Technical Report

UCAM-CL-TR-735
ISSN 1476-2986

Number 735

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Learning compound noun semantics

Diarmuid Ó Séaghdha

December 2008

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted July 2008 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Corpus Christi College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Learning compound noun semantics

Diarmuid Ó Séaghdha

Summary

This thesis investigates computational approaches for analysing the semantic relations in compound nouns and other noun-noun constructions. Compound nouns in particular have received a great deal of attention in recent years due to the challenges they pose for natural language processing systems. One reason for this is that the semantic relation between the constituents of a compound is not explicitly expressed and must be retrieved from other sources of linguistic and world knowledge.

I present a new scheme for the semantic annotation of compounds, describing in detail the motivation for the scheme and the development process. This scheme is applied to create an annotated dataset for use in compound interpretation experiments. The results of a dual-annotator experiment indicate that good agreement can be obtained with this scheme relative to previously reported results and also provide insights into the challenging nature of the annotation task.

I describe two corpus-driven paradigms for comparing pairs of nouns: lexical similarity and relational similarity. Lexical similarity is based on comparing each constituent of a noun pair to the corresponding constituent of another pair. Relational similarity is based on comparing the contexts in which both constituents of a noun pair occur together with the corresponding contexts of another pair. Using the flexible framework of kernel methods, I develop techniques for implementing both similarity paradigms.

A standard approach to lexical similarity represents words by their co-occurrence distributions. I describe a family of kernel functions that are designed for the classification of probability distributions. The appropriateness of these distributional kernels for semantic tasks is suggested by their close connection to proven measures of distributional lexical similarity. I demonstrate the effectiveness of the lexical similarity model by applying it to two classification tasks: compound noun interpretation and the 2007 SemEval task on classifying semantic relations between nominals.

To implement relational similarity I use kernels on strings and sets of strings. I show that distributional set kernels based on a multinomial probability model can be computed many times more efficiently than previously proposed kernels, while still achieving equal or better performance. Relational similarity does not perform as well as lexical similarity in my experiments. However, combining the two models brings an improvement over either model alone and achieves state-of-the-art results on both the compound noun and SemEval Task 4 datasets.

Acknowledgments

The past four years have been a never-boring mixture of learning and relearning, experiments that worked in the end, experiments that didn't, stress, relief, more stress, some croquet and an awful lot of running. And at the end a thesis was produced. A good number of people have contributed to getting me there, by offering advice and feedback, by answering questions and by providing me with hard-to-find papers; these include Tim Baldwin, John Beavers, Barry Devereux, Roxana Girju, Anders Søgaard, Simone Teufel, Peter Turney and Andreas Vlachos. Andreas also read and commented on parts of this thesis, and deserves to be thanked a second time. Julie Bazin read the whole thing, and many times saved me from my carelessness. Diane Nicholls enthusiastically and diligently participated in the annotation experiments. My supervisor Ann Copestake has been an unfailing source of support, from the beginning – by accepting an erstwhile Sanskritist who barely knew what a probability distribution was for a vague project involving compound nouns and machine learning – to the end – she has influenced many aspects of my writing and presentation for the better. I am grateful to my PhD examiners, Anna Korhonen and Diana McCarthy, for their constructive comments and good-humoured criticism.

This thesis is dedicated to my parents for giving me a good head start, to Julie for sticking by me through good results and bad, and to Cormac, for whom research was just one thing that ended too soon.

Contents

1 Introduction **11**

2 Compound semantics in linguistics and NLP **15**

2.1 Introduction 15

2.2 Linguistic perspectives on compounds 15

2.2.1 Compounding as a linguistic phenomenon 15

2.2.2 Inventories, integrated structures and pro-verbs: A survey of representational theories 16

2.3 Compounds and semantic relations in NLP 21

2.3.1 Inventories, integrated structures and pro-verbs (again) 21

2.3.2 Inventory approaches 22

2.3.3 Integrational approaches 24

2.4 Conclusion 25

3 Developing a relational annotation scheme **27**

3.1 Introduction 27

3.2 Desiderata for a semantic annotation scheme 27

3.3 Development procedure 30

3.4 The annotation scheme 32

3.4.1 Overview 32

3.4.2 General principles 33

3.4.3 BE 35

3.4.4 HAVE 35

3.4.5 IN 36

3.4.6 ACTOR, INST 36

3.4.7 ABOUT 38

3.4.8 REL, LEX, UNKNOWN 38

3.4.9 MISTAG, NONCOMPOUND 39

3.5 Conclusion 39

4 Evaluating the annotation scheme **41**
4.1 Introduction . . . 41
4.2 Data . . . 41
4.3 Procedure . . . 44
4.4 Analysis . . . 45
4.4.1 Agreement . . . 45
4.4.2 Causes of disagreement . . . 46
4.5 Prior work and discussion . . . 51
4.6 Conclusion . . . 53

5 Semantic similarity and kernel methods **55**
5.1 Introduction . . . 55
5.2 A similarity-based approach to relational classification . . . 55
5.3 Methods for computing noun pair similarity . . . 56
5.3.1 Constituent lexical similarity . . . 57
5.3.1.1 Lexical similarity paradigms . . . 57
5.3.1.2 The distributional model . . . 59
5.3.1.3 Measures of similarity and distance . . . 62
5.3.1.4 From words to word pairs . . . 67
5.3.2 Relational similarity . . . 68
5.3.2.1 The relational distributional hypothesis . . . 68
5.3.2.2 Methods for token-level relational similarity . . . 70
5.3.2.3 Methods for type-level relational similarity . . . 71
5.4 Kernel methods and support vector machines . . . 74
5.4.1 Kernels . . . 75
5.4.2 Classification with support vector machines . . . 77
5.4.3 Combining heterogeneous information for classification . . . 81
5.5 Conclusion . . . 82

6 Learning with co-occurrence vectors **83**
6.1 Introduction . . . 83
6.2 Distributional kernels for semantic similarity . . . 83
6.3 Datasets . . . 86
6.3.1 1,443 Compounds . . . 86
6.3.2 SemEval Task 4 . . . 87
6.4 Co-occurrence corpora . . . 88

6.4.1 British National Corpus . . . 89
6.4.2 Web 1T 5-Gram Corpus . . . 90
6.5 Methodology . . . 91
6.6 Compound noun experiments . . . 93
6.7 SemEval Task 4 experiments . . . 97
6.8 Further analysis . . . 100
6.8.1 Investigating the behaviour of distributional kernels . . . 100
6.8.2 Experiments with co-occurrence weighting . . . 102
6.9 Conclusion . . . 104

7 Learning with strings and sets **107**
7.1 Introduction . . . 107
7.2 String kernels for token-level relational similarity . . . 107
7.2.1 Kernels on strings . . . 107
7.2.2 Distributional string kernels . . . 109
7.2.3 Application to SemEval Task 4 . . . 110
7.2.3.1 Method . . . 110
7.2.3.2 Results . . . 111
7.3 Set kernels for type-level relational similarity . . . 113
7.3.1 Kernels on sets . . . 113
7.3.2 Multinomial set kernels . . . 115
7.3.3 Related work . . . 116
7.3.4 Application to compound noun interpretation . . . 118
7.3.4.1 Method . . . 118
7.3.4.2 Comparison of time requirements . . . 120
7.3.4.3 Results . . . 121
7.4 Conclusion . . . 125

8 Conclusions and future work **127**
8.1 Contributions of the thesis . . . 127
8.2 Future work . . . 128

Appendices

A Notational conventions **155**

B Annotation guidelines for compound nouns **157**

Chapter 1

Introduction

Noun-noun compounds are familiar facts of our daily linguistic lives. To take a simple example from my typical *morning routine*: each *weekday morning* I eat breakfast in the *living room*, while catching up on *email correspondence* and reading *news websites* or (if I'm feeling particularly diligent) some *research papers*. Unless my *training schedule* prescribes a *rest day*, I pack some *running clothes and shoes*. If the weather looks threatening I'll fetch my *rain jacket* before leaving. By the time I begin my *cycle commute* to the *University Computer Laboratory* I have already encountered a plethora of concepts that are most conveniently denoted by combinations of nouns.

The frequency of compounding is just one reason why it is a perennially popular topic in theoretical, computational and psychological language research.[1] Compounding is also notable for its great productivity. Almost any pair of English nouns can be combined to produce a valid, if not always sensible, compound, and users of a language routinely produce and comprehend compounds they have never heard before. If a friend were to tell you he had just bought a new *pineapple radio*, chances are you would have some idea of what he was referring to even though the term would most likely be new to you.[2] Based on your knowledge of pineapples, radios and the possible relations that can hold between them you might decide that a *pineapple radio* is probably a *radio that looks like a pineapple* or a *radio contained in a pineapple*, rather than a *radio used for eating pineapples* or a *radio owned by a pineapple*. Indeed, you may reach this conclusion without much consideration or even awareness of the compound's ambiguity.

