of?
What does McDaniels believe in?
What is Bob King a president of?
What asks about Kadyrov?
What withdraws from Open?
What does A.O. Scott write in?
What practices with Washington Redskins?
What does Coke live in?
What works with Meetup.com?
What does Shumate slide around?
What does Sam Snead fume at?
What does Nicklaus win in?
What talks about Lee?
What is a president of United States?
What is a draw in AFC West?
What settles in Sand Lake?
What breaks into TJX?
What is a place for Mays?
What looks at Institute?
What works with Pavarotti?
What is a resident of Greenwich?
What is Sean Daly a critic of?
What does Marcelus live on?
What stands on Iraq?
What starts for Cardinals?
What secedes from United States?
What does Spalding graduate from?
What is a commentator for National Public Radio?
What relys on Manning?
What does Denis O'Hare open on?
What is a uproar among Muslims?
What is Cleaves a guard from?
What is a creation of Arthur Conan Doyle?
What is Steve Persall a critic of?
What does Brazil look at?
What works with Mike Leahy?
What is VANCOUVER a tournament at?
What is a candidate in Toronto?
What writes Therese Murray?
What routes from Kabul?
What goes by Lady Gaga?
What does Tim Anderson move from?
What is a influence on Steve?
What teaches at Harvard Business School?
What is Thomas a director for?
What does Atlanta Hawks part with?

What does NFL start for?
What do Democrats convene in?
What convenes in South?
What pays for Greece?
What reports from Bishkek?
What is Bakhtiar a minister under?
What reports in Washington Post?
What does BABA BOOEY abate with?
What does Pasco County Library Cooperative work with?
What does Salazar call on?
What is a native of San Antonio?
What is Barry Lutz a cousin of?
What is a area for Republicans?
What performs at East Room?
What is a investigator at National Institute?
What does Mitchell meet with?
What does Ensign speak with?
What does Lindsey Vonn arrive at?
What is a columnist for Denver Post?
What originates in Grand Canyon?
What votes for Pelosi?
What does McCaskill join with?
What does Martin report in?
What does Betty live in?
What does Bellamy sign with?
What do Chargers look at?
What looks at L.A.?
What do Miracles happen in?
What pairs with MTV?
What is El Camino a school in?
What starts for Rams?
What is Mark Kiszla a columnist for?
What does Charles Smith head for?
What does Taliban route from?
What does LeBron James heel in?
What does Lee prosper with?
What is a coach at West Virginia?
What is a senator in Illinois?
What cooperates with Renault-Nissan?
What is Robert M. Hertzberg a co-chair of?
What works with Israel?
What does Sam Dolnick report from?
What is a deal with Colorado Rockies?
What is Miss Universe Organization a company of?
What is McGregor a end at?
What does Kenya arrive in?
What is Treacy a son of?
What is Colette Bancroft a editor of?
What is a critic of St. Petersburg Times?
What does Bauer work on?
What does Cohl think beyond?
What is Germany a partner in?
What does Ayestaran live in?
What withdraws from Vieques?
What does Sunderland compete with?
What is a experiment for Viacom?
What is Woody Paige a columnist for?
What is a owner of Clearwire?
What moves from Merritt Island?
What is a presence in Kasumigaseki?
What does Beara rank above?
What is a editor of New Republic?
What is Postal Service a employer after?
What is a manager for Mark?
What does Keselowski call for?
What works at St. John?
What does Deevy live in?
What does Greer fly with?
What does DeMint run for?
What is a quarterback for Cowboys?
What does Jennifer stay in?
What does McInnis appear on?

What is a rookie from Texas?
What is Dr. Berger a customer of?
What is Wetherell a receiver at?
What is Jordan a student at?
What is Lindsey Vonn a star of?
What uses for EMI?
What does Slovenia qualify for?
What does Fischer leave for?
What leaves for Credit Suisse?
What does Livonia Republican fly aboard?
What is Charles Schwab a face of?
What studys in Israel?
What does Mullen meet with?
What is Bedford Post Inn a hour from?
What is John Henderson a columnist for?
What does Automotive News report from?
What is a coach for Kim?
What is Marie Valencia a president of?
What spars with Republicans?
What is Gregoire a ally on?
What resigns from HP?
What does Bill Richardson fly into?
What reports from Houston?
What is Allen a worker from?
What lives in Mumbai?
What does Newton land in?
What is Lanchester a admirer of?
What does Sayle resign from?
What remains at Guantanamo?
What meets with Steadman?
What is Hirsch a son-in-law of?
What does Mullen meet with?
What is a director of Brant Publications?
What does Lecavalier center for?
What talks with Times?
What does Joe Biden arrive in?
What works with Palomar College?
What belongs in Denver?
What is a president of IMA?
What graduates from University?
What is Trattou a guy from?
What does Miller resign from?
What trades for Carter?
What does Obama arrive in?
What does Camden work with?
What does Microsoft remain in?
What stands between Rays?
What is a editor of Albany?
What heads into U.S. Open?
What is Dave Krieger a columnist for?
What reports in International Herald Tribune?
What does Mascheroni work for?
What graduates from James Madison High School?
What is a rock for Karls?
What is a leader of Afghanistan?
What is Ogilvy a presence on?
What do Americans blow into?
What is a resident of Mission Viejo?
What is a nominee in Illinois?
What writes in International Herald Tribune?
What appears with Murray?
What does Janessa Goldbeck work in?
What does GE work with?
What forgets about Rodney Stuckey?
What does Lysacek edge out?
What does Kathy Brearley testify before?
What is South Africa a host for?
What does Washington work for?
What is Joe Lombardi a grandson of?
What battles in Superior Court?
What is a daughter of Oliver Warbucks?
What calls for Gulf Coast Restoration Plan?
What does O'Donnell run for?

What do Knicks drift among?
What does Morefield graduate from?
What is a friend of Allison?
What writes in Times?
What pens in Red-Headed League?
What is Chile a country in?
What interviews on MSNBC?
What jails in Iran?
What do Giants receive from?
What writes in Times?
What does Barack Obama meet with?
What does Bauer live in?
What does El Tovar Lodge perch on?
What skates in Tampa?
What joins with Lamar Lundy?
What confers with Arab?
What lives in Troy?
What is Carroll a chaplain at?
What differs from Ellis Island?
What wins at Raymond James Stadium?
What is Dave Krieger a columnist for?
What does Malcolm Forbes live at?
What does Christie Collbran remain in?
What arrives in Lebanon?
What does Zazi drive from?
What do Rays arrive in?
What does Gelber write?
What looms over Scarlett?
What designates by United States?
What is a rookie in NBA?
What is Onyango a aunt of?
What is a coordinator at Youngstown State?
What is Romo a quarterback for?
What is a part of Federation?
What is a member of Palm Beach Post Editorial Board?
What stands behind Kabul Bank?
What returns from Pakistan?
What votes for George W. Bush?
What returns from Mexico City?
What is a president with College Board?
What does Rudolph W. Giuliani campaign with?
What does Big Boi collaborate with?
What is Buick a brand after?
What does Karachi rely on?
What does Sheldon kneel beside?
What is Alexandre a member of?
What is a columnist for Denver Post?
What is a agency of United Nations?
What is Vinas a ace of?
What is Kirby a talent at?
What is a champion in New York?
What do Bucs practice at?
What does Hagan live in?
What is a coach at Temple?
What is Morris a officer in?
What flys aboard Air Force?
What writes in Times?
What clashes with Phelps?
What is Russell Long a whip of?
What does Chung work in?
What comments on Meehan?
What is a chairwoman of Senate Agriculture Committee?
What lives in Vilcabamba?
What meets with Dalai Lama?
What waits for Longoria?
What survives in Tampa Bay?
What does Emanuel light into?
What is Thomas a coach at?
What does McMahon level at?
What does Bill Marriott meet in?
What travels in Middle East?

What works for CIA?
What does McCoy testify before?
What is a professor at Harvard Law?
What is Shahzad a buyer of?
What is Peterman a legislator from?
What does Florida Dance Festival move from?
What arrives in North Korea?
What is a director of Philharmonic?
What is Gifford a writer from?
What is a player for Mets?
What does Judge Richard A. Posner clerk for?
What is a cousin of Oppenheim?
What partakes of Coors Light?
What does Latson report from?
What writes in New York Times?
What scowls about New Jersey?
What does Michael Silverman think about?
What is Clegg a agent in?
What shuttles around Jerusalem?
What does Justin Olsen hop behind?
What does al-Bashir charge by?
What is Mr. Tully a publisher of?
What is a bank in Ireland?
What signs with Yankees?
What plans FOR National Aeronautics?
What stands at Audubon Nature Institute?
What brings from Colorado?
What is a president of CBS Television Network?
What meets with Hillary Rodham Clinton?
What does Barack Obama meet with?
What does Hayes Jenkins win at?
What does Havasupai originate in?
What is Thomas a part of?
What meets with Izzo?
What is a representative from Providence?
What do Democrats vote in?
What runs for Congress?
What is Ash a investor in?
What is a investor in Bayview?
What does Abdullah meet in?
What races in Florida?
What does Reid talk about?
What does Krajewski work at?
What expects from Grand Prix?
What does Darragh perform alongside?
What is a director of Boston Lawyers Group?
What lands at Florida International University?
What is Reynolds a executive of?
What is a agent in Columbus?
What does Icahn lean on?
What flys with Tuskegee Airmen?
What coaches at Toledo?
What is Stephen Alexander a chairman at?
What produces from Social Security Administration?
What is Pirozhkova a edition in?
What is Bowser a teenager during?
What is a antithesis of Berlin?
What is Chad Doll a bartender in?
What is a chairman of Protean Holdings?
What is a obstacle in U.N. Security Council?
What writes in New York Times?
What does Barone live in?
What is a part of U.S. Virgin Islands?
What does Peter Watrous write in?
What does Maddon look at?
What does HP work WITH?
What expects in Beethoven?
What does Catan move from?
What is Tom Blackburn a member of?
What does Drumm resign from?
What is Stern a member of?

What does Goya work with?
What does New Hampshire vote for?
What bolts from GOP?
What does Lawrence live in?
What exhibits at LACMA?
What is a brainchild of Irwandi Yusuf?
What is a denomination in United States?
What reaches around Berrian?
What is a correspondent for New York Times?
What is a columnist for Denver Post?
What does Sweeney scowl about?
What does Don Kreamer vote for?
What is a critic of Times?
What talks with Richard Sestak?
What does IRS look at?
What does Hillsborough County School District undertake with?
What is ElBaradei a figure in?
What is Big Red a seed from?
What leaves for Harvard?
What is a president of National Ocean Industries Association?
What graduates from Brown University?
What runs from James O. Eastland?
What does Germany prepare for?
What does Gordon Brown meet near?
What does Octavio Paz come from?
What looks for Afghanistan?
What does Ben Brantley write in?
What is a end at Colorado State University?
What is a focus of St. Petersburg Times Festival?
What lives in India?
What does Bondi edge out?
What comes from Chicago?
What is Joel Brinkley a correspondent for?
What is a editor of Austin American-Statesman?
What does Boise State bolt for?
What is a evocation of Liverpool?
What is Tampa Bay a team in?
What does Civil Rights Act work alongside?
What builds in Chattanooga?
What is a criticism of GM?
What calls for United States?
What is Higgins a scorer for?
What does White House rely on?
What does GM focus on?
What reports from Washington?
What is a fixture in Senate?
What does Hefner talk about?
What works with Kerry?
What is Peter Lewis a editor at?
What is a forest in United States?
What coaches with Steve Addazio?
What does Hillary Rodham Clinton speak with?
What does Gaiutra Bahadur write in?
What does Napolitano testify before?
What is Felix Carroll a writer for?
What draws in Connecticut?
What is Harley a director at?
What do Italians stick in?
What walks onto George Washington Bridge?
What publicizes on Facebook?
What is Samantha a daughter of?
What drowns with Karzai?
What writes in Times?
What does Viktor Kassai preside over?
What is Pinot Grigio a benchmark for?
What is Michael Yakes a mayor of?
What is Medicaid a deal in?
What is Brody a wife of?
What does Khloponin ask about?

What does Mian graduate from?
What is Marco Rubio a member of?
What is a topic on Weibo?
What is Feldstein a chairman of?
What is ASL a part of?
What does Matt Duchene arrive in?
What returns from China?
What wins over Mississippi State?
What agrees with Krzyzewski?
What follows by Hutchison?
What is Tom Marshall a writer for?
What does Jeremy W. Peters report from?
What is a daughter of Dan Reeves?
What drops below Redskins?
What is a tenant in Avery Fisher Hall?
What happens in Massachusetts?
What does Jim Tracy look at?
What does A.O. Scott write in?
What is a governor in Kunduz?
What arrives in Portland?
What is a longshoreman in San Francisco?
What is a manager of Kansas City Royals?
What is Richardson a project of?
What is Sander M. Levin a chairman of?
What lands at Bagram Air Base?
What does Shahzad return from?
What is Mark Kiszla a columnist for?
What is Toomey a banker for?
What is James Baker III a operator on?
What is a laughingstock of NFL?
What does Mason work at?
What writes in New York Times Book Review?
What lives in England?
What does Mattioli enrol at?
What does Dietz agree with?
What does Dan Aykroyd team with?
What works for Toyota?
What does Edward Kennedy run for?
What is Carpenter a president of?
What is a republic in Central Asia?
What speaks about Fed?
What does National Conference meet in?
What qualifys for Olympics?
What is Germany a man of?
What is Woody Paige a columnist for?
What does Whitacre speak at?
What writes on About.com?
What does Dalai Lama arrive in?
What does Rogers score at?
What is a executive at Ford Motor?
What studys at UCLA?
What does Blumenthal serve in?
What is a critic of Times?
What is a critic of China?
What tackles for USC?
What replys from Houston?
What does Meyers enjoy following?
What does James stay in?
What is a teacher in West Palm Beach?
What is Elena Kagan a student at?
What works on Capitol Hill?
What does Mr. Salinger serve in?
What does Mikenley dream of?
What is Mark Kiszla a columnist for?
What advances unlike Dinara Safina?
What is Mark Kiszla a columnist for?
What stands before Bill Ford Jr.?
What does Open compete in?
What does Brad Ellsworth run for?
What is Laland a biologist at?
What inquires about Carlos Queiroz?
What does Mihos write?
What is Carey a president at?