In the context of natural language processing (NLP), an ability to unpack the semantics encoded in compounds is necessary for high-recall semantic processing. Due to the frequency and productivity of compounding, a system for semantic parsing or information extraction cannot simply ignore compounds without compromising recall, nor can it simply access a dictionary of known compounds as most items it encounters will not be listed. Yet despite the relative ease with which human speakers and hearers handle novel compounds, modelling the inferential processes involved has proven very challenging. One difficulty that particularly affects computational approaches is that the surface form of a

[1] In the course of my research I have compiled a comprehensive bibliography of publications on compound nouns, which can be found online at `http://www.cl.cam.ac.uk/~do242/bibsonomy.p.html`. At the time of writing this thesis, the bibliography contains 279 items, 157 of which have appeared since the year 2000 and 33 of which appeared in 2007.

[2] Google currently finds 41 hits for *pineapple radio* (not counting "very similar" pages omitted in the results), of which only a few instances are actual compounds referring to a kind of radio.

compound noun does not reliably disambiguate its meaning. It is not sufficient to simply associate particular constituent words with particular semantic relations. A *cheese knife* is a *knife for cutting cheese* and a *kitchen salesman* is a *salesman who sells kitchens*, yet a *kitchen knife* is neither a *knife for cutting kitchens* nor a *knife that sells kitchens*. Computational systems for interpretation must therefore approximate the kinds of conceptual knowledge that humans possess. In practice this is often done by extracting statistical information from large text corpora, which may seem a poor proxy for knowledge gained through experience of the world but actually works well for many tasks.

Reasoning about compound meaning involves working with at least two levels of semantics: lexical and relational. Reasoning at the lexical level involves processing information about the meanings of constituent words and comparing them to the constituents of other known compounds. The relational level involves knowledge about how particular kinds of entities tend to interact in the world and which semantic relations tend to be expressed in language. The general concepts of lexical and relational meaning are of fundamental importance in the broader field of computational semantics. Hence, one motivation for studying automatic compound interpretation is that it is a challenging test of core NLP methods, and approaches that work well on this task may be expected to be useful for a range of other problems. In my research I have pursued this theme by transferring the methods I develop for compound data to other semantic phenomena (Ó Séaghdha and Copestake (2008)). In Chapters 6 and 7 below, I show that my methods can be applied directly to the detection of general semantic noun-noun relations in sentences, attaining state-of-the-art performance on SemEval 2007 Task 4 (Girju et al., 2007) with minimal porting effort.

In this thesis I focus on developing computational methods for the classification of semantic relations in compound nouns and other noun-noun constructs. There are a number of interesting computational problems relating to compounds which I do not consider; these include machine translation of compounds (Rackow et al., 1992; Baldwin and Tanaka, 2004; Nakov and Hearst, 2007a), structural disambiguation of compounds with three or more constituents (Lauer, 1995; Nakov and Hearst, 2005), identification of constituents in languages where compounds are written as a single orthographical unit (Koehn and Knight, 2003), interpretation of deverbal nominalisations as a special case (Hull and Gomez, 1996; Lapata, 2002; Nicholson and Baldwin, 2006) and the role of compound analysis in improving information retrieval systems (Hoenkamp and de Groot, 2000; Karlgren, 2005; Pedersen, 2007).

The thesis is structured in two thematic halves. The first half deals with the construction of a semantically annotated corpus of compound nouns. Chapter 2 is a survey of prior work in theoretical and computational linguistics on appropriate frameworks for representing compound meaning. In Chapter 3 I motivate and describe the development of a new semantic annotation scheme for compounds, and in Chapter 4 I evaluate the reproducibility of this annotation scheme through a dual-annotator experiment. The second half of the thesis deals with computational methods for automatically classifying semantic relations between nouns in compounds and in other contexts. Chapter 5 presents a similarity-based approach to relation classification based on lexical and relational similarity, and describes kernel methods for machine learning. In Chapter 6 I show how the flexible kernel learning framework allows the implementation of classification methods that are particularly suitable for capturing lexical similarity. In Chapter 7 I investigate kernel methods for implementing relational similarity. I also show that techniques com-

bining lexical and relational similarity can achieve state-of-the-art performance on two relation classification tasks: compound noun interpretation and the SemEval 2007 task on identifying semantic relations between nominals (Girju et al., 2007). Chapter 8 contains a concluding summary and some speculation about promising directions for future research.

Chapter 2

Compound semantics in linguistics and NLP

2.1 Introduction

In this chapter I describe prior work on semantic representations for compounds in theoretical and computational linguistics.[1] The focus is on relational rather than referential semantics, so there is no discussion of questions concerning semantic headedness or metaphorical and metonymic reference. In Section 2.2 I outline the general status of compounding in linguistic theory and present a chronological and thematic survey of the semantic models that have been proposed by linguists. These proposals can be grouped into three broad classes: *inventory-based* theories, which posit a restricted set of general semantic relations, *pro-verb* theories, which provide only an underspecified skeletal representation of a compound's meaning and shift the task of further interpretation to pragmatic inference, and *integrational* theories, which generate structural representations of compounds by combining aspects of the constituent nouns. In Section 2.3 I describe the representations that have been used in prior computational work, showing that these also follow the trends seen in linguistic theory. Finally, I describe the representational assumptions that will be implemented in subsequent chapters of this thesis.

2.2 Linguistic perspectives on compounds

2.2.1 Compounding as a linguistic phenomenon

Compounding is used in a great variety of languages to create new words out of old. As was observed by Grimm (1826), compounding allows us to express complex concepts

[1]There is also a rich psycholinguistic literature on compound semantics, which I do not discuss here in detail. It is interesting to note that research in this field also fits the patterns observed in linguistics and NLP. In particular, many of the the models proposed can be described as being based on a restricted inventory of semantic relations, as in Gagné and Shoben's (1997) CARIN model, or on integrated representations produced by modifying or combining constituent structures (Murphy, 1990; Costello and Keane, 2000). There is also a *dual process* theory (Wisniewski, 1997; Estes, 2003), which brings the two perspectives together by proposing that some compounds are interpreted through assigning discrete relations while others are interpreted integrationally, through mapping features between structures.

more easily and elegantly than we could otherwise achieve.[2] Bauer (2001) gives a concise definition of the phenomenon: "We can now define a compound as a lexical unit made up of two or more elements, each of which can function as a lexeme independent of the other(s) in other contexts, and which shows some phonological and/or grammatical isolation from normal syntactic usage" (p. 695). Among the languages that do have compounding, not all use it with the same frequency or flexibility. For example, noun-noun compounds are rare in Romance languages such as French and translations of English compounds often take the form of a postmodified noun (*steel knife* ⇒ *couteau en acier*, *cheese knife* ⇒ *couteau à fromage*). On the other hand, Classical Sanskrit surpasses even English in its flexibility of compounding; nominalisations such as *khaṭvārūḍha* (?*bed-climbee*, *one who has climbed into bed*) and copulatives such as *candrasūrya* (*moon and sun*) do not have compound equivalents in English. English is nevertheless notable for the flexibility and frequency of its noun-noun compounding processes. The empirical data I discuss in Chapter 4 suggest that close to 3% of all words in the British National Corpus are constituents of compounds. Biber and Clark (2002) observe that the use of compounding has been increasing in recent times, especially in more formal genres such as news and academic writing, and it is among the most common methods in the language for noun modification.

Despite its prevalence, there is a tendency among linguists to view compounding as a particularly ill-behaved phenomenon that does not fit into standard categories. A long-running debate argues the existence of a putative distinction between compounds that function as words and compounds that function as phrases (Levi, 1978; Liberman and Sproat, 1992; Bauer, 1998). Libben (2006) writes that "compound words are structures at the crossroads between words and sentences reflecting both the properties of linguistic representation in the mind and grammatical processing". In morphology, Sadock (1998) argues that the morphological processes pertaining to compounding are separate from other morphological mechanisms. In semantics, many authors have banished compound meaning from the remit of their theories, and those who do give accounts of compound semantics often propose representations that are specific to that class of linguistic data (Section 2.2.2). The apparent idiosyncratic status of compounding has led Jackendoff (2002) to suggest that it is a "fossil" left over from an earlier, more basic stage of language evolution.

2.2.2 Inventories, integrated structures and pro-verbs: A survey of representational theories

As with many other areas of linguistic enquiry, the semantics of noun compounds was first investigated over two millennia ago by the scholars of the Sanskrit linguistic tradition (*vyākaraṇa*).[3] The famous categorisation of compounds as *bahuvrīhi* (exocentric), *tatpuruṣa* (endocentric), *avyayībhāva* (indeclinable) and *dvandva* (copulative) was described by Pāṇini in his *Aṣṭādhyāyī*. While these categories are now usually understood as distinguishing between compounds on the basis of their semantic heads (as in Bauer

[2] "Zweck der zusammensetzung scheint zu sein, daß dadurch begriffe leichter und schöner, als es sonst geschehen kann, ausgedrückt werden." (p. 407–408)

[3] The dates of the preeminent Sanskrit grammarians remain subject to debate, but Cardona (1976) presents evidence for dating Pāṇini not later than the early to mid fourth century B.C. and Patañjali around three hundred years later.