What does Association convene in?
What is a partner in Bracewell?
What honeymoons on Sea Island?
What does Ellen land in?
What does CenterPoint wait for?
What does Clausen start against?
What operates in India?
What is Eskendereya a winner in?
What testifys before Congress?
What is Ryskamp a director of?
What is Bill a group of?
What is AEI a subsidiary of?
What performs with Santana?
What is a columnist for Denver Post?
What votes with Republican Party?
What does Freyman graduate from?
What is WellPoint a example of?
What docks in Tampa?
What does Michael J. Lohman arrive at?
What flys through Tulsa?
What does George A. Papandreou meet with?
What meets at Tropicana Field?
What is Tom Blackburn a member of?
What volunteers at Lenox Hill Hospital?
What does Carter serve in?
What is John Solomon a reporter with?
What finds by Loria?
What does Keal talk about?
What is Elena Kagan a student at?
What looks at Cody?
What is a professor at University?
What does Elizabeth Gilbert write of?
What does Wolf speak with?
What do Colts lose in?
What works in Afghanistan?
What identifys with Iran?
What writes on Huffington Post?
What is Kennedy a sailor on?
What does Clifford J. Levy report from?
What is a president of Hudson Castle?
What does Goldman bet against?
What is a partner at Hogan?
What is a athlete among Rays?
What lives in West Bloomfield?
What is Erin a intern at?
What chokes against Ghana?
What testifys before Congress?
What is a analyst at Basketball-Reference?
What is Tim Foley a student at?
What is Wilhelm a teammate of?
What does Agassi write of?
What is Ellis a disciple of?
What does Zobrist hear about?
What expands in Switzerland?
What does David Garrard intend for?
What sings at Cafe Society?
What is a member of National Commission?
What is a city in Punjab?
What meets with Bobby Jindal?
What does Robert Allenby withdraw from?
What is Europe a slogan of?
What does Mattek-Sands excel in?
What does Peter Baker report from?
What does Manohla Dargis write in?
What is Shirley a member of?
What is Baldwin a son of?
What arrives at San Francisco International Airport?
What recalls in U.S.?
What works for Linden Lab?
What wins in Europe?
What does Henry VIII break with?
What does Barack Obama meet with?
What does Saints arrive in?

What is Obama a senator in?
What do Knicks plan for?
What does DSi XL arrive in?
What loses in Foxborough?
What graduates from Harvard?
What is a figure at Roulette?
What cheers for Netherlands?
What is a head of National Ocean Industries Association?
What does Williams board at?
What tucks in Indonesia?
What is a blow for Corzine?
What compares with MFA?
What is a housewife of D.C.?
What works with White House?
What is a teacher at Agoura?
What does Saab build outside?
What is a champion with PGA Tour?
What surrenders at Cumberland Federal Correctional Institutional?
What merges with Metavante Technologies?
What competes in Salt Lake City?
What is a automaker behind BMW?
What does Charlie Crist bolt from?
What does International Monetary Fund meet in?
What is a dean of Harvard Law School?
What is Joel Brinkley a correspondent for?
What does Hough coach for?
What writes in Le Monde?
What merges with United Airlines?
What is a critic of Times?
What is a goalie for San Jose?
What drives for Lotus?
What appears before Political Action Conference?
What apologizes FOR Carl Paladino?
What is Lennie Bennett a critic of?
What does Blumenthal serve in?
What lives on MTV?
What contracts with Defense Energy Supply?
What does Paul McCartney perform at?
What re-signs with Denver?
What is a player on Team USA?
What coaches at USC?
What does No Child leave Behind?
What remains at Tropicana Field?
What does Rodriguez bolt for?
What arrives in New Delhi?
What lives on Rue Macajoux?
What exiles in India?
What does Faisal settle in?
What works with Stan?
What does Hellickson remain at?
What does Sarasota win over?
What counts on Delhomme?
What is a sister of Cruz Bustamante?
What is Goolsbee a professor at?
What moves into Nuevo Laredo?
What is Woody Paige a columnist for?
What retreats from Asia?
What is Berkowitz a follower of?
What acquires from Toronto?
What does Cathy Connolly meet with?
What sides with Democrats?
What does Bud Perrone stay in?
What does Virginia Heffernan write in?
What do Bucs count on?
What does Hossa sign with?
What serves in House?
What calls for Congress?
What does Democrat win in?
What does Philip Langridge die in?
What does Steve Ipsen run against?

What reports from Houston?
What does Jessica Park graduate from?
What graduates from Mount Greylock High School?
What talks with Barack Obama?
What graduates from Williams College?
What resides in Elysee Palace?
What pictures between Betty?
What does Lazy Beetle Bailey star with?
What does Charpak join with?
What does John Hudson report on?
What is George Kiefer a assistant at?
What is Mexico City a jurisdiction in?
What is a jurisdiction in Latin America?
What does Washington report from?
What negotiates with White House?
What is Gary Indiana a writer in?
What does Shami vouch for?
What writes in Times?
What does BUFFALO report from?
What is Chapman a guest of?
What does Naipauls meet with?
What works for Pernod Ricard USA?
What is Johnny Boy a descendant of?
What arrives in New York?
What does Teixeira land in?
What is Kresa a director at?
What runs into Pfizer?
What moves from Boston?
What is a editor of Albany?
What is BMG Rights Management a venture between?
What is a brother of Daleisha Carn?
What does Levin argue with?
What is a editor of Times?
What does Andy Warhol look for?
What does Dana Milbank write in?
What is a supporter of Muqtada al-Sadr?
What does Delta merge with?
What does Wade Davis pitch in?
What arrives at Open?
What works at General Foods?
What is Saskatchewan a base for?
What is Israel a member of?
What is Elway a contributor on?
What does Jenkins win against?
What is Elena Kagan a dean of?
What files from Alabama?
What is a surgeon from Crystal Falls?
What is Dave Krieger a columnist for?
What is a editor of Times?
What is a champion of Kleibacker?
What does A.O. Scott write in?
What is a president at Conde Nast?
What is Peyton Manning a player in?
What bashes on Blake?
What does Youkilis meet with?
What is a kind of Nantucket?
What runs for County Commission?
What fumes at Wal-Mart?
What does Graham Bowley contribute from?
What is McCotter a chair of?
What calls for China?
What is Anton Renault a reporter for?
What does Manohla Dargis write in?
What is John Henderson a columnist for?
What does Dwight Howard tug at?
What is Bruce Froemming a umpire at?
What does Reuters report from?
What is a emeritus at National Zoo?
What meets in Vietnam?
What is Florida a focus of?
What do Romans talk about?
What speaks at Political Action Conference?

What does Karzai talk with?
What is Bryan a professor of?
What prepares for World Expo?
What is Tom Blackburn a member of?
What is a also-ran in United States?
What arrives in Washington?
What is a pastor of New Birth Missionary Baptist Church?
What is Zuckerman a supporter of?
What warns about Bayou?
What does Robert M. Gates arrive in?
What does Doug Busch invest in?
What is a critic of Times?
What is Sandberg a veteran of?
What does Google team with?
What works for U.S. Army?
What arrives in Tampa?
What does Affiliated Transaction Committee meet in?
What arrives in South Africa?
What calls from Washington?
What works with Alliance?
What speaks at Brookings Institution?
What is a colonel from Army Reserves?
What do Canadians skate at?
What meets with Leibman?
What contemplates following Beilein?
What is David D'Arcy a correspondent for?
What is Cano a child of?
What is a editor of San Antonio Express-News?
What is Jack a junior at?
What is a legislator from St. Petersburg?
What does Tiffany reside in?
What is a producer with Peter Jankowski?
What is Bill Barton a coach at?
What is Miller a star in?
What is a star in Miami?
What is a stage in Washington?
What does Lucic tangle with?
What is Mr. Williams a professor at?
What wins at Pebble Beach?
What is a chief at HUD?
What is Rich a critic for?
What runs for Congress?
What does Tom Kelleher come from?
What is Craig Updyke a manager for?
What arrives at Boston Marathon?
What does Joshua M. Stone appear in?
What is a tale of London?
What is a critic of St. Petersburg Times?
What does Wendel settle on?
What begins in English?
What does Bob Bradley slip into?
What is Guadalupe Herrera a eighth-grader at?
What focuses on Northeast?
What is a resident of United States?
What hails from Birmingham?
What is Tampa Bay Bucs a speaker at?
What runs between Santa Fe?
What does Taliban operate in?
What returns from Pakistan?
What leaves for China?
What withdraws from Action Partnership?
What emerges from Communism?
What does Stover break into?
What is a talent at Comics?
What is a automaker behind GM?
What do Turks feel toward?
What is Fisher a coordinator at?
What is Jackson a artist in?
What does IBM team with?
What is Tom Blackburn a member of?
What writes in Washington Post?

What opts for Seguin?

What remains in Toronto?

What arrives in United States?

What is a officer in Pacific?

What does Blumenthal run for?

What races past United States?

What opts for Taser?

What does Sadik-Khan speak at?

What is Lourd a page at?

What is Posey a star for?

What is Lowe a place for?

What is a student at UCLA?

What does Jim O'Rourke live in?

What is Dr. Carlos A. Zarate Jr. a investigator at?

What does Marquez arrive in?

What is Bigeleisen a member of?

What warns from Afghanistan?

What sits on University?

What is a masterstroke for Sabean?

What is Afghanistan a version of?

What does Obama meet with?

What reports from New York?

What does Harrison sign with?

What works for CBS?

What is Kuchar a champion with?

What is a lawyer at SEC?

What does Ian Urbina report from?

What works with Yanick Alleno?

What reports in Times?

What folds in United States?

What does Army storm into?

What does Barack Obama state in?

What do Bucs go with?

What is a jockey at Mango Radio?

What writes in Times?

What suggests in Europe?

What is Udall a supporter of?

What is Gibney a match for?

What does Kottkamp work for?

What does Sen. John McCain meet with?

What signals ON Iran?

What does Michael R. Gordon report from?

What abates with Chad Millman?

What lives with Feldman?

What is a child of Dolph?

What does Rubio benefit from?

What is Sean Daly a critic of?

What does Rachel Alexandra lose in?

What does Rafanelli live in?

What lives in South End?

What does Tucker appear in?

What is Lahore a city in?

What debuts in New York?

What does Cabrera glare at?

What does Holmes vote for?

What appears with Branca?

What does DreamWorks march into?

What does Holladay vote for?

What does Mark Kiszla file from?

What is a market after China?

What does Francisco Liriano start for?

What is a student at Marefat High School?

What storms into World Series?

What is a executive of Chicago Public Schools?

What does Omar Lopez walk into?

What meets with Ted Branch?

What goes into St. Louis?

What is Lugano a center in?

What is Yvonne Walker a president of?

What waits for PUC?

What does Houston Astros pitch for?

What does George J. Mitchell meet with?

What fights in Afghanistan?
What competes in BMX?
What is China a neighbor of?
What is a anchor at Fox News?
What does Ashton know in?
What is Paulin-Ramirez a wife of?
What does Association meet in?
What does Polgar live in?
What writes with Benjamin?
What does Navy withdraw from?
What is a instructor at Upsala College?
What is Germany a market in?
What travels from Texas?
What is a senator in Illinois?
What is a mentor for Cabrera?
What do Olympics approach in?
What works with Isabelle Huppert?
What does Kurt Wallander travel by?
What is Jeff Saitas a lobbyist for?
What meets with Tony Blair?
What writes in Times?
What does Marsha Collier live in?
What is Cheung Kong Infrastructure a part of?
What does Tyson discriminate against?
What works with Bruce Allen?
What does Woody Johnson speak with?
What does Delahunt travel in?
What competes in Olympics?
What qualifys for World Cup?
What wins in Massachusetts?
What resurfaces at Hofstra?
What does Miller acknowledge at?
What is a senator from Orlando?
What does Errol Kerr compete in?
What arrives in Colorado?
What does Michael Billington write in?
What is a capital of U.S.?
What does Farhi Saeed bin Mohammed capture in?
What do Americans wait for?
What is BP a producer in?

8.2 English Questions used in Chapter 5

This section contains the French question set used in Chapter 5. Answers (in French) were annotated by a native French speaker who speaks fluent English.

XYZ inhabits Slovakia
XYZ leaves Ireland
XYZ starts Heathkit
Bullies reaches XYZ
XYZ pins Darren Matthews
France means XYZ
Joni Mitchell visits XYZ
XYZ accepts Christianity
Japan looks to XYZ
South Africa rests XYZ
XYZ travels to Africa
Monteux works with XYZ
Rue Morgue Magazine interviews XYZ
Ghassan Tueni returns to XYZ
XYZ works in Roatán
Radostin Stoychev replaces XYZ
XYZ bases The Defense
XYZ goes on Tennessee
XYZ is a director of Schola Cantorum
XYZ captures Chiang Mai

XYZ stars to The Stars
XYZ includes John Gutfreund
Willy Vandersteen chooses XYZ
Cichiva River is a tributary in XYZ
XYZ comes from Wembley
XYZ borders Gmina Czemierniki
XYZ is a member of Abstract Rude
XYZ uploads to Y-O-U
Pacoima Wash continues to XYZ
XYZ returns to Celebrity Fit Club
XYZ is a village in India
XYZ drives United States of America
XYZ enrols at Yale University
XYZ succeeds Keith Joseph
William Careless serves at XYZ
Henry Doetsch leaves XYZ
Liu Shan sends XYZ
Admiralty requisitions XYZ
Priscian cites XYZ
XYZ names Automatic Data Processing
Dominique Lapierre renovates XYZ
Pope John Paul II apologises of XYZ
XYZ links Montreal
XYZ availables on Compact Disc
XYZ arrives on Earth
Robert Hübner vses XYZ
Emperor Shomu remotes XYZ
XYZ demobilizes in England
XYZ is a settlement in California
Joe Diffie meets XYZ
Canada drains into XYZ
Mary Bonnin enlists in XYZ
XYZ reports to Clement Wood
Davison attends XYZ
XYZ gains South Hornsey
Negley meets XYZ
Chad Valley is an area of XYZ

XYZ becomes Steeler
XYZ returns to DEL
INS Sarayu serves with XYZ
XYZ announces Phillip Burrows
Lindsay Lohan portrays XYZ
XYZ becomes African American
XYZ campaigns for Barack Obama
XYZ is a town of Hounslow
XYZ moves to Wikia, Inc.
New York nicknames XYZ
XYZ attends Pennsylvania
XYZ reaches Davey Allison
XYZ transfers Sergey Korolyov
Bob Peak teaches at XYZ
Ardanuç is a village in XYZ
Serge Brammertz replaces XYZ
XYZ sees Gillian Polack
XYZ confirms Zeuss
XYZ dissolves Euroregion
XYZ marries Sylvius Leopold Weiss
Paul Sturrock brings XYZ
XYZ records What You Know
Henry Wadsworth Longfellow publishes XYZ
XYZ reveals to Earth
PennYo performs at XYZ
XYZ enters Waseda University
Thomas Kyd is a son of XYZ
Australia matches XYZ
John C. Frémont learns XYZ
XYZ attends Farragut High School
Padthaway naracoortes XYZ
Marshal acquires XYZ
XYZ visits Australia
XYZ moves to San Antonio
XYZ works for William Randolph Hearst
Leon Surmelian goes to XYZ
Singapore is a state of XYZ