(2001)), their purpose in the *Aṣṭādhyāyī* was to license particular morphological and phonological operations and they were not directly related to semantics (Cardona, 1976). Later commentators developed further the semantic themes touched on by Pāṇini's grammar; Patañjali's *Mahābhāṣya* contains discussion of such topics as the semantic basis of the compounding operation, the structural ambiguity of three-noun compounds and the semantic distinctions between heads and modifiers (Joshi, 1968).

In the 20th century, compound noun semantics once again became a popular topic of linguistic research. Early work tended to focus on description, taking an inventory-style approach of documenting the variety of semantic relations observed in attested compounds (Noreen, 1904; Jespersen, 1942; Hatcher, 1960).[4] The rise of generative grammar in the 1950s and 1960s led to a greater concern with matters of representation. In a transformational (or at least multistratal) framework it made sense to analyse compound nouns as derived from a fuller structure at a deeper representational level, but what that deeper level should look like was the subject of much debate. Many proposals of this era can be viewed as continuing the "relation inventory" tradition. Adopting a highly restricted set of possible semantic relations was deemed necessary to avoid the theoretically undesirable phenomenon of *irrecoverable deletion*, whereby the deeper levels of a derivation could not be recreated from the surface form. Outside of a strict transformational framework this can be seen as a desire to reduce the degree of ambiguity allowed to compounds in the interests of maintaining an orderly linguistic theory. Lees (1970) describes an underlying representation where the constituents of a compound fill the thematic role slots of one of a small number of *generalised verbs*. For example, *air rifle*, *motor car* and *water wheel* are all derived from triples *V-Object-Instrument*, where V is a generalised verb with the meaning shared by the English verbs *energise, drive, power, actuate, propel, impel,...*, the Object role is assigned to the modifier noun and the Instrument role is assigned to the head. Other inventory-style analyses of this period include Li (1971) and Warren (1978); the latter is notable for its empirical basis and aim for comprehensive coverage, arising from a study of a large corpus of naturally occurring compounds.

Levi (1978) presents a highly detailed analysis of complex nominals, i.e., noun-noun compounds, nominalisations and nominals containing nonpredicating adjectives.[5] The semantics of noun-noun compounds are captured by nine *Recoverably Deletable Predicates (RDPs)*, which are similar in function to Lees' generalised verbs. The RDPs proposed by Levi are as follows:

CAUSE_1	*flu virus*	CAUSE_2	*snow blindness*
HAVE_1	*college town*	HAVE_2	*company assets*
MAKE_1	*honey bee*	MAKE_2	*daisy chains*
USE	*water wheel*		
BE	*chocolate bar*		
IN	*mountain lodge*		
FOR	*headache pills*		
FROM	*bacon grease*		
ABOUT	*adventure story*		

[4] An even earlier inventory of semantic relations was given by Grimm (1826), who analysed compounds in terms of prepositional, appositional or case-like meaning.

[5] Nonpredicating adjectives are adjectives that rarely, if ever, appear in copula (predicative) constructions and are shown by Levi to function much like noun modifiers, e.g., *solar (heating)*, *automotive (emissions)*, *mechanical (engineer)*.

A *flu virus* is a *virus that CAUSES flu*, an *adventure story* is a *story ABOUT adventure*, and so on. The three RDPs *CAUSE*, *HAVE* and *MAKE* each have two variants, as either the head or modifier of a compound can fill the first argument of these predicates, while the other arguments are either symmetric (*BE*) or restricted to taking the compound head as first argument.[6] RDPs are deleted in the derivation of compound nominals, and hence all compounds are ambiguous in exactly 12 ways. Levi deems this degree of ambiguity to be sufficiently restricted for a hearer to identify the relation intended by a speaker by recourse to lexical or encyclopaedic knowledge, while still allowing for the semantic flexibility that undoubtedly characterises compound nouns. In Levi's theory, nominalisations such as *student demonstration* and *draft dodger* are produced by a separate set of rules and are assigned to separate categories.[7] This is primarily motivated by the theoretical framework she is working in and the assumption that the relational predicates underlying nominalisations are related to lexical verbs rather than primitive RDPs. However, the separation of nominalised and non-nominalised relations leads to arbitrary distinctions (*horse doctor* is *FOR*, but *horse healer* would be *AGENT*) and seeming inconsistencies (Levi labels *hysterical paralysis* as $CAUSE_2$ but *dream analysis* as *ACT*).

Inventory-style analyses have been criticised from many angles. One criticism holds that the variety of compound relations is so great that listing them is impossible. Downing (1977), considering such idiosyncratic compounds as *plate length*,[8] writes (p. 828): "The existence of numerous novel compounds like these guarantees the futility of any attempt to enumerate an absolute and finite class of compounding relationships".[9] Zimmer (1971) proposes that it is more sensible to list the semantic relationships that *cannot* be encoded in compounds rather than those that can. Zimmer's solution is to exclude relations that fail the test of *classificatory appropriateness*, a vaguely defined concept applied to relations that are useful for distinguishing or naming entities.[10] Hence it is usually inappropriate to refer to a town far from the sea as a *sea town* or to a cat that happens to be in a tree at an arbitrary time as a *tree cat*.

A second criticism is that the relations proposed in the inventories of Levi and others are nevertheless too general and vague. It is often hard to say which relation should apply to a certain compound, and there are many cases where many relations seem appropriate. Levi recognises an amount of "analytic indeterminacy" in her model whereby particular classes of compounds can have two analyses: *made-of* compounds such as *copper coins* and *chocolate bunny* are equally analysable as *BE* or $MAKE_2$, while *part-whole* compounds such as *ocean floor* and *brain cells* can be labelled $HAVE_2$ or *IN*. This kind of indeterminacy is not limited to the cases mentioned by Levi; Lehnert (1988) gives the example of *dog collar* (*collar USED by a dog* or *collar that a dog HAS*?), and I discuss

[6] The annotation scheme for compound semantics that I develop in Chapter 3, which is derived from Levi's inventory, permits all relations (except symmetric *BE*) to take both possible constituent-argument mappings. Examples of analyses not permitted in Levi's framework are *boulder field* (IN_1) and *sitcom character* ($ABOUT_1$).

[7] The relations for nominalisations are *ACT*, *AGENT*, *PRODUCT* and *PATIENT*.

[8] "What your hair is when it drags in your food"

[9] In an oft-quoted statement, Jespersen (1942) makes a similar point (p. 137–8): "On account of all this it is difficult to find a satisfactory classification of all the logical relations that may be encountered in compounds. In many cases the relation is hard to define accurately... The analysis of the possible sense-relations can never be exhaustive." As noted above, these sentiments did not prevent Jespersen from proposing a non-exhaustive inventory of relations.

[10] Zimmer (1971), p. C15: "Anything at all can be described, but only relevant categories are given names."

further examples in Section 3.3. These issues are not unique to Levi's analysis; rather, they must be addressed by any model that uses a restricted set of relations to describe compound semantics. In Chapter 3 I describe how they are also relevant to computational research and show how the amount of indeterminacy in the model can be reduced through a rigorous characterisation of the boundaries between relations.

A third criticism of restricted inventories is that they give too impoverished a representation of compound semantics. On this view, the meaning of a compound cannot be reduced to one of a small number of general relations. Downing (1977) cites the examples *headache pills* and *fertility pills*, which are both analysed as *FOR* by Levi but have very different relational semantics. Other examples of over-reduction given by Downing are *dinner-bath*, interpreted by a subject in an experiment as *a bath taken in preparation for dinner*, and *oil-bowl*, explained as *the bowl into which the oil in the engine is drained during an oil change*. She writes: "These interpretations are at best REDUCIBLE to underlying relationships as suggested by Li and others, but only with the loss of much of the semantic material considered by subjects to be relevant or essential to the definitions" (p. 826).

Some authors have chosen to sidestep the problems of inventory approaches by eliminating the complexities of compound meaning from the proper domain of semantic processing (Gleitman and Gleitman, 1970; Bauer, 1979; Selkirk, 1982; Lieber, 2004). The semantics of a compound is then simply the assertion of an unspecified relation between the referents of its constituents, and the task of identifying what manner of relation that might be is passed on to a combination of world knowledge, discourse context and inferential mechanisms under the general term "pragmatics". In the approaches of Gleitman and Gleitman (1970) and Bauer (1979) this is formalised by the introduction of a minimally specific *pro-verb* (as in *pronoun*) in the underlying representation of compounds. Hence Bauer (1979) writes (p. 46): "The gloss given above for the abstract pro-verb of compounding can... be expanded to read 'there is a connection between lexeme A and lexeme B in a compound of the form AB such as can be predicted by the speaker/hearer partially on the basis of her knowledge of the semantic make-up of the lexemes involved and partially on the basis of other pragmatic factors'." While this fix rescues formal models from the pitfalls of uncontrolled ambiguity and non-compositionality, it is far from pleasing as an account of compound meaning. It relies on pragmatic mechanisms of interpretation, but the nature of these mechanisms has rarely been specified.[11] As even sceptics such as Jespersen (1942) and Downing (1977) recognise, there are many useful generalisations that can be made about compound semantics. It is striking that in spite of the variety of compounding theories, the inventories that have been proposed are more notable for their commonalities than their differences. There is no doubt that large classes of compound nouns describe relations of location, possession or topichood, even if the labels used for these classes gloss over finer details and many other compounds have idiosyncratic meanings.

While the pro-verb analysis can be viewed as the product of the inventory-makers' reductionist tendencies taken to their natural conclusion, other linguists have proposed richly detailed and structured representations of compound semantics. Researchers working in the tradition of cognitive linguistics deny the existence of a divide between compositional semantic structures and "pragmatic" kinds of conceptual and contextual knowledge. They also reject distinctions between the productive and interpretive processes relating to compounding and other combinatory processes in language. In the models of Ryder (1994)

[11] Notable exceptions are Hobbs et al. (1993) and Copestake and Lascarides (1997).

and Coulson (2001), language users combine their knowledge about the constituents of a compound to arrive at an integrated representation of its meaning. Unlike the semantic relations posited in the inventories of Levi (1978) and others, the relations assumed by Ryder and Coulson do not exist independently of their arguments, but rather emerge from properties of those arguments in the process of conceptual combination or *blending*.

One kind of knowledge that is central to this process takes the form of *event frames*, schematic representations of the events or situations in which an entity typically plays a role. For example, to interpret the compound *cheese knife*, a hearer accesses his/her knowledge about knives and cheese, which includes the information that knives by design are strongly associated with cutting and are not infrequently used in the preparation of food, and that cheese is a sensible thing to cut. Contextual and other factors being equal, the resulting interpretation will represent a knife that has an associated event frame of cutting cheese and differs from a generic knife representation in ways that reflect the hearer's beliefs about what makes a knife appropriate for cheese-cutting. This representation can then be further integrated into the frame corresponding to the current sentence or discourse. If needs be, the compound meaning can be fleshed out with other relevant beliefs, as in Downing's (1977) example of *oil-bowl* where the subject extended the basic relational semantics of containment to suggest a context in which such an object might be used. In cognitive theories, an emphasis is placed on the creation of meaning by both the speaker and hearer, and the analysis extends to metaphorical combinations such as *stone lion*, which has some properties of lions but lacks others, and *trashcan basketball*, which is introduced by Coulson (2001) as a game superficially resembling basketball in which crumpled balls of wastepaper are thrown in a wastepaper bin.[12]

Regularities and patterns in compound meanings arise in cognitive models not because they are generated by a finite set of rules, but rather because of regularities and patterns in how language users experience and conceptualise their environment. That is, compounds frequently encode locative relations because knowledge about an entity's location can suggest further distinctive facts about the entity (a *mountain hut* will usually look quite different to a *beach hut*) and illuminate its relation to other entities and events in the discourse. Likewise event frames play an important role because we have a strong tendency to categorise entities according to the events they can or typically do partake in. Ryder (1994) also proposes that regularities in production and interpretation are enforced by analogical pressures from previously encountered compounds. Beyond the simple case where a compound acquires a conventional meaning through repeated exposure, Ryder suggests that speakers and hearers generalise more abstract templates relating constituents to probable meanings. These templates vary in reliability and specificity; some require the presence of particular lexical items (e.g., *X + box = box for putting X in*, *sea + X = a metaphorical extension of X that lives in the sea*), while others are very general (*Location Y + X = X typically found in Y*). The most general and reliable templates correspond to Levi-style rules.

The great flexibility which is an undoubted advantage of cognitive theories also contributes some disadvantages: while frame semantics can *explain* that *trashcan basketball* blends aspects of trashcans and basketball in a metaphorical way, it is not (yet) able to *predict*

[12]The active role of the hearer in creating his/her own understanding is underlined by Coulson (p. 141): "Because the function of language is to enable the listener to participate in the interactive frame set up in a shared context, a noun phrase need only provide the listener with enough information about the element in question to connect the phrase with contextual information and/or background knowledge."

why the compound has exactly that meaning and not another. To achieve predictive power for even standard compositional compounds, a great deal of representational and inferential details must be spelled out. Researchers working in Pustejovsky's (1995) Generative Lexicon framework have attempted to produce a theory of compound meaning that combines the structural richness of frame semantics with the tractability of more restricted analyses (Johnston and Busa, 1996; Søgaard, 2005). In these approaches, the lexical entry for a noun is enriched with *qualia structures*, which represent particularly salient aspects of its meaning. There are four kinds of qualia structure:

Constitutive:	What the entity is made of
Formal:	The entity's ontological status
Telic:	The entity's purpose or function
Agentive:	How the entity was created

Thus the representation of *steel knife* is constructed by matching the denotation of *steel* to the argument of the constitutive quale in the representation of *knife*, and *cheese knife* is represented by matching *cheese* with the object of the event of cutting, which is the value of *knife*'s telic quale. The restricted nature of qualia structures mean that the combination of two concepts can be predicted with some reliability. This approach works best for man-made artefacts which have a definite function; for many other entities it is harder to state what the appropriate qualia values are. For example, what is the telic quale of *dog*? Is it herding (*sheep dog*) or hunting (*bird dog*), or something else entirely (*gun dog*, *police dog*)? Providing a comprehensive account of compounding relations in this framework would seem to entail enriching the lexicon to a degree that the boundary with full frame-semantic theories becomes unclear.

2.3 Compounds and semantic relations in NLP

2.3.1 Inventories, integrated structures and pro-verbs (again)

The kinds of semantic representations used in computational treatments of compound interpretation mirror those proposed by linguistic theorists. Approaches based on relation inventories have often been favoured due to their tractability; they can robustly analyse compounds with previously unseen constituents and are well-suited to the paradigm of statistical multiclass classification. More structured representations and "emergent" representations that are informed by the semantics of compound constituents have also been investigated, but these approaches face a number of challenges that have yet to be surmounted. Extensive lexical engineering is often required, and the resulting interpretations are difficult to evaluate precisely due to their richness. Analogues of the "pro-verb" analysis have also been proposed for broad-coverage semantic parsing systems, often with the expectation that the underspecified output representation can be passed onto general or domain-specific inference systems for further disambiguation. This approach was pursued in early work on natural language database interfaces, e.g., by Boguraev and Spärck Jones (1983), and is also implemented in the English Resource Grammar (Copestake and Flickinger, 2000).

2.3.2 Inventory approaches

Su (1969), to my knowledge the first researcher to report on compound interpretation from a computational perspective, describes 24 semantic categories to be used for producing paraphrase analyses of compounds.[13] These categories contain many relations familiar from linguistically motivated inventories: *Use*, *Possessor*, *Spatial Location*, *Cause*, etc. A second early work is by Russell (1972), who implements a compound interpreter for a small lexicon. Russell's set of semantic relations is slightly different in that it consists of all "semantic dependencies" expected to occur between nouns in general semantic analysis, but the standard compound relations are all featured. Other inventories that have been proposed for compound analysis include those of Leonard (1984) and Vanderwende (1994). A set of domain-specific relations for biomedical noun compounds is described by Rosario and Hearst (2001).

Lauer (1995) proposes a classification of compounds based on prepositional paraphrasing. His relation inventory contains eight prepositions: *about*, *at*, *for*, *from*, *in*, *of*, *on* and *with*. Hence a *baby chair* is a *chair for a baby*, *reactor waste* is *waste from a reactor* and a *war story* is a *story about war*. The distinctive characteristic of this inventory is that its members are lexical items, not the abstract relational concepts stipulated by other theories. This allows the use of unsupervised statistical learning methods that require little human engineering effort. The most probable relation for a noun-noun compound can be estimated by simply counting preposition-noun co-occurrences in a corpus or on the World Wide Web (Lauer, 1995; Lapata and Keller, 2004). However, the "surfacy" nature of Lauer's relations also brings disadvantages. Prepositions are themselves polysemous lexical items, and the assignment of a prepositional paraphrase to a compound does not unambiguously identify its meaning.[14] In other words: once we have identified a compound as, say, an *of*-compound, we still must ask what kind of *of* we are dealing with. The paraphrases *school of music*, *theory of computation* and *bell of (the) church* seem to describe very different kinds of semantic relations. Furthermore, the assignment of different categories does not necessarily entail a difference in semantic relations. The categories *in*, *on* and *at* share significant overlap (if not near-synonymy), and the distinction between *prayer in (the) morning*, *prayer at night*, and *prayer on (a) feastday* seems rooted in shallow lexical association rather than any interesting semantic issue. It seems fair to conclude that while prepositional paraphrases clearly correlate with underlying semantic relations, they do not reliably map onto those relations.[15] Another problem is that many compounds cannot be paraphrased using prepositions (*woman driver*, *taxi driver*) and are excluded from the model, while others admit only unintuitive paraphrases (*honey bee* = *bee for honey*?).

The relational model introduced by Nastase and Szpakowicz (2001) for general semantic text processing and applied to noun-modifier interpretation by Nastase and Szpakowicz

[13] It is not clear from the technical report whether the method described by Su was actually implemented.

[14] The disambiguation of prepositions has been studied as a difficult NLP task in its own right, for example in a task at the 2007 SemEval competition (Litkowski and Hargraves, 2007).