XYZ trails Mary Norwood
Aberystwyth University attends XYZ
Jay tricks XYZ
Edward Bellamy pens XYZ
Rainey battles XYZ
Encantadia returns to XYZ
Muso Gonnosuke encounters XYZ
XYZ is a team from Canada
XYZ describes Harley Psalter
Giuseppe Ottavio Pitoni arrives in XYZ
Sarah McCarron attends XYZ
XYZ goes Dave Lovering
XYZ populars with Ernest Hemingway
XYZ competes for Sweden
Mohsin Hamid is a finalist for XYZ
XYZ premiers in New York City
Schlossplatz is a square in XYZ
XYZ confirms Aki Maeda
Australia defends XYZ
XYZ leaves Patton Boggs LLP
XYZ terminates in Grant-Valkaria
XYZ sails to Africa
XYZ announces Boeing
China helps XYZ
XYZ publishes Indonesia Handbook
Mochdre is a village in XYZ
Roger Dubuis collaborates with XYZ
Turner Network Television negotiates XYZ
Crown Limited informs XYZ
XYZ pressures Vichaichan
XYZ invades Soviet Union
Gary Hines serves on XYZ
XYZ dies Hamar
Abihu is a son of XYZ
Hugh Douglas marches against XYZ
Delta Air Lines starts XYZ
Paul Kadak works for XYZ

XYZ parodies Tokimeki Memorial
XYZ is a start of Davar
Houston Rockets drafts XYZ
XYZ develops Howard Hughes
XYZ moves to Derbyshire
Moscow studies in XYZ
XYZ overtakes California
Carthage destroys XYZ
Roigheim survives XYZ
XYZ is a battle of American Civil War
Islam is an extension of XYZ
Davar means XYZ
British Broadcasting Corporation contracts XYZ
XYZ actives in The Association
Kenny Young recruits XYZ
Bruce Springsteen states XYZ
Modwheelmood releases XYZ
Some enters XYZ
Wyryki lies of XYZ
Charles-Pierre Colardeau returns to XYZ
XYZ goes on Lawrie McMenemy
Dorothy Hill attends XYZ
XYZ stretches to Pett
Batu Khan leaves XYZ
XYZ is an engine from Microsoft
XYZ is an attendance in Malmö FF
United States of America arrives at XYZ
Shinya Aoki fights XYZ
XYZ moves from Forbes Field
Nathan Hindmarsh immigrates from XYZ
José Basora meets XYZ
XYZ starts with Barani Department
XYZ works David Morales
Bill Evans is a thing in XYZ
XYZ stars Suresh Oberoi
XYZ regards M.o.v.e

XYZ visits Scotland
Paper availables in XYZ
XYZ teaches at Brandeis University
Christopher Walken sings XYZ
XYZ meets in London
XYZ announces Barking
XYZ buys Boston Red Sox
XYZ attacks Israel
ASCII Media Works publishes XYZ
XYZ occurs from New South Wales
United Kingdom withdraws from XYZ
XYZ withdraws from International Olympic Committee
XYZ is a band from Finland
XYZ 'blessings King
XYZ instructs Shankar Kistaiya
XYZ writes to Muhammad
Sleeping Satellite goes to XYZ
XYZ records Manchester Square
XYZ releases Need You Now
Shahbaz Sharif includes XYZ
XYZ goes in Eger
British Broadcasting Corporation shows XYZ
XYZ stars for Metro-Goldwyn-Mayer
Gary Williams beats XYZ
XYZ wins World Cup
Garris joins XYZ
James Pinnock joins XYZ
The Cardinals is a member of XYZ
XYZ joins Raith Rovers F.C.
XYZ beats Mickey Rooney
John Sutter leases XYZ
Arabic Language is a language in XYZ
Solon works in XYZ
XYZ works in Japan
Kougny Department is a commune of XYZ
XYZ distinguishes Professor
XYZ represents India
XYZ drives Germans
XYZ goes on Hermann Buhl
Indianapolis scores with XYZ
Rey Bucanero feuds with XYZ
XYZ ports to Xbox Live Arcade
Shaun Morgan joins XYZ
XYZ records Warren G
XYZ works at Kent State University
Earth returns to XYZ
McColl joins XYZ
Humbert, Pas-de-Calais leaves XYZ
Citigroup buys XYZ
XYZ is a region of Prussia
Cherubs sails in XYZ
Ralph Smart produces XYZ
XYZ operates from Rambouillet
All Blacks thrashes XYZ
XYZ is a district in St. Charles County
Fourth Macedonian War fights from XYZ
XYZ moves to Berlin
XYZ ensures East Bengal
Ray Charles titles XYZ
St. James's Gate is a home of XYZ
XYZ is a building in Philadelphia
XYZ conquers Association for Intercollegiate Athletics for Women
XYZ wears Naoki Maeda
Cortés returns to XYZ
XYZ is a mayor of Evansville-Vanderburgh School Corporation, Vanderburgh County, Indiana
XYZ leads Co-operative Championship
XYZ is a tributary in Romania
New York Yankees wins XYZ
Indian Army leaves XYZ
XYZ becomes The Association

XYZ exits RCA
Warnock disappoints with XYZ
XYZ is a mountain of Scotland
XYZ anticipates Thomas Aquinas
Arthur Blomfield builds from XYZ
Western Telegraph borders XYZ
XYZ designs Wharncliffe Viaduct
XYZ beats Austria
Holy Trinity Monastery is a monastery in XYZ
XYZ establishes Lowell National Historical Park
XYZ returns to Chicago
Amritsar translates from XYZ
XYZ drafts Ricky Williams
Stefan Batory Foundation establishes XYZ
XYZ views Gundi
Spencer Day opens at XYZ
XYZ serves Empress Dowager Ding
Stanislas Wawrinka defeats XYZ
Bristol Rovers F.C. joins XYZ
XYZ is a village of Vietnam
XYZ participates in World War II
Cass Technical High School is a school in XYZ
Lewis remains in XYZ
XYZ continues C60
XYZ presents Richard Dunwoody
Buckley purchases XYZ
Gloucester Green is a square in XYZ
Peters quits XYZ
Darius James quotes XYZ
Sordello arrives at XYZ
XYZ owns United States of America
Weiner interesteds in XYZ
Floyd Allen beats XYZ
Government Street constitutes XYZ

Calkins Media publishes XYZ
Honduras replaces XYZ
Germany invades XYZ
XYZ is a 21 for Mac OS X
XYZ interviews Chinese
XYZ uses Napoletano-Calabrese Language
Yeager lives in XYZ
Mbabaram Language is a language of XYZ
XYZ defeats Amélie Mauresmo
XYZ writes Rhapsody in Blue
XYZ signs Simone Loria
Mauritania recognizes XYZ
Eva Perón visits XYZ
XYZ lies of Jihlava
XYZ uses Davar
Mazarin studies in XYZ
Plum returns to XYZ
XYZ finishes The Muppets Take Manhattan
Anarchy Online consists of XYZ
Partibrejkers performs in XYZ
XYZ spreads Zoroastrianism
XYZ crowns King
XYZ is a figure in Ireland
XYZ operates Veolia Transport
Davar is a shape for XYZ
XYZ collaborates The Connoisseur
The Lucy Show is an episode of XYZ
Y-O-U sees XYZ
XYZ believes in Allah
Roland wins XYZ
XYZ headquarters in New York
XYZ merges into Bank of America
XYZ signs Travis Kvapil
Aluminij is a company from XYZ
XYZ hourlies to Bradford
XYZ accepts Russia
XYZ sculpts Tolerance Monument

Liu Xin sees XYZ
XYZ meets Italy
Simon Pedersen Holmesland sits in XYZ
XYZ moves with Pat Pottle
XYZ caves Elephanta Island
Major availables at XYZ
XYZ forms Oklahoma
XYZ is a manufacturer in Earth
Rose Creek is a stream in XYZ
Overtones travels to XYZ
XYZ emigrates to France
Garrett Morris stars XYZ
XYZ invades Earl of Sutherland
Terry Slesser joins XYZ
XYZ inherits Hainaut
Andrew Young opposes XYZ
XYZ meets Ralph Waldo Emerson
XYZ begins in San Antonio
XYZ marries Patrice Wymore
Cornelius Gemma dies in XYZ
James Stewart stars in XYZ
Francisco Franco leaves for XYZ
Fumio Nanri lives in XYZ
Robert Earl announces XYZ
Marlon Fernández returns to XYZ
Warren Cormier is a ceo of XYZ
XYZ writes Mobile Suit Gundam
XYZ embarks on Far East
Brad Sham replaces XYZ
XYZ involves with William Aberhart
Duke University recruits XYZ
XYZ portrays Marcella
XYZ sells Flanders
Irm Hermann stars on XYZ
XYZ falls to Duke University
Miles Copeland III understands XYZ
XYZ leaves N.W.A
Dragon's Lair joins XYZ
XYZ lists Reggie Watts
XYZ gathers Followers
Plato returns to XYZ
XYZ accuses Thaksin Shinawatra
XYZ goes to Campbell College
XYZ succeeds Wenno
XYZ varietieses Manseng
XYZ departs Japan
Muse performs at XYZ
Björk grabs XYZ
Cofton Hackett works at XYZ
XYZ marries Frederick William, Elector of Brandenburg
Thornton Burgess broadcasts XYZ
XYZ is an university in Europe
XYZ is a way in Delhi
XYZ goes to Paris
Eva Luckes lives in XYZ
King attends XYZ
Gamba Osaka retains XYZ
Moffat contributes to XYZ
XYZ attends Sedbergh School
XYZ travels to Kyoto
Germania explores XYZ
XYZ engages Li Zitong
XYZ succeeds Chick Hearn
Clement Smyth is a bishop of XYZ
XYZ admonishes Luxo Jr.
Leinster defeats XYZ
Brown attacks XYZ
XYZ rises in Illinois
Hangangno-dong is a neighbourhood in XYZ
Clairefontaine produces XYZ
XYZ lies of Jihlava
Candice Night performs in XYZ
XYZ moves to New York

XYZ lies of Jihlava
XYZ resides England
XYZ describes Happy Accidents
The Federation is a representative from XYZ
XYZ rises in East Sussex
Back to Black is a seller in XYZ
Colin Montgomerie is a captain for XYZ
XYZ works for Scînteia
XYZ runs for Mayor of Chicago
Wedmore leaves XYZ
XYZ is a stadium in Chorley
Scarling. releases XYZ
XYZ writes Paper
Ashburne Hall is a hall on XYZ
Tooting Bec acquires XYZ
XYZ is a part of Brie
XYZ goes to Medina
XYZ dies in Moscow
Philadelphia Eagles drafts XYZ
XYZ draws New South Wales
XYZ diagrams West Virginia
XYZ is a character in Naked Lunch
XYZ is a municipality in Brazil
May Fortescue dies in XYZ
XYZ is a benefactor of Lapham Institute
Jarvis rejoins XYZ
Milton Shapp challenges XYZ
Amaranth reaches XYZ
Hungary rechambereds XYZ
California anchors in XYZ
Finland allies with XYZ
XYZ defeats Mike Kyle
Samuel Taylor Coleridge drifts from XYZ
XYZ reaches New York
Oracle Corporation develops XYZ
Diego serenades XYZ
United States of America attacks XYZ
Netherlands Antilles consists of XYZ
Pierce-Arrow carries XYZ
XYZ attends Texas High School
Australia sells XYZ
Greece competes in XYZ
XYZ moves to Venice
Sweden invades XYZ
XYZ is a band from England
Akalovo is a village in XYZ
XYZ joins Robert Borden
XYZ visits Havana
XYZ forms Pro Wrestling Noah
XYZ is a student of Bible
Apple Inc. joins XYZ
XYZ is a location of The Importance of Being Earnest
XYZ wins FA Cup
XYZ reserves Carl Monroe
XYZ is a concentration of Marist Brothers
Hillman goes to XYZ
XYZ wins GHC Tag Team Championship
Nathaniel Baldwin moves to XYZ
Anthony Lewis writes in XYZ
XYZ arrives at Virginia
XYZ stars Joel McCrea
Lewis travels to XYZ
XYZ is a replacement of Currie Cup
Errett Bishop teaches at XYZ
XYZ joins Crowded House
Poland forms XYZ
David Savan devoteds to XYZ
Buffet Crampon buys XYZ
XYZ serves Hong Kong Island
XYZ shares Nobel Prize
XYZ is a hero at Roush Fenway Racing
XYZ visits Japan
Son Ngoc Thanh escapes from XYZ

XYZ is a home to Military Academy
James D. Watson comes to XYZ
XYZ stops at Itami
XYZ strikes Union Army
XYZ serves at Inc.
McGehee studies at XYZ
XYZ moves to Greenwich Village
Jerry Kirkbride freelances in XYZ
XYZ departs Australia
XYZ describes Aristotle
XYZ returns to Nootka Sound
Two Horses of Genghis Khan lives in XYZ
Brown serves at XYZ
XYZ taps for WWE HEAT
Pontiac GTO promotes XYZ
Sternberg works in XYZ
Troop guards XYZ
Vicksburg Campaign importants to XYZ
XYZ lies of Třebíč
XYZ works with South Africa
XYZ defeats Don Allen
XYZ is a market for The Atlas
Herut: The National Movement departs from XYZ
Praz Bansi cashes in XYZ
Long Island is an extension of XYZ
XYZ annexs Oak Knoll
George H. Crosby Manitou State Park is a park on XYZ
XYZ includes Planet Hulk
XYZ attacks Shawn Michaels
XYZ replaces Niki Evans
XYZ studies in England
Lyndon B. Johnson goes to XYZ
XYZ walks on Moon
Writers includes XYZ
Osgoode returns to XYZ

Walter V. Shipley is a chairman of XYZ
XYZ tours Europe
XYZ is a school in Somalia
XYZ beats Steve Davis
XYZ includes Nora Andy Napaltjarri
XYZ travels to Paris
XYZ coaches at FC Winterthur
Chris Benoit chases XYZ
Syria is a member of XYZ
Doorways hints at XYZ
Kathleen Waldron becomes XYZ
Namco ports XYZ
XYZ onwards to Morocco
XYZ appears in Sex
XYZ works in Public Relations
Steve Bracks replaces XYZ
XYZ deprives Hannibal Barca
XYZ loses Staffordshire County Cricket Club
Sakuye adopts XYZ
Volkswagen Passenger Cars evolves into XYZ
XYZ replaces Adam McKay
Piyush Chawla replaces XYZ
XYZ moves into Silesia
XYZ investigates Seibal
XYZ jilts Gino Cervi
XYZ stars John Longden
Croatia extradites XYZ
XYZ beats Gomez
XYZ sails for California
XYZ joins in CSS Alabama
Wrexham Industrial Estate is a large in XYZ
XYZ serves on Trustee
XYZ moves to England
Paolo Sorrentino attends XYZ
Anacostia High School serves XYZ
XYZ begins Alejandro Pena
XYZ is a tributary in Romania