[15] In an interesting experiment, Girju et al. (2005) investigated the predictive power of Lauer's model by training an SVM classifier to recognise relations from their own inventory (see below), using data annotated with the appropriate prepositional paraphrase. They found that adding these prepositional features increased performance by about 20 points but classification remained far from perfect, with best scores of 66.8% and 83.9% on their two datasets.

(2003) provides a large inventory of semantic classes. As they are not tethered to specific lexical items, Nastase and Szpakowicz' relations do not share the problems that affect Lauer's prepositions. To avoid the sparsity issues that inevitably affect such a fine-grained set of classes when performing data-driven classification with small datasets, the relations are grouped into five supercategories: *CAUSALITY*, *TEMPORALITY*, *SPATIAL*, *PARTICIPANT* and *QUALITY*. The 30 relations used by Nastase and Szpakowicz are as follows:[16]

CAUSALITY		**PARTICIPANT**	
CAUSE	*flu virus*	AGENT	*student protest*
EFFECT	*exam anxiety*	BENEFICIARY	*student discount*
PURPOSE	*concert hall*	INSTRUMENT	*laser printer*
DETRACTION	*headache pill*	OBJECT	*metal separator*
		OBJECT PROPERTY	*sunken ship*
QUALITY		PART	*printer tray*
CONTAINER	*film music*	POSSESSOR	*group plan*
CONTENT	*apple cake*	PROPERTY	*novelty item*
EQUATIVE	*player coach*	PRODUCT	*plum tree*
MATERIAL	*brick house*	SOURCE	*olive oil*
MEASURE	*saturation point*	STATIVE	*cell division*
TOPIC	*weather report*	WHOLE	*daisy chain*
TYPE	*oak tree*		
TEMPORALITY		**SPATIAL**	
FREQUENCY	*daily exercise*	DIRECTION	*exit route*
TIME AT	*morning exercise*	LOCATION	*home town*
TIME THROUGH	*six-hour meeting*	LOCATION AT	*desert storm*
		LOCATION FROM	*country butter*

These relations seem to be better candidates than Lauer's for representing "deep" semantic knowledge. The size of the inventory facilitates fine distinctions in meaning, e.g., between *headache pill* (*DETRACTION*) and *fertility pill* (*CAUSE* or *PURPOSE*), though this comes at the cost of sparsity and imbalance in the distribution of relations. Nastase and Szpakowicz' (2003) annotated dataset has subsequently been used in numerous classification experiments (Nastase et al., 2006; Turney, 2006; Nulty, 2007a) and their relations have been used to annotate new data by Kim and Baldwin (2005). There is arguably a degree of incoherence in the supercategory groupings: *PART/WHOLE* and *CONTAINER/CONTENT* belong to separate supercategories, while *QUALITY* subsumes a variety of relations, including hyponymy, containment and topichood. As observed in Chapter 3, no annotation guidelines have been reported for this inventory and there are many cases where more than one relation seems appropriate for a particular compound.

Girju and colleagues have developed an inventory that is similar in spirit to Nastase and Szpakowicz' model and shares many of its advantages. Different versions of this inventory have appeared – Girju et al. (2005) describe 35 relations of which 21 are attested in their

[16] I include only those relations that were attested in Nastase and Szpakowicz' (2003) dataset of 600 items. A sixth supercategory, *CONJUNCTIVE*, features in the set of 50 relations they initially considered but was not attested. The dataset contains examples of both noun and adjective modifiers; adjectival examples have been used only where there are no noun-noun examples for the relation in question.

data, while Girju (2006; 2007a) describes 22 relations. The newer set of relations is as follows:[17]

POSSESSION	*family estate*	KINSHIP	*sons of men*
PROPERTY	*pellet diameter*	AGENT	*insect bites*
TEMPORAL	*night club*	DEPICTION	*caressing gestures*
PART-WHOLE	*hawk wings*	HYPERNYMY	*coyote pup*
CAUSE	*fire shadows*	MAKE/PRODUCE	*sun light*
INSTRUMENT	*cooking plate*	LOCATION	*staircase door*
PURPOSE	*identity card*	SOURCE	*orange juice*
TOPIC	*war movie*	MANNER	*performance with passion*
MEANS	*bus service*	EXPERIENCER	*consumer confidence*
MEASURE	*fishing production*	TYPE	*member state*
THEME	*cab driver*	BENEFICIARY	*victim aid*

2.3.3 Integrational approaches

The general semantic interpretation system of McDonald and Hayes-Roth (1978) uses a semantic network to represent lexical knowledge. The nodes in the network correspond to words and the links between nodes are based on information extracted from dictionary definitions. The meaning of a noun-noun compound is processed by applying heuristics to integrate directed paths originating in the compound constituent nodes. To interpret the compound *lawn mower*, the system first adds a new node to the network, connected by an *IS-A* link to *mower* and a *MODIFIED-BY* link to *lawn*. The network representation of *lawn* is derived from the definition *A lawn is a mown area or plot planted with grass or similar plants* and that of *mower* is derived from *A mower is a machine that cuts grass, grain or hay*. Through a heuristic search procedure, links are added to *lawn mower* that essentially specialise the representation of *mower*, yielding an integrated representation that corresponds to the definition *A lawn mower is a machine that cuts grass or similar plants*.

Another system developed around the same time was that of Finin (1980), which interpreted compounds in a restricted domain (naval aircraft maintenance and flight records). Finin's system used a frame-like representation of the attributes and typical event structures associated with nouns. Interpretation of non-lexicalised compounds proceeded through the application of rules integrating the frame information for the two constituents (e.g., *F4 planes*, *woman doctor*, *crescent wrench*) or by matching one constituent to the argument slot of the event structure associated with the other constituent (e.g., *maintenance crew*, *engine repair*, *oil pump*). The systems described by McDonald and Hayes-Roth and Finin were necessarily restricted to toy implementations or closed vocabularies. These knowledge-rich approaches, and similar ones such as Isabelle (1984) or implementations of the Generative Lexicon theory, suffer from the inherent difficulties of constructing robust large-scale knowledge bases and of avoiding an exponential increase in complexity as the system grows. The first problem at least may now be tractable, with recent promising developments in the efficient construction of very large semantic networks (Harrington

[17] The examples are taken from the annotated datasets of Girju (2007a) (available from `http://apfel.ai.uiuc.edu/resources.html`), with the exception of the *MANNER* and *MEANS* relations which are not attested. The data contains instances of nominal and prepositional modifiers; the former have been used as examples wherever possible (there is no noun-noun instance of *KINSHIP*).

and Clark, 2007) and the automatic extraction of qualia structures for nouns (Cimiano and Wenderoth, 2007; Yamada et al., 2007).

A second strand of research that seeks to go beyond restricted inventories uses what can be called *emergent representations*. Here the range of possible semantic relations expressed by compounds is not determined in advance, but is generated by the data itself. Rosario et al. (2002) report a study in which compound meanings were associated with pairs of concepts in a domain-specific hierarchical lexical ontology. The constituents of each compound were mapped onto the ontology's top level and then specialised by moving down in the hierarchy to remove relational ambiguities. For example, *scalp arteries*, *heel capillary* and *limb vein* were all mapped onto the same pair of lexical concepts (*Body Regions-Cardiovascular System*) and are judged to express the same relation. This method generates thousands of concept pairs of varying frequency and specificity, and is shown to accurately generalise to unseen concept pairs.

Another exploratory approach to compound interpretation is Nakov and Hearst's (2006) method for discovering verbal paraphrases for compounds through search engine queries. By submitting queries such as N_2 *that*|*which*|*who* * N_1 to Google and extracting verbs from the returned snippet strings, Nakov and Hearst identify the predicates most likely to link the constituents of a compound N_1 N_2. For example, the top verbs returned for the compound *migraine drug* are *treat, be used for, prevent, work for, stop*. While it is difficult to perform intrinsic evaluation of this method, Nakov (2007) demonstrates that the information it extracts can provide useful features for many tasks including verbal analogy solving, identification of relations in text and machine translation of compounds. Nakov (2008) investigates how more natural paraphrase information can be obtained from non-expert human subjects through online experiments.

2.4 Conclusion

In this chapter I have surveyed the spectrum of representational theories that have been proposed for compound noun relations. The experiments in compound annotation and classification that I present in subsequent chapters assume an inventory-style representation. Using a small fixed set of relations facilitates the application of multiclass machine learning methods and of standard evaluation methods for evaluating classifier performance. It also reduces the sparsity problems associated with small-to-medium-sized datasets. Restricted inventories may not capture the finer aspects of meaning that rich lexical structures do, but given the current state of the art success at coarse-grained compound interpretation would constitute significant progress. Furthermore, the two representational philosophies are not exclusive: coarse relations can be useful for reasoning about generalisations over more specific compound meanings.

Looking beyond the special case of compound nouns, the general concept of semantic relation is a fundamental one in all fields of language research. Many well-established NLP tasks involve identifying semantic relations between words in a text (e.g., semantic role labelling, relation classification) or between concepts (automatic ontology creation, relation extraction). The kinds of relations that are studied vary greatly, from lexical synonymy to protein-protein interactions, from binary predicates to complex structures. Some of the methods that have been developed for these tasks are outlined in Chapter 5. In Chapter 6 I will describe a problem of recognising particular relations between nouns

in text, showing that it is not dissimilar to compound interpretation and is amenable to the same computational approaches.