Graham Taylor manages XYZ
Borland starts in XYZ
XYZ levels at Rangers F.C.
XYZ resides in Cape Town
XYZ is a municipality of Piauí
XYZ goes on Heart of Midlothian F.C.
Tenedos falls to XYZ
Tan Zhongyi replaces XYZ
XYZ is a nazim of Abbottabad District
Reel Big Fish includes XYZ
XYZ bases American Airlines
XYZ preaches for Islam
Mahan sells XYZ
XYZ runs Cromer
XYZ becomes Vice President
William Hull surrenders XYZ
XYZ is a castle in Farnham
Russia influences XYZ
XYZ migrates into Byzantine Empire
XYZ blames Claudia Jordan
James Stewart enlists XYZ
Brian moves from XYZ
XYZ is a municipality in Brazil
XYZ is a peak in Bulgaria
B-45 Tornado is a bomber in XYZ
Shade Sheist features XYZ
Cove Rangers F.C. sells XYZ
XYZ allies with France
XYZ refers to Undertaker
XYZ leaves Roman Catholicism
The Trust arranges XYZ
Ku Klux Klan disperses from XYZ
Othello receives XYZ
David Amram meets XYZ
Key Tower rises on XYZ
XYZ speaks with Jason Pierce
Gideon returns XYZ

McCartney supplants XYZ
Allah prohibits XYZ
XYZ is a founder of Word of Life Church
XYZ studies Somerset
Tring stops at XYZ
XYZ consults Dinosaur
XYZ annexs Mobile District
XYZ loses in Wally Masur
Cherry co-createds XYZ
Camurus partners with XYZ
Matt Cameron attends XYZ
Nexcom Bulgaria LLC is an operator in XYZ
XYZ loses Division of Canberra
XYZ features John Entwistle
XYZ defeats Low Ki
XYZ relates to Typha
XYZ is a system in Canada
XYZ goes at Keystone Studios
Jesse James Leija loses to XYZ
Antonín Dvořák arrives in XYZ
XYZ leads Cleveland Cavaliers
XYZ bounds Ezzahra
XYZ regards South West Africa
XYZ beats Lancashire County Cricket Club
Kemak Language is a dialect of XYZ
Stephen F. Austin moves to XYZ
Luhden is a municipality in XYZ
Thomas Patrick Moore represents XYZ
XYZ arrives in United Kingdom
XYZ moves to Colorado
Pete Wilson becomes XYZ
XYZ returns to Van Nuys High School
XYZ withdraws from Lebanon
XYZ kills Pryderi
XYZ stops at Blue Mounds Fort
Oruk-Zar is a village of XYZ
Brian Shaw finds XYZ

Sokal releases XYZ
Redlight is a composer from XYZ
Bogner follows into XYZ
Susannah reports from XYZ
Peru competes in XYZ
Notts County F.C. returns to XYZ
XYZ rightbacks Nicky Hunt
Western Abenaki emigrates to XYZ
Colombia is an exporter in XYZ
Vardenis is a settlement in XYZ
Adolf Hitler rules XYZ
O'Donnell hits XYZ
XYZ is a stream from Ranchi
XYZ is a west of Mississippi
XYZ returns to Chicago
Pinheiro Machado is a municipality in XYZ
XYZ weakens Kentucky
Pontymoile Basin is a site to XYZ
XYZ is a school in United States of America
XYZ is a municipality in Schleswig-Holstein
XYZ employs Mates
XYZ conquers Russia
Bulgari works in XYZ
Norway qualifies from XYZ
XYZ becomes Prime Minister
Labor loses XYZ
XYZ runs for Connecticut
XYZ requires Viasat
XYZ defeats The Diamonds
Smith's Fort Plantation is a house of XYZ
Dattus looks to XYZ
XYZ stretches from Den Helder
Ormiscraig tens XYZ
Peach is a flavor in XYZ
XYZ returns to New York
Cicero undermines XYZ
XYZ populars in Darlington

XYZ terms Asif Ali Zardari
XYZ returns to Saint Petersburg
Archie Reynolds attends XYZ
XYZ replaces Dusty Baker
XYZ runs from Waiblingen
Charlie Earp Bridge is a bridge over XYZ
XYZ is a district of Cabo Delgado
XYZ relocates from Brooklyn
Danny Payne moves of XYZ
XYZ returns to Queens Park Rangers F.C.
XYZ arrives at New York City
Joseph Haines goes to XYZ
Rosa Parks exits XYZ
Venus Williams beats XYZ
XYZ feuds with David Bautista
Mars orbits XYZ
Viet Minh ups to XYZ
XYZ is a tributary in Romania
XYZ competes with Cees Paauwe
XYZ is a secretary of CDB
Brantley is a double in XYZ
XYZ partners Stan Smith
James Courtney moves to XYZ
XYZ chooses Grand Master
XYZ returns to Co-operative Championship
Christine Fernandes moves to XYZ
Germans travels to XYZ
XYZ withdraws from Tier
Chris Myers pairs with XYZ
Ethelbert of Kent meets XYZ
XYZ sees Samuel Beckett
XYZ screens at Melbourne Underground Film Festival
XYZ is a neighborhood in United States of America
Fräntorp belongs to XYZ
XYZ studies Margate

Cheyenne High School is a school in XYZ
XYZ appears in Toronto
XYZ includes Things We Said Today
XYZ announces with Indie Recordings
XYZ attends Pepperdine University
XYZ criticises Government of Pakistan
XYZ partners with NBC
Greece competes in XYZ
Railroad Tycoon II is a game for XYZ
XYZ backs Greg Urwin
Otto Vogl joins XYZ
XYZ visits Nepal
XYZ replaces Amanda Holden
XYZ leaves ABC Records
XYZ comes from Rothley
XYZ includes Edinburgh Gunners
Black joins XYZ
Schofield marches XYZ
Holy Roman Emperor unites XYZ
XYZ is a henge in Leagrave
Barbara Goldsmith becomes XYZ
XYZ defeats Killings
Gmina Zdzieszowice borders XYZ
Nate meets XYZ
XYZ parts from EMI
James A. King names XYZ
XYZ operates Brisbane
Oliver Reed assaults XYZ
XYZ leicesters in United Kingdom
Edward Canby defends XYZ
Eartha wins XYZ
Ronan Keating confirms XYZ
XYZ dominates Magahi Language
Olin occurs on XYZ
Bulldog defeats XYZ
Krumstedt is a municipality in XYZ
Milne returns to XYZ
XYZ visits Istanbul
USS Kinzer departs XYZ
XYZ remains under Lloyd D. George
XYZ defeats Ferreira
Lakshmi is a resettle in XYZ
Piribebuy River ends at XYZ
Tufanganj femaleses XYZ
Benny Andersson submits XYZ
Dissidenten tours XYZ
XYZ archives Department
Y-O-U asks XYZ
Malli worships XYZ
XYZ wins Drama Desk Award
XYZ moves to Melbourne
XYZ headlines Take Action Tour
XYZ records Could I Have This Kiss Forever
Ho Yeow Sun represents XYZ
XYZ progresses to Finals
Chris Jericho unmasks XYZ
Tyler Saint occupies XYZ
Mersin covers XYZ
Brett Steven loses to XYZ
XYZ fights with George Washington
Areas includes XYZ
XYZ marrieds to Latvians
XYZ rescues Semih Kaya
XYZ represents Japan
XYZ leaves London
Brent Weedman fights XYZ
XYZ features Ken's Labyrinth
XYZ creates Green Mountain Coffee Roasters
XYZ is a municipality in Slovakia
Gann is a pilot for XYZ
Connecticut lives in XYZ
XYZ loses to Johnny Curtis
XYZ introduces Japan
Jabez Bryce invests XYZ

Hunter Douglas expands into XYZ
XYZ creates Graham Goddard
XYZ obsesses with Fanny Pelopaja
Lake Macleod is a lake in XYZ
XYZ requests Masahiro Sakurai
XYZ retires from Sarah Lawrence College
XYZ is an actress from England
XYZ proposes HOPE
New York climbs XYZ
Nickelodeon partners with XYZ
Taliban Movement flees XYZ
XYZ moves to Prudential Center
XYZ delists from NASDAQ
Little Fyodor is a musician from XYZ
Li Cunxu aids XYZ
XYZ is a tributary in Romania
XYZ defeats University of Virginia
XYZ borders Haryana
Bjørgulv Braanen succeeds XYZ
Dinamo Riga signs XYZ
XYZ settles in Tushino
XYZ releases Amused to Death
Rose Kelly represents XYZ
Don Dunstan builds XYZ
Třebelovice lies of XYZ
XYZ is a graduate of Air War College
XYZ defeats Syuri
XYZ speaks English Language
Tivi is a municipality in XYZ
Madhur Bhandarkar re-approacheds XYZ
XYZ is a suburb of Australia
Gavin returns from XYZ
XYZ arrives in Guantánamo
XYZ is a car from United Kingdom
Daniel defeats XYZ
Lheebroek resides in XYZ
Port of Yingkou is a seaport in XYZ
XYZ stars Peter Davison
Ksawerów is a village in XYZ
XYZ goes to Massachusetts
XYZ buys Paper
Cookie Mueller writes XYZ
Davey Allison plows into XYZ
XYZ joins Iris Associates
XYZ qualifies for NCAA Men's Ice Hockey Championship
Michael Crozier deafeatings XYZ
XYZ teaches English Language
Andrew W.K. provides XYZ
XYZ replaces Psycho Clown
XYZ is a broadcast on NBC
XYZ is a partner with Professor
Rivers Guthrie attends XYZ
XYZ flamboyants in Newsday
Joseph Haydn arrives in XYZ
Clement Attlee becomes XYZ
Adam Smith publishes XYZ
England assigns to XYZ
XYZ dramatizes The Murder of Roger Ackroyd
Leddra Chapman releases XYZ
Powderfinger tours XYZ
XYZ grows Stange
XYZ immigrates Ontario
XYZ worships God
XYZ is an ostler in British English
XYZ moves to Los Angeles
Grampian is a region of XYZ
United States Agency for International Development assists XYZ
XYZ forms Rodinia
The Truth About Youth is a drama from XYZ
White returns to XYZ
Laughlin moves to XYZ

XYZ moves to CNN
Irving Allen directs XYZ
XYZ enrols at Harvard University
Ruben Bemelmans replaces XYZ
XYZ joins Titanium
XYZ goes to Stonyhurst Saint Mary's Hall
Charles Gordone returns to XYZ
Geoff Mack goes with XYZ
XYZ joins Pet Shop Boys
Texas Battle stars in XYZ
XYZ regards Soviet Union
XYZ runs The Hollywood Reporter
XYZ commissions Li Shenfu
Marie Webster lives in XYZ
XYZ describes Tovik
XYZ joins Janata Dal
XYZ is a member of Fier
Fyodor Dostoyevsky works on XYZ
Houston dies at XYZ
Nicholas I of Russia visits XYZ
Cassiodorus writes XYZ
Don Luce trades to XYZ
Canada becomes XYZ
XYZ dependents on Treneglos
Moses reminds XYZ
Roman Empire adopts XYZ
Rantířov lies on XYZ
Syama Sastri hails XYZ
Richard M. Elliot serves at XYZ
Morgul signs to XYZ
Nu Aurigae is a light-year from XYZ
River Tyne is a river in XYZ
Special Criminal Investigation publishes XYZ
Amherstview Jets becomes XYZ
Arthur Lismer immigrates XYZ
XYZ steps in Yushin Okami
Hadley Richardson travels to XYZ
XYZ leaves for Fox Kids
XYZ houses in Florence
Dan Wood creates XYZ
XYZ moves from Birmingham
XYZ results in Kid Knievel
EMI releases XYZ
XYZ occupies Kengtong
Malaysia vses XYZ
Luce Lopez-Baralt sees XYZ
XYZ meets Andrew Breitbart
Masjid Al-Iman is a mosque in XYZ
MIR is a member of XYZ
XYZ releases Elantris
XYZ spawns The Waltons
XYZ throws Gatorade
Peter Thiel supports XYZ
XYZ provides CNN
Dimondale is a village in XYZ
Ante Gotovina returns to XYZ
Andre Williams releases XYZ
Supreme Court of Canada rules of XYZ
Kuryer Polski refers to XYZ
XYZ accredits Turpin High School
XYZ is a brother-in-law of Hadrian
XYZ wins Award Software
XYZ waits for Y-O-U
XYZ returns to England
XYZ comprises Bernard Sumner
Italy enters XYZ
XYZ connects to Nishinomiya-Kitaguchi Station
XYZ wins at The Olympic Club
XYZ loyals to Gallienus
Neale coaches XYZ
XYZ votes for Daniel D. Tompkins
Motnău River is a tributary in XYZ
XYZ is a professor at Columbia Law School

XYZ defeats Pyle
XYZ remarks Venus
Bruce Campbell serves on XYZ
Leverett DeVeber attends XYZ
XYZ links Bristol
Joan Rivers works with XYZ
XYZ invades England
XYZ buys Bottle Rack
Bret Harte moves to XYZ
Rajesh Khanna tutors XYZ
Wigan Warriors meets XYZ
XYZ is a son of Burgate
Jove Francisco is a journalist from XYZ
Thangal Kunju Musaliar is an author of XYZ
XYZ recognizes North Korea
M.o.v.e comes XYZ
Buckshot Roberts kills XYZ
XYZ visits Europe
XYZ collaborates with Chesney Hawkes
XYZ wins Maria João Koehler
Grant Morrison writes XYZ
XYZ runs Anstruther
William J. Byron distinguishes XYZ
White leaves XYZ
XYZ beats Royal Engineers A.F.C.
Iltutmish circles XYZ
Valea Mare River is a tributary in XYZ
Parnitha relies on XYZ
New York Jets places XYZ
XYZ releases In Search of Solid Ground
XYZ becomes President
XYZ is a tree in England
XYZ returns to Germany
XYZ continues with Jeremy Roenick
Malachi is a prophet of XYZ
XYZ heads to Michigan
XYZ creates Timbuktu

Liberia completes XYZ
Oslo is a city in XYZ
GameSpy adds XYZ
Mary Robinson visits XYZ
XYZ replaces Darrell Nulisch
XYZ studies at Makerere University
Jacques Goddet succeeds XYZ
XYZ files Los Angeles Police Department
Finland joins XYZ
XYZ visits Venice
United States of America enters XYZ
XYZ studies with Ralph Shapey
XYZ rules Germany
Dogen refers to XYZ
Paul Friedmann publishes XYZ
XYZ ends with Restless Farewell
XYZ returns to Leipzig
Pérez teams for XYZ
XYZ goes to Paris
Gachantivá is a municipality in XYZ
XYZ divides Earth
XYZ serves in Las Vegas
James M. Swift attends XYZ
XYZ continues Babylon
XYZ becomes Chief Executive Officer
God charges XYZ
Getawarayo stars XYZ
XYZ is a critic of Israel
XYZ stars Dana Andrews
Bălţi invades XYZ
XYZ travels to California
The Palace is a complex in XYZ
XYZ attacks Republic of Venice
United States Navy provides XYZ
Coe Booth graduates in XYZ
Henry IV retains XYZ
XYZ regains Victor McLaglen

XYZ travels to England
XYZ situates Gloucester
XYZ appears in Domesday Book
XYZ distributes The Golf Channel
Paul Bryant Bridge absorbs XYZ
XYZ splits into Eurasia
XYZ performs in Moscow
XYZ defeats The Godwinns
XYZ resides in Vienna
XYZ works with Yoko Ono
Blumenthal, Schleswig-Holstein accuses XYZ
Ellery Hanley involves XYZ
XYZ pens I'll Never Break Your Heart
Basil II repulses XYZ
XYZ attends Michigan
NK Engines Company succeeds XYZ
XYZ stands against A.D. Patel
President arrives at XYZ
XYZ rides Comanche
XYZ purchases Blue Poles
XYZ travels to Australia
XYZ appears in FA Cup Final
XYZ leads Watford F.C.
XYZ is an airport in Mohave County
Norway follows XYZ
XYZ lives in Norfolk
XYZ federates with Barstable School
Moon works in XYZ
XYZ wins at Huddersfield Town F.C.
Chris Wragge replaces XYZ
XYZ stars Nikolaj Lie Kaas
Marwan spies for XYZ
Coupling is a broadcast on XYZ
XYZ is a district in England
Roos falls with XYZ
Möngke Khan returns to XYZ

8.3 French Questions used in Chapter 5

This section contains the French question set used in Chapter 5. Answers (in English) were annotated by a native English speaker who self-assessed as being fluent in French.