Chapter 3

Developing a relational annotation scheme

3.1 Introduction

As outlined in Chapter 2 the past 50 years have seen a great proliferation of relation inventories for the theoretical and computational analysis of compound nouns. Despite this fact, it became clear to me during the initial stages of my research that it would be of benefit to work on developing a new compound annotation scheme. One primary motivation for this decision was the non-availability of detailed annotation guidelines, making it extremely difficult to adopt an existing scheme for annotation of new data. A second motivation was the low levels of agreement reported by those researchers who have performed multiple-annotator evaluations of semantic annotation schemes (Section 4.5).

This chapter describes the new scheme I have developed for annotating compound noun semantics and the guidelines that accompany it. In Section 3.2 I list a number of criteria which can be used to evaluate and compare semantic annotation schemes. Section 3.3 describes the development procedure and the most significant decisions that guided the design of the new scheme. This is followed by a summary description of the finished relation inventory as well as the theoretical concepts that underlie it.

3.2 Desiderata for a semantic annotation scheme

In deciding on a classification scheme for compound relations, we are trying to pin down aspects of human conceptualisation that cannot be described using clear-cut observable distinctions such as syntactic patterns or cue phrases. However, it is important not to choose a classification of relations on the sole basis of introspective intuition, as there is no guarantee that two subjects will share the same intuitions and it does not give us a basis to select one scheme among many. When dealing with semantics it is therefore crucial that decisions are based on solid methodological concerns. That said, the literature on "best practice" for semantic annotation schemes is rather sparse. The compound annotation task shares some of the nature of ontology building and semantic field analysis, for which some design guidelines have been given by Hovy (2005) and Wilson and Thomas (1997)

respectively. The discussion in this section has much in common with Wilson and Thomas' proposals.

Faced with the need to select an appropriate classification scheme for compound relations, I identified a number of desirable criteria. They are sufficiently general to have relevance for all semantic annotation studies. Most have an *a priori* theoretical motivation but they are also informed by the experience of developing the compound annotation scheme and became clear in the course of the development process:

1. **Coverage: The inventory of informative categories should account for as much data as possible.** Semantic annotation tends to be labour-intensive and the amount of data that can be annotated is usually restricted by the resources at hand. It is therefore desirable to maximise the amount of annotated data that can subsequently be used for experiments. For example, if items assigned to a "miscellaneous" category are to be discarded, the coverage of the other categories should be expanded so that the proportion of data assigned "miscellaneous" is minimised. Discarding some classes of data can also lead to arbitrary patterns of exclusion. The compound relation inventories of Levi (1978) and Lauer (1995) do not assign semantic relations to compounds whose head is a nominalised verb and whose modifier is an argument of that verb, leading to the unintuitive situation where *history professor* is assigned a semantic relation and *history teacher* is assigned to a different, syntactically motivated category (Levi) or to no category at all. Lauer's scheme, which identifies semantic relations with prepositional paraphrases, also excludes appositional compounds such as *woman driver* as they cannot be paraphrased prepositionally.

2. **Coherence: The category boundaries should be clear and categories should describe a coherent concept.** If categories are vague or overlapping then consistent annotation will be very difficult. Detailed annotation guidelines are invaluable for the clarification of category boundaries, but cannot save a scheme with bad conceptual design.

3. **Generalisation: The concepts underlying the categories should generalise to other linguistic phenomena.** The regularities we hope to identify in compound relations or similar phenomena are assumed to reflect more general regularities in human semantic processing. Such regularities have been studied extensively by researchers in cognitive linguistics, and a categorisation scheme can be defended on the basis that it is consistent with and supported by those researchers' findings.

4. **Annotation Guidelines: There should be detailed annotation guidelines which make the annotation process as simple as possible.** Where possible, guidelines should be made publicly available to aid comparison of annotation schemes.

5. **Utility: The categories should provide useful semantic information.** The usefulness of a classification scheme is a subjective matter, and depends on how the annotated data will be applied. However, we can impose minimal criteria for utility. Each label in the scheme should be unambiguous and should carry truly semantic information. Hence Lauer's prepositional categories do not meet this requirement,

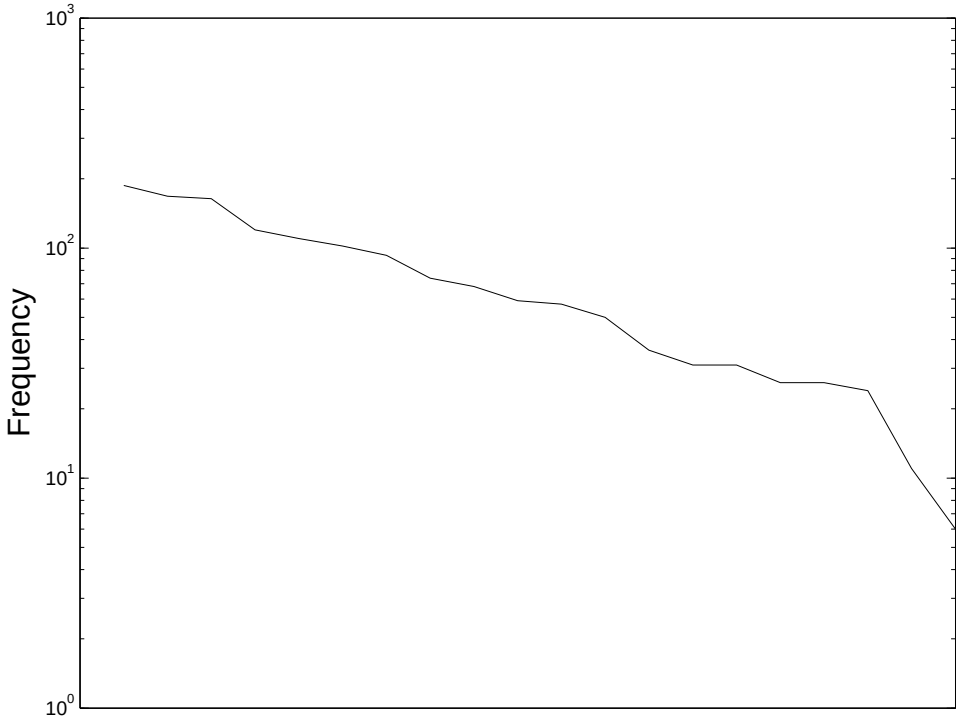


Figure 3.1: Log plot of annotation rules for the six categories *BE*, *HAVE*, *IN*, *ACTOR*, *INST* and *ABOUT* in the 2,000-compound dataset described in Chapter 4

> given the inherent ambiguity of prepositions (Section 2.3.2). A further concern affecting utility is the selection of granularity level, which must be fine enough for the intended application yet coarse enough to facilitate non-trivial generalisations about the data.

The initial version of this list published in Ó Séaghdha (2007b) also recommended that the distribution of categories should aim for balance, all other factors being equal. This recommendation has proven controversial and may warrant reconsideration. One argument against balance is the assumption that semantic relations will usually have a skewed distribution, as is known to be the case for lexical items and has also been argued for word senses (Kilgarriff, 2004). For example, it has been observed that the distribution of relations in Nastase and Szpakowicz' (2003) collection of modifier-noun pairs appears similar to a Zipfian one.[1] However, given that the "true" set of compound relations is unknown, the distribution of categories in such a dataset cannot be disentangled from the assumptions underlying its construction. For example, lexicographers and other designers of categorisation schemes are often classed as "lumpers" or "splitters", depending on their degree of willingness to tolerate heterogeneity within a single class or to create new and finer classes, respectively (Hanks, 2000).

Where an annotation scheme is developed incrementally through examination of naturally occurring data, a lumper approach would be expected to yield skewed distributions due to a "rich get richer" pattern: new instances will tend to be added to existing categories rather than used to establish new categories. I observed such a pattern during the development of my annotation guidelines for compounds (see Section 3.4), where the annotation rules initially devised for each category tended to reflect more central and

[1] Peter Turney, personal communication.

frequent examples of the category and rules added later tended to be more peripheral and rare. In the annotated dataset of compounds that is described in Chapter 4, the observed distribution of annotation rule applications within each category is skewed and the resulting overall distribution of rules is close to Zipfian (see Figure 3.1). However, it would have been possible to partition the space of semantic relations differently so as to make the distribution of category labels more or less skewed, without affecting the true underlying relations. Regarding the primary level of semantic categories, a decision was made to strive for conceptual (as opposed to purely numerical) balance so that the choice between categories by human or machine would entail comparing like with like. As described below, this resulted in six main semantic categories that were felt to have comparable conceptual granularity and provided high but not exhaustive coverage of the space of all observed compound relations.

It is clear that the five desiderata described above can interact in different ways. A good set of annotation guidelines will enhance category coherence. On the other hand, a more "surfacy" set of categories may be easier to annotate with but provide less useful semantic information. In the language of optimisation theory, it is difficult to know how to recognise a global optimum (the best possible solution) or even whether such an optimum exists.