Tarentule est un espèce de XYZ
Scott Steiner défie XYZ
XYZ compte Serbes
Larzac est un réacteur de XYZ
XYZ emmène Syracuse
XYZ est un commune de Territoire de Belfort
XYZ est un membre d'Eurorégion
Tina Turner reçoit XYZ
XYZ est un officier de Schutzstaffel
Vandales envahit XYZ
XYZ sort Game Boy Micro
Scott Steiner bat XYZ
XYZ quitte Londres
Colette de Corbie rencontre à XYZ
Finlande remporte XYZ
XYZ lance Game Boy Advance SP
Michel Marie Claparède chasse XYZ
XYZ est un membre de Commission
Ulamburiash est un roi de XYZ
Tosawi est un chef de XYZ
Afrasiab reçoit XYZ
Australie gagne XYZ

XYZ est un album de Heart
An Fheothanach est un village de XYZ
XYZ part à Rome
Gmina Kościelec est un commune de XYZ
XYZ est un ville de Suède
Oberroth est un commune de XYZ
XYZ repart à Londres
Charlotte Casiraghi est un fille de XYZ
XYZ est un fils de Dionysos
Jerry Lawler pousse XYZ
Cao Cao envoie XYZ
XYZ trouve Pétrus Borel
XYZ est un commune d'Indre-et-Loire
Jonny Storm challenge XYZ
XYZ passe par Albert II de Belgique
XYZ défie Michael Coulthard
La Dernière Femme est un film de XYZ
Pritul̆any est un village de XYZ
Joan Baez part pour XYZ
XYZ est un inventeur d'Alfred Bird
XYZ publie Porcie
XYZ est un fils de Marcus Livius Salinator
XYZ est un ville de Suède
XYZ accueille Intel
Alexandros Papanastasiou soutient XYZ
Gmina Kraszewice est un commune de XYZ
Madeleine Castaing est un amie de XYZ
XYZ occupe Malacca Town
XYZ vit à Saint-Germain-en-Laye
Sebastian Vettel double XYZ
Lady Catherine Grey visite XYZ
Henri II de Rohan défend XYZ
Namangi Aute est un mouvement de XYZ
XYZ gouverne Syrie
Allen Dulles est un numéro de XYZ
Chelsea Football Club est un champion de XYZ
XYZ est un ville de Pays-Bas
XYZ est un officier de Cao Cao
XYZ est un prévôt de Douai
Chrétienté célèbre XYZ
Norvège commande XYZ
XYZ est un ville de Yémen
XYZ entre dans Milan
XYZ allie avec France
Antsiranana est un province de XYZ
XYZ est un disciple de Martin Heidegger
XYZ est un album de Jay Brannan
XYZ occupe Tchécoslovaquie
Cara Black représente XYZ
XYZ expulse Juifs
XYZ est un patinoire de Winnipeg Jets
XYZ est un village de Tasmanie
XYZ retire sur San Miguel de Tucumán
Hizan est un district de XYZ
XYZ vit à Frohnau
XYZ est un ville d'Alaska
XYZ est un commune de Haïti
XYZ nie La Cité de Dieu
The Undertaker bat XYZ
Invertigo est un copie de XYZ
XYZ cite Wilhelm Röpke
Acacia est un roman de XYZ
XYZ est un ville de Pologne
The Four Tops est un quartet de XYZ
Pedra Badejo est un localité de XYZ
XYZ porte RKO Pictures
XYZ est un clone de Roxy Theater
Roxy Theater est un clone de XYZ
Georgenberg est un commune de XYZ
XYZ est un volcan de Russie
XYZ est un artiste d'Israël
Buse inspire XYZ
XYZ est un commune de Voïvodie de Grande-

Pologne
Francesco Cairo part à XYZ
XYZ remplace Tully Blanchard
Béez est un rivière de XYZ
XYZ rejoint Rome
XYZ revient sur Officine Meccaniche
Seveso est un rivière de XYZ
XYZ est un album de The Bee Gees
Barnkanalen est un chaîne de XYZ
XYZ regagne Russie
XYZ découvre Thébé
Hamilcar Barca investit XYZ
Benoît XV nomme XYZ
Ryan Reeves regagne XYZ
XYZ écarte Meaux
Graham Parker quitte XYZ
Racing Club de France Football affronte XYZ
Francisco Mancebo gagne XYZ
XYZ est un film de Douglas Sirk
XYZ est un numéro de Central Intelligence Agency
XYZ est un étang de Pyrénées
Pavel Pabst est un ami de XYZ
XYZ invite Jean Monnet
Capitaine Blood est un roman de XYZ
Rim-Sin I est un roi de XYZ
Gino Paoli persuade XYZ
XYZ est un ville de Saxe
XYZ nomme Machaon
Zduny est un ville de XYZ
XYZ est un village de Bosnie-Herzégovine
XYZ remporte European Table Tennis Union
Microïds publie XYZ
XYZ est un groupe de Saxe
XYZ est un rue de Londres
Alexander Creek est un communauté de XYZ
XYZ assiège Perpignan
XYZ est un ville de Pologne
Maine-et-Loire situe à XYZ
XYZ joue avec Gibson Guitar Corporation
XYZ est un père de Satsuki
John Bolling est un petit-fils de XYZ
Les Colocs remporte XYZ
XYZ accompagne Oliver Hardy
XYZ gagne Prix de Diane
Owen Hart bat XYZ
XYZ est un commune de Pas-de-Calais
Francs annexe XYZ
XYZ est un commune de Voïvodie de Grande-Pologne
XYZ est un préfecture de Bas-Rhin
XYZ est un localité d'Alaska
XYZ est un cité de Kent
XYZ ramène Nankin
Lantern gagne XYZ
Simeria est un ville de XYZ
XYZ vend Atari
Charles Rogier quitte XYZ
Sega sort XYZ
Numérien atteint XYZ
Birmanie perd XYZ
XYZ attaque Ségeste
XYZ est un capitale d'Australie
XYZ reconquiert Angleterre
Catalans ravage XYZ
Scafell Pike est un sommet de XYZ
Michel de Montaigne est un précurseur de XYZ
XYZ devance Alonso
Canton Charge transfère XYZ
Royaume-Uni détache XYZ
Azerbaïdjan envoie XYZ
Chinzei est un nom de XYZ
Winchell est un ami de XYZ

Finlande achète XYZ
Ditzingen est un ville de XYZ
XYZ aime Labé
XYZ bat Roumanie
Eventful est un single de XYZ
XYZ est un localité de Sénégal
Gmina Duszniki est un commune de XYZ
XYZ est un commune de Terre de Feu
XYZ est un femme de Mao Zedong
XYZ est un espèce d'Amphibia
Villefort est un commune de XYZ
Evonne Goolagong bat XYZ
XYZ vit à Vis-en-Artois
Ryan Peake joue sur XYZ
XYZ est un pseudonyme de Per Yngve Ohlin
XYZ est un espèce d'Urodèle
Serenade gagne XYZ
XYZ bat Victoria Azarenka
Daniel Iffla dit XYZ
Michael Matthews adjuge XYZ
Chinese Stripe-necked Turtle est un espèce de XYZ
XYZ quitte Bauhaus
Auguste Frédéric Louis Viesse de Marmont abandonne XYZ
XYZ habite Paris
Robertsport est un ville de XYZ
Phraortès est un roi de XYZ
XYZ est un ville de Liberia
Manhattan Valley est un quartier de XYZ
Preciosa est un surnom de XYZ
Historia de Gentibus Septentrionalibus est un œuvre de XYZ
China Europe International Business School existe à XYZ
XYZ traverse Océan Atlantique
XYZ est un capitale de Pas-de-Calais
Laye est un ville de XYZ
XYZ est un l'édition de Wikipédia
Stanislas Skalski obtient XYZ
Mahomet est un descendant de XYZ
Montluçon est un h de XYZ
Catch dit XYZ
XYZ est un fils de Mathieu de Foix-Castelbon
Bensonville est un ville de XYZ
South African Airlink rejoint XYZ
Sanniquellie est un ville de XYZ
XYZ est un fils de César de Vendôme
Diego Forlán remporte XYZ
Andrée Putman crée XYZ
Sony Ericsson XPERIA X10 est un incursion de XYZ
Libye accuse XYZ
XYZ est un ville d'Alaska
KAA La Gantoise accueille XYZ
Léonora Dori est un confident de XYZ
XYZ est un ville de Sreten Stojanović
XYZ est un commune de Bade-Wurtemberg
Jasenov est un village de XYZ
Majapahit attaque XYZ
XYZ part de Goa
XYZ est un comédie de Roger Donaldson
Samuel Taylor Coleridge rencontre XYZ
Socrate encourage XYZ
XYZ gagne Aria
XYZ cite Audovera
Normands pille XYZ
XYZ quitte Uruguay
Patricia Rozema considère XYZ
XYZ détache Birmanie
XYZ est un ville d'Alberta
Milagro emmène XYZ
XYZ est un ville de Philippines
XYZ est un ville de Népal

Français remporte sur XYZ
XYZ relègue Andy Schleck
South African Air Force investit XYZ
XYZ est un supergroupe de Limp Bizkit
Chen est un ancêtre de XYZ
Christophe Colomb perd XYZ
XYZ est un témoin de Breton
Calamia gagne XYZ
Kalwaria Zebrzydowska est un ville de XYZ
Masiliwa Snout-burrower est un espèce de XYZ
XYZ dit Jacques Feyder
XYZ enregistre Magic Night
XYZ quitte La Haye
Andromaque est un tragédie de XYZ
Saint-Michel-des-Saints est un municipalité de XYZ
XYZ quitte HAL Laboratory
XYZ est un espèce de Serpentes
XYZ est un roman de Georges Simenon
XYZ dit Coluche
Francisco Pizarro quitte XYZ
XYZ est un espèce de Sauria
Audi repose sur XYZ
XYZ est un subdivision de Birmanie
XYZ insulte Insane Clown Posse
XYZ est un single de Depeche Mode
Gaulois fond XYZ
XYZ quitte Pickfair
Altman réalise XYZ
XYZ envahit Pologne
Samarra est un ville de XYZ
Gutenzell-Hürbel est un commune de XYZ
Gérald Passi est un frère de XYZ
XYZ est un fils de Sven II de Danemark
Robert Trujillo quitte XYZ
XYZ obtient Belgrade
XYZ quitte Damiette
XYZ vit dans Viêt Nam
XYZ est un roi de Bhoutan
Coquimbo est un ville de XYZ
XYZ est un fils de Robert Francis Kennedy
XYZ est un général de British Army
Roumains côtoie XYZ
XYZ charge Jean-Henri Fabre
Zubné est un village de XYZ
Roumains côtoie XYZ
Minor Swing est un composition de XYZ
XYZ situe sur Pouancé
Craig Quinnell quitte XYZ
XYZ est un quartier de Rodez
XYZ bat Lindsay Davenport
XYZ retire Matt Holliday
XYZ chasse Matveï Platov
Vasily Petrenko enregistre XYZ
XYZ est un ville de Burkina Faso
Roumains côtoie XYZ
Marty Friedman accueille XYZ
Esther Dale joue à XYZ
XYZ remplace Leone
Machów, Lublin Voivodeship est un village de XYZ
Jordanów est un ville de XYZ
André Luis Garcia dit XYZ
Bentiu est un ville de XYZ
Vijfheerenlanden est un région de XYZ
Cristina Fernández de Kirchner soutient XYZ
Madura va de XYZ
Ted Parsons vit à XYZ
XYZ bat Saint Louis Athletica
XYZ est un village de Neerijnen
XYZ est un condottiere de Pesaro
XYZ perd Hulk Hogan
XYZ acquiert Ping.fm

Punta Perrucchetti est un sommet de XYZ
XYZ est un ville d'Israël
Marcus Loew achète XYZ
Death Dealer est un peinture de XYZ
Maggie Mae est un chanson de XYZ
Wülfrath est un ville de XYZ
Wake Up Dead Man est un chanson de XYZ
XYZ est un commune de Bavière
Umberto Eco mentionne XYZ
Roumains côtoie XYZ
Wigéric de Bidgau accueille XYZ
XYZ est un fois de Pologne
XYZ est un espèce d'Amphibia
Hollandais occupe XYZ
Mérovingiens nomme XYZ
XYZ est un commune de Savoie
XYZ bat Kurt Angle
XYZ dit Big Bill Broonzy
XYZ est un ville de Kirghizistan
XYZ gagne Prix de Diane
XYZ est un nom de Tivoli
Démocrates veut XYZ
XYZ est un rivière de Sibérie
XYZ côtoie Roumains
Charles W. Bartlett quitte XYZ
XYZ est un membre de Club de Budapest
XYZ guide Windows Communication Foundation
Jean-Baptiste Nicolas Roch de Ramezay est un fils de XYZ
Tepoztecatl est un frère de XYZ
XYZ envahit Israël
Wheat Kings de Brandon est un club de XYZ
XYZ est un voix de Harvey Keitel
XYZ rejoint Russie
XYZ est un fils de Giacomo Attendolo
Kool Herc appelle XYZ
XYZ est un village de Colombie-Britannique
Keremeos est un village de XYZ
XYZ bat Helen Gourlay
XYZ sort Stone Cold Sober
XYZ est un village de Serbie
LVG C.VI est un amélioration de XYZ
Mszana Dolna est un ville de XYZ
Brecon est un ville de XYZ
XYZ reçoit Modibo Keïta
Glenn Whelan rejoint XYZ
Xanten est un ville de XYZ
John Petrucci inaugure XYZ
XYZ est un commune de Slovénie
Sonnaz regroupe XYZ
Ville de Shoalhaven quitte XYZ
Toulouse découvre XYZ
Saint ressemble par XYZ
XYZ bombarde Kaboul
XYZ remporte Brixia Tour
Hans-Georg Gadamer est un disciple de XYZ
Jesús Fernández Sáenz dit XYZ
XYZ remporte Anémie de Fanconi
XYZ invite Michael Hutchence
XYZ tue Arabes
Mikhaïl Gorbatchev reçoit XYZ
Libertarias est un film de XYZ
XYZ envahit Hollande
Carleton-sur-Mer est un ville de XYZ
Gengis Khan occupe XYZ
XYZ est un album de Dalida
XYZ porte RKO Pictures
Chiefs de Johnstown est un franchise de XYZ
Jules César défend XYZ
Alojzy Ehrlich représente XYZ
XYZ produit Internationalist
XYZ adjuge Tour de Castille-et-León
Civita Castellana est un cité de XYZ

Saint-Chély-d'Aubrac absorbe XYZ
XYZ accompagne Jay Farrar
XYZ est un volcan d'Islande
XYZ bat Fergal Devitt
Gornja Trepča est un village de XYZ
Olimpia Milan retrouve XYZ
XYZ occupe Kobryn
Königsplatz est un place de XYZ
Warburg est un ville de XYZ
XYZ est un ville d'Allemagne
Christian Bale joue XYZ
Sammy Hagar est un chanteur de XYZ
Eagles de Philadelphie affronte XYZ
XYZ est un province de Japon
XYZ est un chanson d'Alice Cooper
Kay Khusraw Ier assiège XYZ
Florent III de Hollande accompagne XYZ
Mike Tyson bat XYZ
Roumains côtoie XYZ
Liliane Bettencourt est un femme de XYZ
XYZ vit à Thionville
Fenerbahçe SK accueille XYZ
XYZ bat Rosie Casals
Alicia, Bohol est un municipalité de XYZ
XYZ est un wali de Pampelune
We Want Miles est un album de XYZ
Ryan Vogelsong joue avec XYZ
Monica Seles bat XYZ
XYZ est un ami d'Owney Madden
XYZ est un point de Jamaïque
Totila reprend XYZ
XYZ dit Kenny Washington
XYZ fonde Nouvelle-Amsterdam
XYZ connaît Wilhelm Furtwängler
Soleil réchauffe XYZ
Lutèce devient XYZ
XYZ bat Lesley Turner
Paris est un capitale de XYZ
XYZ est un h d'Orléans
XYZ occupe Balkh
XYZ franchit Rhin
XYZ invite Guerrilla War
Nokia E70 est un successeur de XYZ
Chelsea Football Club recrute XYZ
XYZ nomme Yoshihiko Noda
XYZ sort Hunky Dory
XYZ est un ville de Comté de Moira
Praia est un ville de XYZ
XYZ est un fille de Nigel Lawson
XYZ est un roi de Babylone
Burna-Buriash est un roi de XYZ
Joseph Simmons est un frère de XYZ
Cologne menace XYZ
Long Island est un île de XYZ
Joe R. Lansdale vit à XYZ
Juffureh est un ville de XYZ
XYZ engendre Pontos
XYZ vit dans Connecticut
Thiodina force XYZ
XYZ est un sœur de Modoald de Trèves
Arabella Steinbacher joue XYZ
XYZ dit Ivan IV de Russie
XYZ est un épouse de Christian VIII de Danemark
XYZ quitte Dublin
Saintes est un chef-lieu de XYZ
XYZ joue contre Fluminense Football Club
Raeapteek est un pharmacie de XYZ
XYZ part dans Bornéo
Tahiti domine XYZ
Paris ramène XYZ
The Stranger Next Door est un roman de XYZ
XYZ envoie Alcibiade
XYZ est un sophiste d'Athènes

Angleterre passe par XYZ
XYZ est un ville de Michigan
Meyenburg est un ville de XYZ
XYZ devance Jenson Button
XYZ est un ville d'Argentine
XYZ nomme Greg Byrne
XYZ est un fils d'Oscar Aguirregaray
RMS Mauretania quitte XYZ
XYZ affronte Syracuse
Fun House est un album de XYZ
XYZ tourne Ernest Hemingway
XYZ annexe Damaraland
XYZ est un ville de Gilan
Suns de Phoenix appuie sur XYZ
XYZ transfère Darvin Ham
Hué est un ville de XYZ
XYZ est un île de Philippines
XYZ est un dialogue de Platon
XYZ bat Ann Haydon Jones
XYZ voit Afrique
Eslarn est un commune de XYZ
Phillies de Philadelphie rapatrie XYZ
XYZ remplace Henton
XYZ aide Hayao Miyazaki
XYZ est un parodie de Wikipédia
Asian Dub Foundation est un composante de XYZ
XYZ nomme Conrad II de Bavière
XYZ accueille Coupe Memorial
XYZ rejoint Machen
XYZ est un municipalité de Benguet
XYZ est un capitale de Province de Kibuye
XYZ absorbe Cité
XYZ quitte Marinus de Tyr
Catherine de Médicis regagne XYZ
Polanes forme XYZ
Jean-Hugues Ateba rejoint XYZ
XYZ dépasse Hughes H-4 Hercules
XYZ est un commune de Bavière
XYZ est un sommet d'Iran
Ratko Svilar rejoint XYZ
Alix de Vergy est un duchesse de XYZ
Esham quitte XYZ
XYZ remporte Tour de Suisse
XYZ envoie Luis
Royaume-Uni annexe XYZ
Canale est un canal de XYZ
Ferguson Jenkins prononce sur XYZ
Steve Corino attaque XYZ
Philadelphia Independence bat XYZ
XYZ occupe Landshut
XYZ embarque Pékin
Wehrmacht entre dans XYZ
Central Intelligence Agency aide XYZ
Piémont soulève contre XYZ
XYZ affronte Sale Sharks
Sunny est un reprise de XYZ
XYZ réalise sur Jean Renoir
Bavarois est un habitant de XYZ
Byumba est un capitale de XYZ
XYZ est un surnom d'Ahmed Abdallah Sambi
XYZ propose Star Trek
XYZ devance Cadel Evans
Don Escudero représente XYZ
Orénoque est un fleuve de XYZ
XYZ est un époux de Hathor
XYZ bat Suisse
XYZ vit à Summerside
XYZ connecte Téhéran
XYZ est un fan de Depeche Mode
Mr. Natural est un album de XYZ
Barnabé règne sur XYZ
XYZ est un 12single de Mami Kawada
XYZ est un roi de France

Lion de Belfort est un sculpture de XYZ
Joan Baez rencontre XYZ
Tarnówka est un commune de XYZ
BOMA est un acronyme de XYZ
Malchin est un ville de XYZ
XYZ est un cimetière de Dresde
XYZ est un ville de Pologne
Juan Francisco García dit XYZ
Some Hearts est un album de XYZ
XYZ incarne James Bond
Seamus Heaney quitte XYZ
Pretoria évince XYZ
XYZ dit Andre Luis
XYZ fonde Allemagne
Kraiburg est un commune de XYZ
XYZ rencontre Etienne Martin
Printemps de Prague conduit XYZ
XYZ bat Eagles de Philadelphie
XYZ reçoit Siyavash
Bigwig est un groupe de XYZ
Wojnicz est un ville de XYZ
Steve Borden attaque XYZ
Ségolène Royal rencontre XYZ
XYZ pousse Werner Best
XYZ saccage Abarkuh
Michel Serrault accueille XYZ
Football Club de Nantes quitte XYZ
XYZ est un commune de Bavière
Niederbergkirchen est un commune de XYZ
Bratislava est un nom de XYZ
XYZ gagne Prix de Diane
XYZ quitte Addis-Abeba
XYZ cède Holstein
Thayetmyo est un ville de XYZ
Justine Henin bat XYZ
MTS Centre est un patinoire de XYZ
XYZ remporte Tour de Suisse
XYZ devance Sarah Hendrickson
Rattenkirchen est un commune de XYZ
XYZ part à Berlin
XYZ joue à Dendre
XYZ quitte Belgique
XYZ est un playoffs de The Women
Rokstarr est un album de XYZ
Schwindegg est un commune de XYZ
Easier Said Than Done est un composition de XYZ
XYZ est un commune de Voïvodie de Grande-Pologne
XYZ est un ami de John Milton
Alt Urgell est un comarque de XYZ
XYZ est un ville de Rhénanie-du-Nord-Westphalie
Shawn Hernandez bat XYZ
XYZ est un ville de Bavière
XYZ rejoint Imerys
XYZ quitte King Oliver
XYZ est un urbaine-rurale de Voïvodie de Grande-Pologne
Spartak Saint-Pétersbourg est un club de XYZ
XYZ est un défenseur de Dion de Syracuse
XYZ est un président d'International Business Machines
Bugojno est un centre de XYZ
Louis IV de Germanie appuie sur XYZ
Selena désigne XYZ
Papín est un village de XYZ
Holzheim, Neu-Ulm est un commune de XYZ
Johannes Kepler quitte XYZ
Clay Shaw attaque XYZ
XYZ est un phare de Stonebridge Press
Bien Unido est un municipalité de XYZ
Manhattan est un comédie de XYZ
XYZ vit avec Hidatsas

Ainaži est un ville de XYZ
XYZ envoie Muqali
XYZ appuie Nikita Khrouchtchev
Cambridge remporte XYZ
Dinklage obtient XYZ
XYZ dote Singapour
XYZ est un ville de Pologne
Gmina Miedzichowo est un commune de XYZ
Závadka est un village de XYZ
Grafenwöhr est un ville de XYZ
XYZ invite Satan
XYZ est un enfant de Biafra
Phil Jackson replace XYZ
XYZ forme Mondo Generator
XYZ est un sommet d'Alpes
XYZ bombarde Liège
XYZ est un commune de Bavière
XYZ est un espèce d'Amphibia
XYZ dit Hugo Grotius
Schrobenhausen est un commune de XYZ
XYZ est un commune de Bavière
Darius Ier est un vainqueur de XYZ
XYZ est un commune de Bavière
XYZ envoie Scott Speed
XYZ déménage de Birmingham
Herbert Hoover vit dans XYZ
Aristoxène accuse XYZ
XYZ rachète Club Méditerranée
Evonne Goolagong bat XYZ
XYZ devance Kamui Kobayashi
Massachusetts General Hospital est un hôpital de XYZ
XYZ occupe Qingdao
XYZ joue avec François-René Duchâble
Jean-Paul II cite XYZ
XYZ est un gratte-ciel de New York
XYZ remporte Bataille de Mylae

Bezymianny est un volcan de XYZ
XYZ est un membre de Hanse
XYZ est un capitale de Roumanie
Mottola retrouve XYZ
Alcmond de Derby écrit XYZ
Aresing est un commune de XYZ
XYZ traverse Gaule
Françoise Dolto rencontre XYZ
XYZ côtoie Hongrois
XYZ est un groupe de Rhénanie-Palatinat
XYZ identifie Cicogni
XYZ devance Marcel Hirscher
XYZ rejoint Indiana
Benito Mussolini rejoint XYZ
Pascal Lissouba obtient XYZ
XYZ reçoit Naples
XYZ contacte Emil Hácha
Autriche envahit XYZ
Italiens occupe XYZ
RMS Carpathia atteint XYZ
Stadtbergen est un ville de XYZ
Grande-Bretagne rend XYZ
Fontenay-Mauvoisin est un commune de XYZ
Halliday quitte XYZ
XYZ est un descendant de Nobunaga Oda
XYZ bat Arantxa Sánchez Vicario
XYZ convainc Paulist Fathers
XYZ envoie Hermès
XYZ est un département de Niger
Clint Eastwood engage XYZ
XYZ part avec Thaïlande
Congressional Plaza est un place de XYZ
Cork bat XYZ
Charlie Chaplin rencontre XYZ
Claris sort XYZ
XYZ bat Roumanie
Sonoma est un municipalité de XYZ

XYZ est un genre de Phoque
Eurorégion organise XYZ
Dirkou est un commune de XYZ
XYZ dit Vivant Denon
XYZ est un roi de Larsa
Ostroróg est un urbaine-rurale de XYZ
XYZ est un espèce de Sauria
Roger Taylor est un membre de XYZ
XYZ est un membre de Crips
Hugues est un habitant de XYZ
XYZ joue à Diegem
Staley enregistre XYZ
XYZ est un île d'Espagne
XYZ est un commune de Bavière
Lukačovce est un village de XYZ
Manco Cápac est un frère de XYZ
XYZ rejoint FK Partizan Belgrade
XYZ quitte France
XYZ est un ville de Chili
XYZ soumet Kirghizes
XYZ domine Mer Méditerranée
Ric Flair perd XYZ
Diamond Heights est un quartier de XYZ
XYZ est un ville de Suède
Roger Miller épouse XYZ
XYZ est un commune de Voïvodie de Grande-Pologne
Hernán Rengifo rejoint XYZ
Angra joue sur XYZ
Gougoush rencontre XYZ
XYZ est un commune de Bavière
Störnstein est un commune de XYZ
XYZ est un ville de Kent
Sugenheim est un commune de XYZ
XYZ est un successeur de Nokia E70
Hollandais installe à XYZ
XYZ est un ville d'Albanie
XYZ possède Taba International Airport
Louis Chevrolet fonde XYZ
Markt Taschendorf est un commune de XYZ
Ségolène Royal recueille XYZ
Comets de Houston appuie sur XYZ
XYZ est un constructeur de Renault
XYZ défend Alfred Kerr
XYZ est un film de Martin McDonagh
XYZ est un pièce de Richard Strauss
Beaubassin-Est est un membre de XYZ
Suédois envahit XYZ
XYZ découvre Adrastée
XYZ remplace Lysandre
XYZ élimine Angleterre
XYZ quitte The Hollies
XYZ est un commune de Bavière
XYZ décrit Homme de Néandertal
XYZ recrute Mau Maus
XYZ quitte Vickers
FK Alania Vladikavkaz prive XYZ
XYZ est un entraîneur d'Ajax Amsterdam
Jerry West recrute XYZ
Woody Allen confie XYZ
XYZ propose Piranha
Kent Cooper est un directeur de XYZ
XYZ rejoint Milan
XYZ compte Nonza
XYZ est un salle de Hongrie
Caramelos de Cianuro est un groupe de XYZ
XYZ veut Amérique
XYZ déclare GNU
XYZ recouvre Milet
XYZ est un famille d'Acari
XYZ cite Régis Debray
XYZ est un prince de Salm-Kyrburg
XYZ atteint Jérusalem
XYZ est un gratte-ciel de Hong Kong