How then can these criteria be used to judge an annotation scheme? Generalisation can primarily be argued with theoretical evidence, though in the context of large-scale lexicon development the use of linguistically general building blocks could be demonstrated by parsimony of representation. Likewise, utility is subjective, but a context-specific notion of utility can be tested by the suitability of annotated data for particular applications. The other criteria can be evaluated empirically through annotation experiments. Coverage can be directly measured from an annotated corpus as the proportion of data that is assigned an "informative" relation, i.e. one other than *OTHER*, *UNKNOWN*, etc. Ease of annotation can be estimated through inter-annotator agreement between multiple annotators. Problems with coherence can be identified by analysis of inter-annotator disagreements. A definitive comparison of multiple schemes would require annotation of a single corpus with every scheme, but in practice this is rarely done.

3.3 Development procedure

The set of nine compound relations (*BE*, *HAVE*, *IN*, *ABOUT*, *FOR*, *MAKE*, *CAUSE*, *USE*, *FROM*) proposed by Levi (1978) was taken as an initial classification scheme. As described in Section 2.2.2, Levi's proposals are informed by linguistic theory and by empirical observations, and they intuitively seem to comprise the right kind of relations for capturing compound semantics. Their granularity also seems appropriate for the dataset sizes used in NLP research on compounds, which are usually in the very low thousands. In attempting to annotate trial data with this scheme, however, a number of problems were identified that necessitated major revisions:

- The *CAUSE* relation is extremely infrequent, with only two unambiguous examples (*blaze victim* and *staff cost*) identified in a sample of 300 compounds.
- *MAKE* is also a scarce relation (9 occurrences in 300). More seriously, most if not all examples given by Levi for this relation can also be analysed as expressing other

relations (for example, *sap tree* is also *HAVE*, *music box* is also *FOR* and *sugar cube* is also *BE*).

- Nominalisations are analysed with a separate set of relations. This is due to the assumptions of Levi's linguistic theory and not desirable under my approach.
- More generally, Levi does not provide detailed guidelines for the application of her categories, and is not concerned with avoiding overlapping or vague category boundaries.

The annotation scheme was refined over the course of six months through a series of annotation trials followed by analysis of disagreements and changes in the scheme. Extensive guidelines were developed to clarify the application of the categories and the boundaries between them. The most serious and pervasive problem encountered was that most compounds can be assigned multiple semantic relations even when their meanings are clear, though only one category per compound is permitted by the desired experimental design. For example, a *car factory* is plausibly a *factory for producing cars* (*FOR*), a *factory that causes cars to be created* (*CAUSE*), a *factory in which cars are produced* (*IN*) and a *factory from which cars originate* (*FROM*). This does not reflect an ambiguity in the semantics of *car factory*, as the interaction between the factory and the cars is the same under each paraphrase. Rather, the problematic ambiguity lies in the question of which category best fits this interaction. In the same way, an *office chair* can be a *chair typically used/found in an office* (*IN*), a *chair for use in an office* (*FOR*) and a *chair belonging to an office* (*HAVE*).[2] This phenomenon is problematic not just for Levi's scheme, but also for most other relation inventories described in the literature.[3] Devereux and Costello (2005) report an experimental demonstration of this "label ambiguity" problem. In their study, subjects were presented with compound nouns and explanatory glosses and asked to select all appropriate relational categories from an inventory of 16. Only 28.2% of compounds were assigned just one relation, and on average compounds were assigned 3.2 relations.

To surmount this problem, the guidelines were refined to guide category selection in cases of doubt and the set of categories was modified. The *MAKE*, *CAUSE* and *USE* relations were replaced by two more general relations *ACTOR* and *INST(rument)* which apply to all compounds describing an event or situation in which the constituents are participants. These new relations also account for most nominalised compounds and many compounds typically analysed as *FOR*. A consequence of this change was that *FOR* itself became redundant and was removed. This may seem surprising, given that *FOR/PURPOSE* is traditionally an uncontroversial entry in compound taxonomies and it is of course the case that many compounds mention the purpose of an item. However, most purpose-expressing compounds also seem to qualify for other relations: *dining room* and *kitchen knife* have strong locative senses, *cheese knife* and *welding iron* are good candidates for *INST* and *mining engineer* and *stamp collector* seem more naturally analysed as *ACTOR*.

[2]These paraphrases of *office chair* are not entirely synonymous; sometimes a chair of the kind typically used in offices will be located in a place that is not an office, and sometimes a chair in an office will not be of the kind typically associated with offices. In the frequent scenario that a typical office chair is used in an office, however, the paraphrase relations will overlap and problems of spurious ambiguity will arise.

[3]For example, in the dataset of Nastase and Szpakowicz (2003) *concert hall* has the label *PURPOSE* but the semantically similar compound *building site* has the label *LOCATION*, while *orchestral conductor* and *blood donor* are *OBJECT* but *city planner* and *teaching professor* are PURPOSE.

I would argue that the purposive aspect of such compounds is not in opposition to what might be called their "core semantics". Rather, it is simply a fact that a compound may have a particular semantics because that semantics captures a salient characteristic of the compound's referent, and this may be due to intentionality, habituality, contrast with other instances of the head noun denotatum, or some other kind of "classificatory appropriateness" in the sense of Zimmer (1971).

An alternative experimental design for annotation and classification would permit the assignation of multiple labels to ambiguous compounds such as *car factory* and *office chair*. This would reduce the burden involved in developing annotation guidelines and grant annotators more freedom to follow their intuitions about multifaceted interpretations. On the other hand the annotators would then have to reason about the degree to which a compound expresses a particular relation, and whether that degree is sufficient to license labelling the compound with that relation. For example, a *bank account* could be construed as containing information relating to the bank with which it is held, but it is not clear whether the *ABOUT* should therefore be applied to this compound. Under a multilabel experimental design, annotation evaluation and classification would most likely be performed in a binary fashion on each relation independently, as is usually done in multilabel document classification tasks. Unless degree information is also provided by the annotators for each relation, the resulting classification system would be unable to answer the important question of which relation is most central to a given compound's meaning. One promising approach to annotation which surmounts some of these complications is that of Nakov (2008), whereby a large set of paraphrases is collected for each compound from non-expert annotators via the Amazon Mechanical Turk website[4] and compiled to create a ranked list of verbal paraphrases. These paraphrases are lexical in nature and not equivalent to the generalised semantic relations considered here, but they are connected to the "integrational representations" discussed in Section 2.3.3.

3.4 The annotation scheme

3.4.1 Overview

The finalised annotation scheme consists of six categories capturing coherent semantic relations (*BE*, *HAVE*, *IN*, *ACTOR*, *INST(rument)*, *ABOUT*), three categories for compounds to which those relations do not apply (*REL*, *LEX*, *UNKNOWN*) and two categories for sequences that are not valid compounds but have been erroneously identified as such by the automatic extraction procedure (*MISTAG*, *NONCOMPOUND*). The correct application of these categories is set out in 12 pages of annotation guidelines, included in this thesis as Appendix B.[5] The first section of the guidelines contains a number of general principles that are not specific to a single relation. These are followed by one or more annotation rules for each category, the full set of which is summarised in Table 3.1. These rules can be viewed as providing a more fine-grained relational annotation; for example, the four rules licensing the *IN* label distinguish between spatial and temporal location and between objects and events as located entities. However, these subrelations

[4] `www.mturk.com`

[5] The guidelines are also available online at `http://www.cl.cam.ac.uk/~do242/guidelines.pdf`.

have not been formulated with the same concern for maintaining clear boundaries as the main categories were, and there is an overlap in some cases.

With the exception of appositions, which are usually symmetric, the head and modifier of a compound can match the arguments of a binary semantic relation in two ways. For example, a *boulder field* is a field in which a boulder or boulders are located, and a *field boulder* is a boulder located in a field. The underlying locative relation or situation is the same in both cases, but the order of the constituents are reversed with regard to the arguments of that relation. This distinction is captured in the annotation scheme by the concept of *directionality*. In addition to the semantic relation it expresses and the annotation rule licensing that relation, each compound is labelled with a marker ("1" or "2") that notes whether or not the order of its constituents matches the order of the argument nouns as stated in the annotation rule. For example, *boulder field* belongs to the category *IN* in accordance with Rule 2.1.3.1 *N1/N2 is an object spatially located in or near N2/N1*; as the order of head and modifier matches the ordering of arguments in the rule, the compound is annotated as IN_1. *Field boulder* is labelled IN_2 by the same rule.

3.4.2 General principles

The first section of the annotation guidelines sets out a number of general principles that are not specific to a single relation. These include the principle that compounds are to be annotated according to their sentential context, and that knowledge about the typical meaning of a compound type is to be relied on only when the context does not disambiguate the semantic relation. There is also a description of the event-participant framework underlying the *ACTOR* and *INST* categories (Section 3.4.6 below).