XYZ est un leader de MSC Croisières
Mikhaïl Mikhaïlovitch Speranski invite XYZ
Axel Braun est un fils de XYZ
Prussiens assiège XYZ
Heini Hediger est un directeur de XYZ
XYZ rencontre Anatole Demidoff
XYZ est un rivière de Russie
Ludmila Javorová vit à XYZ
Dong Zhuo soumet XYZ
XYZ dit Randy Williams
XYZ réintègre Château de Prague
Serena Williams bat XYZ
Marcos Aoás Corrêa dit XYZ
Hobart quitte XYZ
Afternoon apparaît sur XYZ
Ransart est un commune de XYZ
Castanheira quitte XYZ
XYZ est un vainqueur d'Aristagoras
XYZ est un ville de Mecklembourg-Poméranie-Occidentale
XYZ affronte Rangers de New York
Le Solitaire est un long-métrage de XYZ
Conrad Ferdinand Meyer écrit XYZ
XYZ rencontre Francisco Franco
XYZ décide Commission
Syracuse détruit XYZ
XYZ achète Metro-Goldwyn-Mayer
Oaxaca Mud Turtle est un espèce de XYZ
Christopher Tolkien vit en XYZ
XYZ est un commune de Voïvodie de Grande-Pologne
XYZ envoie Chen Deng
Emmaüs est un hospitalité de XYZ
XYZ est un poète de Pléiade
XYZ occupe Vilnius
Ladislas abandonne XYZ
XYZ bat Akiba Rubinstein

Doris Hart bat XYZ
XYZ lance Club Jenna
Mori Yoshinari bat contre XYZ
Osogbo est un ville de XYZ
XYZ obtient Dornier
XYZ bat Comté de Laois
XYZ est un fils de Johny Schleck
XYZ accuse Hans Küng
Bragi Boddason est un scalde de XYZ
XYZ quitte Shinouï
Gisenyi est un ville de XYZ
Clérette est un rivière de XYZ
Royaume-Uni annexe XYZ
XYZ est un drôle de Schutzstaffel
XYZ domine Vanuatu
Qanûn est un œuvre de XYZ
XYZ obtient Ottawa
Causapscal est un ville de XYZ
XYZ bat Jack Bobridge
Kirwan est un amie de XYZ
XYZ est un ville de Québec
Roger Miller naît XYZ
XYZ marche sur Lahore
Royale Union Saint-Gilloise représente XYZ
XYZ situe à Saint-Cyr-en-Bourg
XYZ est un village de Slovaquie
Kevin Nash affronte XYZ
Charles Quint concède XYZ
Cyborg est un film de XYZ
Wihtred de Kent laisse XYZ
Napoléon Ier quitte XYZ
XYZ rachète Charisma Records
Trinquetaille est un quartier de XYZ
XYZ est un groupe d'Allemagne
Verchen est un commune de XYZ
Marlik est un archéologie de XYZ
XYZ est un ville d'Allemagne

XYZ annonce DVD
XYZ épouse Frédéric-Guillaume II de Prusse
Kan River est un rivière de XYZ
XYZ rejoint Extreme Championship Wrestling
XYZ bat Mary Pierce
Flávio Sérgio Viana revient sur XYZ
XYZ bat Cork
XYZ vit en France
Canterbury Rugby Football Union remporte XYZ
Assyrie soumet XYZ
Pygmy Salamander est un espèce de XYZ
Geislingen est un ville de XYZ
Tokyo Dome bat XYZ
Turquie conserve XYZ
XYZ vit à Maranello
Plourin-lès-Morlaix situe à XYZ
République de Gênes usurpe XYZ
XYZ est un neveu de Robert Guiscard
XYZ remplace Devon Aoki
The Penguins élimine XYZ
XYZ est un sous-ordre de Squamata
XYZ est un volcan de Chili
XYZ quitte Florence
Pruillé situe à XYZ
XYZ quitte Londres
Benito Mussolini occupe XYZ
XYZ crée Questar
Abakan est un rivière de XYZ
Jules II place XYZ
XYZ signifie AT&T
Persée de Macédoine quitte XYZ
Ohře est un nom de XYZ
XYZ gagne Kentucky Derby
Gmina Ladek est un commune de XYZ
Intensive Care est un album de XYZ
Egra est un nom de XYZ
Sokan Yamazaki quitte XYZ
Cotton Mather contacte XYZ
Beaubassin-Est est un municipalité de XYZ
Alexandre de Wurtemberg est un duc de XYZ
AT&T signifie XYZ
White-faced Tree Rat est un espèce de XYZ
Jean-Jacques Pauvert édite XYZ
XYZ est un port d'Indonésie
Sand Hill Road est un route de XYZ
Westre est un commune de XYZ
Lehrte est un ville de XYZ
Lescure désigne XYZ
XYZ va Ténérife
Rachel part à XYZ
XYZ conquiert Hollywood
XYZ charge Lü Bu
Bobby Bazini est un auteur-compositeur-interprète de XYZ
Nelson de Jesus Silva rejoint XYZ
XYZ rejoint Thibaut Pinot
Bruno Senna remplace XYZ
Anna Leonowens part avec XYZ
XYZ est un municipalité de Québec
XYZ assiste Pierre Mendès France
Koškovce est un village de XYZ
XYZ rencontre Françoise Arnoul
Amélie Mauresmo bat XYZ
Konrad Adenauer est un chancelier de XYZ
Le Gitan est un film de XYZ
Chiriqui Pocket Gopher est un espèce de XYZ
XYZ est un fils d'Owen Tudor
XYZ est un ville de Malaisie
ReinXeed est un groupe de XYZ
Wehrmacht occupe XYZ
XYZ traverse Belgique
XYZ est un claviériste d'Ozzy Osbourne

XYZ crée Studio
XYZ part pour Londres
Boucher lance XYZ
XYZ fonde Reuters
Selayar est un île de XYZ
XYZ est un dame de Liban
XYZ ignore Séville
Métastase joue contre XYZ
Alberto Contador remporte XYZ
Voghiera est un hameau de XYZ
XYZ manque Venezuela
XYZ est un ville de Pologne
XYZ ravage Cilicie
XYZ est un commune d'Allemagne
New York bat XYZ
Bopfingen est un ville de XYZ
Micky Moody rejoint XYZ
Britton Hill est un point de XYZ
XYZ est un ville de Maroc
XYZ bat Lakers de Los Angeles
XYZ est un localité d'Alaska
XYZ est un titre de Tampa Red
Dreux reçoit XYZ
XYZ invite Paris
XYZ occupe Chios
Villerupt possède XYZ
XYZ appelle La Farlède
XYZ quitte Gengis Khan
Hannibal Barca attire XYZ
XYZ est un propriété de Discovery Communications
Johann Turmair dit XYZ
XYZ est un propriétaire de Discovery Channel
XYZ est un lac d'Alaska
Shou envoie XYZ
XYZ bat Zina Garrison
XYZ bat Margaret Smith Court
Chaguaramas est un localité de XYZ
Göllheim est un commune de XYZ
XYZ est un branche d'IJ
XYZ cite Steve Vai
Agnetha Fältskog rencontre XYZ
XYZ dit Sergio Paulo Barbosa Valente
XYZ assiège Byzance
XYZ quitte Inter Milan
Hernán Crespo quitte XYZ
Lydia Aran est un spécialiste de XYZ
Giovanni Antonio Canal dit XYZ
Albert Kirchner destitue XYZ
XYZ dit Pisano
XYZ quitte Apple Computer
XYZ entre dans Tunis
Français devance XYZ
XYZ rappelle Hamren
XYZ quitte Ajax Amsterdam
XYZ est un ville de Rhénanie-du-Nord-Westphalie
XYZ est un département de Niger
XYZ écrit sur Allmusic
Glentoran Football Club bat XYZ
Solmania désigne XYZ
XYZ lance Macintosh LC
XYZ est un espèce d'Amphibia
XYZ vit à Konya
Melbourne est un ville de XYZ
Trinity Church est un bâtiment de XYZ
Samurai Math Beats est un album de XYZ
XYZ est un fils de Djötchi
Gigi Fernández bat XYZ
XYZ est un municipalité de Basilan
XYZ est un époux de Satis
René Desmaison quitte XYZ
Vallespir est un région de XYZ

Michel de Montaigne évoque XYZ
Ernest Shackleton considère XYZ
Victor Serge espère XYZ
XYZ envahit Béotie
XYZ devance Felipe Massa
XYZ est un ville de Guinée-Bissau
XYZ représente Univers
Chams occupe XYZ
XYZ est un rivière de Serbie
XYZ invite Koshi Inaba
XYZ joue contre Shahar Peer
Québec compte XYZ
XYZ remporte Coupe Davis
Elvis Presley dit XYZ
Ustyurt Plateau est un plateau de XYZ
Almucs de Castelnou est un noble de XYZ
XYZ achète Activision
Stephen Frears connaît XYZ
Alphonse Esquiros trouve XYZ
XYZ bat Kim Clijsters
XYZ est un ville de Malte
Guanahani débarque XYZ
XYZ survole Paris
XYZ est un parc de Séville
Judith Gautier prévient XYZ
XYZ traduit par Matthieu Chedid
Stephen P. Synnott découvre XYZ
Arthur Miller regagne XYZ
XYZ subit Pakistan International Airlines
XYZ est un rivière de Belgique
XYZ rejoint Bayonne
XYZ différencie par Rheinmetall
Jean II Casimir Vasa conserve XYZ
Bhola est un île de XYZ
XYZ remplace Lino Ventura
Tamura invite XYZ
Chepo est un ville de XYZ
Abdul Rachid Dostom bombarde XYZ
XYZ rejoint Cardinals de Saint-Louis
Blackbelly Salamander est un espèce de XYZ
XYZ est un village de Bosnie-Herzégovine
XYZ choisit Angerfist
Steamboat Willie est un dessin de XYZ
XYZ crée Ulster Democratic Party
Sainte-Clotilde-de-Beauce, Quebec est un municipalité de XYZ
XYZ reprend Montrésor
XYZ joue sur Canal+
Didier Deschamps déclare XYZ

Bibliography

O. Abend, S. B. Cohen, and M. Steedman. Lexical inference over multi-word predicates: A distributional approach. In *Proceedings of ACL*, 2014.

Kisuh Ahn, Beatrix Alex, Johan Bos, Tiphaine Dalmas, Jochen L Leidner, and Matthew B Smillie. Cross-lingual question answering with QED. In *Working Notes, CLEF Cross-Language Evaluation Forum*, pages 335–342, 2004.

David Aldous. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII—1983*, pages 1–198, 1985.

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. Using ccg categories to improve hindi dependency parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 604–609, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P13-2107`.

C.F. Baker, C.J. Fillmore, and J.B. Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational*

Linguistics (ACL'05), pages 597–604, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P05-1074.

M. Baroni, R. Bernardi, N.Q. Do, and C. Shan. Entailment above the word level in distributional semantics. In *Proceedings of EACL*, pages 23–32. Citeseer, 2012.

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. Frege in space: A program for compositional distributional semantics. *Submitted, draft at http://clic. cimec. unitn. it/composes*, 2013.

V. Basile, J. Bos, K. Evang, and N. Venhuizen. Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). To appear*, 2012.

Stephen Beale, Sergei Nirenburg, and Kavi Mahesh. Semantic analysis in the Mikrokosmos machine translation project. In *Proceedings of the 2nd Symposium on Natural Language Processing*, pages 297–307, 1995.

Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. Montague meets markov: Deep semantics with probabilistic logical form, 2013.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of focused entailment graphs. In *Proceedings of ACL*, Uppsala, Sweden, 2010.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 610–619. Association for Computational Linguistics, 2011. ISBN 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002550.

Jonathan Berant, Ido Dagan, Meni Adler, and Jacob Goldberger. Efficient tree-based approximation for entailment graph learning. In *Proceedings of ACL*, 2012.

Michel Berkelaar, Kjell Eikland, and Peter Notebaert. lp_solve 5.5, open source (mixed-integer) linear programming system. Software, May 1 2004. URL `http://lpsolve.sourceforge.net/5.5/`. Available at ¡http://lpsolve.sourceforge.net/5.5/¿. Last accessed Dec, 18 2009.

C. Biemann. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80. Association for Computational Linguistics, 2006.

Alexandra Birch, Phil Blunsom, and Miles Osborne. A Quantitative Analysis of Reordering Phenomena. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 197–205, Athens, Greece, March 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W09-0434`.

William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics, 2012.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

D. G. Bobrow, C. Condoravdi, R. Crouch, V. De Paiva, L. Karttunen, T. H. King, R. Nairn, L. Price, and A. Zaenen. Precision-focused textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 16–21. Association for Computational Linguistics, 2007. URL `http://dl.acm.org/citation.cfm?id=1654536.1654541`.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-102-6. doi: 10.1145/1376616.1376746. URL `http://doi.acm.org/10.1145/1376616.1376746`.

J. Bos and K. Markert. Recognising textual entailment with logical inference. In *Proceedings of the conference on Human Language Technology and Empirical Methods* *in Natural Language Processing*, pages 628–635. Association for Computational Linguistics, 2005. URL `http://acl.ldc.upenn.edu/H/H05/H05-1079.pdf`.

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

Johan Bos. Is there a place for logic in recognizing textual entailment? *Linguistic Issues in Language Technology*, 9, 2013.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. Converting a Dependency Treebank to a Categorial Grammar Treebank for Italian. In M. Passarotti, Adam Przepiórkowski, S. Raynaud, and Frank Van Eynde, editors, *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT8)*, pages 27–38, Milan, Italy, 2009.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311, June 1993. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=972470.972474`.

P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, and J.C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

Sabine Buchholz and Erwin Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics, 2006.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. Improved statistical machine translation using paraphrases. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 17–24, New York City, USA, June 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N06/N06-1003`.

Marie Candito, Benoît Crabbé, Pascal Denis, et al. Statistical french dependency parsing: treebank conversion and first results. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 1840–1847, 2010.

Matthew Capetola. Towards universal quantification in distributional semantic space. In *Joint Symposium on Semantic Processing.*, page 75, 2013.

S. Clark, M. Steedman, and J.R. Curran. Object-extraction and question-parsing using CCG. In *Proceedings of the EMNLP Conference*, pages 111–118, 2004.

Stephen Clark and James R. Curran. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04. Association for Computational Linguistics, 2004. doi: http://dx.doi.org/10.3115/1218955.1218969. URL `http://dx.doi.org/10.3115/1218955.1218969`.

David Clausen and Christopher D. Manning. Presupposed content and entailments in natural language inference, 2009.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis: A Festschrift for Joachim Lambek*, 36(1-4):345–384, 2010.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

Robin Cooper. *Quantification and syntactic theory*, volume 21. D. Reidel Publishing Company, 1983.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. Using the framework. *FraCaS Deliverable D*, 16, 1996.

Robin Hayes Cooper. Montague's semantic theory and transformational syntax. 1975.

R. Crouch. Packed rewriting for mapping semantics to kr. In *Proceedings of the International Workshop on Computational Semantics*. Citeseer, 2005.

Ido Dagan, O. Glickman, and B. Magnini. The PASCAL recognising textual entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, 2006.

D.A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Proceedings of the workshop on Human Language Technology*, pages 43–48. Association for Computational Linguistics, 1994.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. Frame-semantic parsing. 2013.

Hal Daumé III. Fast search for dirichlet process mixture models. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AIStats)*, San Juan, Puerto Rico, 2007. URL `http://hal3.name/docs/#daume07astar-dp`.

D. Davidson. 6. the logical form of action sentences. *Essays on actions and events*, 1 (9):105–149, 1967.

Pascal Denis, Benoît Sagot, et al. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *PACLIC*, pages 110–119, 2009.

Pedro Domingos and William Austin Webb. A tractable first-order probabilistic logic. In *AAAI*, 2012.

Bonnie J Dorr, Eduard H Hovy, and Lori S Levin. Machine translation: Interlingual methods. 2004.

Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka. Cross-language latent relational search: Mapping knowledge across languages. *Association for the Advancement of Artificial Intelligence*, pages 1237–1242, 2011.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545. Association for Computational Linguistics, 2011. ISBN 978-1-937284-11-4. URL `http://dl.acm.org/citation.cfm?id=2145432.2145596`.

D. Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28, 2000.

T. Fountain and M. Lapata. Incremental models of natural language category acquisition. In *Proceedings of the 32st Annual Conference of the Cognitive Science Society*, 2011.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

D. Garrette, K. Erk, and R. Mooney. Integrating logical representations with probabilistic information using markov logic. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 105–114. Association for Computational Linguistics, 2011.

Peter T Geach. A program for syntax. In *Semantics of Natural Language*, pages 483–497. Springer, 1973.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 1–9. Association for Computational Linguistics, 2007. URL `http://dl.acm.org/citation.cfm?id=1654536.1654538`.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

Yoav Goldberg and Jon Orwant. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 1, pages 241–247, 2013.

D. Graff, J. Kong, K. Chen, and K. Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 2003.

Edward Grefenstette. Towards a formal distributional semantics: Simulating logical calculi with tensors. *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, 2013. URL `http://www.cs.ox.ac.uk/people/edward.grefenstette/GrefenstetteTFDS.pdf`.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. Concrete sentence spaces for compositional distributional models of meaning. *Computational Semantics IWCS 2011*, page 125, 2011.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL http://doi.acm.org/10.1145/1656274.1656278.

ZS Harris. Distributional structure. word 10: 146-162. reprinted in j. fodor and j. katz. *The structure of language: Readings in the philosophy of language*, pages 775–794, 1954.

Marti A Hearst and Hinrich Schütze. Customizing a lexicon to better suit a computational task. In *Proc. of the Workshop on Extracting Lexical Knowledge*. Citeseer, 1996.

Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. " not not bad" is not" bad": A distributional account of negation. In *Proceedings of the 2013 Workshop on Continuous Vector Space Models and their Compositionality*. Proceedings of the 2013 Workshop on Continuous Vector Space Models and their Compositionality, 2013.

Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. Recognizing textual entailment with lcc's groundhog system. In *Proceedings of the Second PASCAL Challenges Workshop*, pages 137–142, 2006.

Julia Hockenmaier. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 505–512, Sydney, Australia, 2006. doi: 10.3115/1220175.1220239. URL http://dx.doi.org/10.3115/1220175.1220239.

Julia Hockenmaier and Mark Steedman. CCGbank: a corpus of CCG derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396, 2007.

M. Honnibal, J.R. Curran, and J. Bos. Rebanking CCGbank for improved NP interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 207–215. Association for Computational Linguistics, 2010. URL http://acl.eldoc.ub.rug.nl/mirror/P/P10/P10-1022.pdf.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics, 2006.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

Bevan Jones, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, and Kevin Knight. Semantics-based machine translation with hyperedge replacement grammars. *Proc. COLING, 2012*, 2012.

L. Karttunen. *The Logic of English Predicate Complement Constructions*. Linguistics Club Bloomington, Ind: IU Linguistics Club. Indiana University Linguistics Club, 1971. URL `http://books.google.co.uk/books?id=BuRCAAAAIAAJ`.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182, 2003.

P. Kingsbury and M. Palmer. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1989–1993. Citeseer, 2002.

Alexandre Klementiev, Ann Irvine, Chris Callison-Burch, and David Yarowsky. Toward statistical machine translation without parallel corpora. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 130–140. Association for Computational Linguistics, 2012a. ISBN 978-1-937284-19-0. URL `http://dl.acm.org/citation.cfm?id=2380816.2380835`.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing crosslingual distributed representations of words. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, Bombay, India, December 2012b.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, 2005.

Philipp Koehn and Christof Monz. Manual and automatic evaluation of machine translation between European languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City, June 2006. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180. Association for Computational Linguistics, 2007. URL `http://dl.acm.org/citation.cfm?id=1557769.1557821`.

Alexander Koller and Stefan Thater. An improved redundancy elimination algorithm for underspecified representations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 409–416. Association for Computational Linguistics, 2006. doi: http://dx.doi.org/10.3115/1220175.1220227. URL `http://dx.doi.org/10.3115/1220175.1220227`.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-geffet. Directional distributional similarity for lexical inference. *Nat. Lang. Eng.*, 16(4):359–389, October 2010. ISSN 1351-3249. doi: 10.1017/S1351324910000124. URL `http://dx.doi.org/10.1017/S1351324910000124`.

Mikhail Kozhevnikov and Ivan Titov. Crosslingual transfer of semantic role models. In *To Appear in Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Saul A Kripke. Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96, 1963.

Iddo Lev, Bill MacCartney, Christopher D. Manning, and Roger Levy. Solving logic puzzles: from robust processing to precise semantics. In *Proceedings of the 2nd*

Workshop on Text Meaning and Interpretation, TextMean '04, pages 9–16. Association for Computational Linguistics, 2004. URL http://dl.acm.org/citation.cfm?id=1628275.1628277.

Mike Lewis and Mark Steedman. Combined Distributional and Logical Semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192, 2013a.

Mike Lewis and Mark Steedman. Unsupervised induction of cross-lingual semantic relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 681–692, Seattle, Washington, USA, October 2013b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1064.

Mike Lewis and Mark Steedman. Combining formal and distributional models of temporal and intensional semantics. *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, June 2014a. URL http://homepages.inf.ed.ac.uk/s1049478/publications/sp2014.pdf.

Mike Lewis and Mark Steedman. Improved CCG parsing with Semi-supervised Supertagging. *Transactions of the Association for Computational Linguistics*, 2014b.

Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584. ACM, 2006.

Dekang Lin and Patrick Pantel. DIRT - Discovery of Inference Rules from Text. In *In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328, 2001.

Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*, 2012.

Amnon Lotan, Asher Stern, and Ido Dagan. Truthteller: Annotating predicate truth. In *Proceedings of the 2013 Conference of the North American Chapter of the ACL*, pages 752–757, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N13-1091.

Bill MacCartney and Christopher D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphras-*

ing, RTE '07, pages 193–200. Association for Computational Linguistics, 2007. URL `http://dl.acm.org/citation.cfm?id=1654536.1654575`.

Bill MacCartney and Christopher D Manning. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 521–528. Association for Computational Linguistics, 2008.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2): 313–330, 1993.

J.T. Maxwell and R. Kaplan. A method for disjunctive constraint satisfaction. *Formal Issues in Lexical-Functional Grammar*, pages 381–401, 1995.

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

W. McCune. Prover9 and Mace4. `http://cs.unm.edu/~mccune/mace4/`, 2005.

Scott McDonald. Environmental determinants of lexical processing effort. 2000.

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas W Oard, and David Doermann. Cross-language entity linking. *Proc. IJCNLP2011*, 2011.

Oren Melamud, Jonathan Berant, Ido Dagan, Jacob Goldberger, and Idan Szpektor. A two level model for context sensitive inference rules. In *Proceedings of ACL*, 2013.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The nombank project: An interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*, pages 24–31, 2004.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b.

G.A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38 (11):39–41, 1995.

David Mimno, Hanna M Wallach, Jason Naradowsky, David A Smith, and Andrew McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 880–889. Association for Computational Linguistics, 2009.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

Teruko Mitamura, Eric H Nyberg, and Jaime G Carbonell. An efficient interlingua translation system for multi-lingual document production. 1991.

J. Mitchell and M. Lapata. Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, pages 236–244, 2008.

Marc Moens and Mark Steedman. Temporal ontology and temporal reference. *Computational linguistics*, 14(2):15–28, 1988.

Richard Montague. The proper treatment of quantification in ordinary english. In *Approaches to natural language*, pages 221–242. Springer, 1973.

Roser Morante and Eduardo Blanco. * sem 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 265–274. Association for Computational Linguistics, 2012.

Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

R.M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Dominick Ng and James R Curran. Dependency hashing for n-best ccg parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 497–505. Association for Computational Linguistics, 2012.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2): 95–135, 2007.

D.O. O'Seaghdha. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 435–444. Association for Computational Linguistics, 2010.

Sebastian Padó and Mirella Lapata. Constructing semantic space models from parsed corpora. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 128–135, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075113. URL `http://dx.doi.org/10.3115/1075096.1075113`.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard H Hovy. Isp: Learning inferential selectional preferences. In *HLT-NAACL*, pages 564–571, 2007.

Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 1–10. Association for Computational Linguistics, 2009. ISBN 978-1-932432-59-6. URL `http://dl.acm.org/citation.cfm?id=1699510.1699512`.

Hoifung Poon and Pedro Domingos. Unsupervised ontology induction from text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 296–305. Association for Computational Linguistics, 2010. URL `http://dl.acm.org/citation.cfm?id=1858681.1858712`.

Reinhard Rapp. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 519–526. Association for Computational Linguistics, 1999. ISBN 1-55860-609-3. doi: 10.3115/1034678.1034756. URL `http://dx.doi.org/10.3115/1034678.1034756`.

Philip Resnik and Noah A Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380, 2003.

M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1): 107–136, 2006.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL-HLT*, pages 74–84, 2013.

A. Ritter, O. Etzioni, et al. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434. Association for Computational Linguistics, 2010.

Aju Thalappillil Scaria, Jonathan Berant, Mengqiu Wang, Peter Clark, Justin Lewis, Brittany Harding, and Christopher D. Manning. Learning biological processes with global constraints. In *Proceedings of EMNLP*, 2013.

Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1088–1098. Association for Computational Linguistics, 2010. URL `http://dl.acm.org/citation.cfm?id=1870658.1870764`.

R. Socher, B. Huval, C.D. Manning, and A.Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*. 2013.

Prakash Srinivasan and Alexander Yates. Quantifier scope disambiguation using extracted pragmatic knowledge: Preliminary results. In *Proceedings of the 2009 Con-*

ference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, pages 1465–1474. Association for Computational Linguistics, 2009.

Mark Steedman. *The Syntactic Process*. MIT Press, 2000.

Mark Steedman. *Taking Scope: The Natural Semantics of Quantifiers*. MIT Press, 2012.

Mark Steedman and Jason Baldridge. Combinatory categorial grammar. *Non-Transformational Syntax Oxford: Blackwell*, pages 181–224, 2011.

Idan Szpektor, Eyal Shnarch, and Ido Dagan. Instance-based evaluation of entailment rule acquisition. In *In Proceedings of ACL 2007*, volume 45, page 456, 2007.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 477–487. Association for Computational Linguistics, 2012. ISBN 978-1-937284-20-6. URL `http://dl.acm.org/citation.cfm?id=2382029.2382096`.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

Emily Thomforde and Mark Steedman. Semi-supervised ccg lexicon extension. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1246–1256. Association for Computational Linguistics, 2011.

Ivan Titov and Alexandre Klementiev. A bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1445–1455, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1145`.

Stephen Tratz and Eduard Hovy. A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 678–687. Association for Computational Linguistics, 2010.

Daniel Tse and James R. Curran. Chinese CCGbank: extracting CCG derivations from the Penn Chinese Treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1083–1091, Beijing, China, 2010. URL http://dl.acm.org/citation.cfm?id=1873781.1873903.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

Peter D. Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 1136–1141, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1642293.1642475.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. Integrating multiple dependency corpora for inducing wide-coverage japanese ccg resources. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1042–1051, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-1103.

Bernard Vauquois. A survey of formal grammars and algorithms for recognition and transformation in machine translation. In *IFIP Congress*, volume 68, pages 254–260, 1968.

Julie Weeds and David Weir. A general framework for distributional similarity. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 81–88. Association for Computational Linguistics, 2003.

Hila Weisman, Jonathan Berant, Idan Szpektor, and Ido Dagan. Learning verb inference rules from linguistically-motivated evidence. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 194–204. Association for Computational Linguistics, 2012. URL http://dl.acm.org/citation.cfm?id=2390948.2390972.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Em-*

pirical Methods in Natural Language Processing, EMNLP '11, pages 1456–1466. Association for Computational Linguistics, 2011. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145587.

Limin Yao, Sebastian Riedel, and Andrew McCallum. Unsupervised relation discovery with sense disambiguation. In *ACL (1)*, pages 712–720, 2012.

David Yarowsky. One sense per collocation. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 266–271. Association for Computational Linguistics, 1993. ISBN 1-55860-324-7. doi: 10.3115/1075671.1075731. URL http://dx.doi.org/10.3115/1075671.1075731.

Naomi Zeichner, Jonathan Berant, and Ido Dagan. Crowdsourcing inference-rule evaluation. In *Proceedings of ACL (short papers)*, 2012.

J.M. Zelle and R.J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1050–1055, 1996.

Congle Zhang and Daniel S Weld. Harvesting parallel news streams to generate paraphrases of event relations.