A fundamental principle of the annotation scheme is that the annotation applies to the semantic relation between the referents of a compound's constituents, not to the referent of the compound as a whole. One implication of this is that exocentric (*bahuvrīhi*) compounds have no special status in the framework – the metaphorical aspect of the compound *bird brain* is in its metonymic application to something which is not a brain rather than in its relational semantics. The relation between *bird* and *brain* here is precisely the same as in a literal application of the compound (when denoting a bird's brain), i.e., a part-whole relation. Exocentric compounds are therefore annotated as any other compound; in the case of *bird brain*, the label *HAVE* is applied.[6] This analysis also accounts for the endocentric example *apple-juice seat*, which was used in a dialogue to denote a seat in front of which a glass of apple juice had been placed (Downing, 1977) and has often been cited as evidence for the trickiness of compound interpretation. The issues raised by this example are denotational – how does a hearer identify the referent of such an unusual compound? On the other hand, its relational semantics are straightforward; *apple-juice seat* is a locative compound expressing a relation very similar to those of *beach house* or *entrance statue*.

The same rationale guides the treatment of compounds that are used as proper names, e.g., *the Law Society*, *the Telecommunications Act*, *Castle Hill*. Again, these have standard compound relational semantics and are different only in the definite reference of the whole

[6]Levi (1978) proposes a similar analysis of exocentric compounds, while excluding them from the main exposition of her theory (p. 6).

Relation	Rule	Definition	Example
BE	2.1.1.1	Identity	*guide dog*
	2.1.1.2	Substance-Form	*rubber wheel*
	2.1.1.3	Similarity	*cat burglar*
HAVE	2.1.2.1	Possession	*family firm*
	2.1.2.2	Condition-Experiencer	*coma victim*
	2.1.2.3	Property-Object	*sentence structure*
	2.1.2.4	Part-Whole	*computer clock*
	2.1.2.5	Group-Member	*star cluster*
IN	2.1.3.1	Spatially located object	*pig pen*
	2.1.3.2	Spatially located event	*air disaster*
	2.1.3.3	Temporally located object	*evening edition*
	2.1.3.4	Temporally located event	*dawn attack*
ACTOR	2.1.4.1	Sentient Participant-Event	*army coup*
	2.1.4.2	Participant-Participant (more prominent is sentient)	*project organiser*
INST	2.1.5.1	Non-Sentient Participant-Event	*cereal cultivation*
	2.1.5.2	Participant-Participant (more prominent is non-sentient)	*foot imprint*
ABOUT	2.1.6.1	Topic-Object	*history book*
	2.1.6.2	Topic-Collection	*waterways museum*
	2.1.6.3	Focus-Mental Activity	*embryo research*
	2.1.6.4	Commodity-Charge	*house price*
REL	2.1.7.1	Other non-lexicalised relation	*fashion essentials*
LEX	2.1.8.1	Lexicalised compound	*life assurance*
UNKNOWN	2.1.9.1	The meaning is unclear	*simularity crystal*
MISTAG	2.2.1.1	Incorrectly tagged	*legalise casino*
NONCOMPOUND	2.2.2.1	Not a 2-noun compound	*[hot water] bottle*

Table 3.1: Summary of annotation rules

compound. It is therefore appropriate to label them with the same categories applied to "common" compounds, e.g., *the Law Society* is labelled *ACTOR*, *the Telecommunications Act* is *ABOUT*, *Castle Hill* is *IN*. It is true that not all naming compounds express a substantial relation between constituents. In cases such as *the Beacon Theatre* and *Corporation Road*, the connection between head and modifier is an arbitrary "naming-after" one. However, this is similar to the relation in common-noun compounds that express a vague relation of association, e.g., *diamond jubilee*. As described in Section 3.4.8, the REL label is given to those compounds and this label is just as applicable to proper names. In contrast, sequences where the head or modifier is in fact a proper noun (*Reagan years*) are not admitted as valid compounds by the annotation scheme and are labelled *MISTAG*.

3.4.3 BE

The label *BE* applies to all relations between nouns N_1 and N_2 that can be paraphrased as N_2 *which is (a)* N_1. This includes the subcategories of appositive compounds (*celebrity winner*, *soya bean*) and material-form compounds (*rubber truncheon*, *ice crystal*). I also include under *BE* compounds describing resemblance, paraphrasable as N_2 *like* N_1 (*cat burglar*, *hairpin bend*). These can be understood as appositive compounds if the comparison noun is understood metaphorically, e.g., *burglar that is a cat* (Levi, 1978).

3.4.4 HAVE

Compounds expressing possession are labelled as *HAVE*. The concept of possession is complex and cannot be reduced to a single simple definition. Langacker (1999) proposes a prototypical model based on "reference point constructions":

> What all possessive locutions have in common, I suggest, is that one entity (the one we call the possessor) is invoked as a reference point for purposes of establishing mental contact with another (the possessed)... And instead of assuming that any one concept (like ownership) necessarily constitutes a unique, clear-cut prototype and basis for metaphorical extension, I propose that the category clusters around several conceptual archetypes, each of which saliently incorporates a reference point relationship: these archetypes include ownership, kinship, and part/whole relations involving physical objects. (p. 176)

Similarly, Taylor (1996) models possession as an "experiential gestalt" with the following prototypical aspects (p. 340):

1. The possessor is a specific human being.
2. The possessed is inanimate, usually a concrete physical object.
3. There is a one-to-many relation between possessor and possessed.
4. The possessor has exclusive rights of access or use regarding the possessed.
5. The possessed has value for the possessor.

6. The possessor's rights arise through a special transaction and remain until transferred through a further transaction.

7. Possession is long-term.

8. The possessed is in proximity to the possessor and may be a regular accompaniment.

The fundamental property shared by Langacker and Taylor's accounts is the asymmetry of the one-to-many relationship between possessor and possessed, and I have taken this property as central to the definition of the *HAVE* category. Annotation rule 2.1.2.1 defines ownership-like possession in terms of properties 3 (one-to-many) and 4 (exclusive use). Compounds expressing mental or physical conditions (*reader mood*, *coma victim*) and properties (*grass scent*, *sentence structure*) are characterised by a one-to-many relationship and also fit intuitively under the *HAVE* category. Likewise, the part-whole relation shares many aspects of possession (as observed by Langacker) and in annotation rule 2.1.2.4 I have adopted the meronymy test of Cruse (1986): "X is a meronym of Y if and only if sentences of the form *A Y has Xs/an X* and *An X is part of a Y* are normal when the noun phrases *an X*, *a Y* are interpreted generically" (p. 160). The part-whole relation is also similar to that between groups and members; hence compounds such as *committee member* and *star cluster* are labelled with *HAVE*.

3.4.5 IN

The label *IN* stands for a unified category of spatial and temporal location. This conflation of space and time is a natural one in view of the strong linguistic and psycholinguistic evidence for a connection in how the two domains are conceptualised (Clark, 1973; Boroditsky, 2000). The individual annotation rules do distinguish between spatial and temporal properties in both the located and location entities, combining to give four finer-grained subcategories.

3.4.6 ACTOR, INST

Many compounds express a relation that is based on an underlying event or situation, and all standard inventories of compound relations provide categories for labelling such compounds. However, the inventories differ as to how the set of event-based compounds is to be subdivided. As noted in Section 3.2, the schemes of Levi (1978) and Lauer (1995) do not assign semantic categories to nominalised compounds, instead either assigning them to distinct categories or discarding them completely. Non-nominalised compounds with an event-based meaning and participant-denoting constituent(s) can be labelled *FOR*, *MAKE*, *CAUSE* or *FROM* in Levi's scheme. Other schemes, such as those of Nastase and Szpakowicz (2003) and Girju (2007a) provide a large number of fine-grained categories reflecting the precise nature of the event and its participants.

The analysis I have adopted defines just two categories, *ACTOR* and *INST(rument)*. The distinction between the categories is based on the concept of *sentience*. Thus a *student demonstration* is labelled *ACTOR* as the participants mentioned are sentient, while a *production line* is labelled *INST* as lines are not sentient. The guidelines state two sufficient conditions for sentience: membership of the animal kingdom is one, the other

requires that an entity be a group of people or an organisation (e.g., *research university*, *manufacturing company*). Sentience is very similar, and arguably identical, to the concept of animacy, which is recognised as a fundamental semantic category across languages (see Zaenen et al. (2004) and the references therein) and has been implicated as playing a role in compound interpretation (Devereux and Costello, 2007). I have preferred to use the term sentience to emphasise that the concept of interest also applies to organisations and hypothetical thinking robots.

The sentience criterion is sufficient to account for compounds that mention an event and a single participant. However, more than half of event-based compounds mention two participants rather than explicitly mentioning the event, for example *school leaver*, *music group*, *air filter*.[7] A minor problem that arises with these examples is how to decide on the directionality of *ACTOR* and *INST* instances. If this were the only problem here, it could be solved simply but inelegantly by ignoring the asymmetry of *ACTOR* and *INST* and thus not annotating these relations for directionality. However, a greater problem is encountered in compounds that mention a sentient participant and a non-sentient participant, e.g., *bee honey*, *honey bee*, *infantry rifle*. One solution, also simple but inelegant, would be to label according to the sentience of the head noun, but this distinction would not reflect the semantics of the underlying event and would introduce a spurious asymmetry in the relational semantics of pairs such as *bee honey* and *honey bee*. I have based my guidelines for these cases on a hierarchy of semantic roles which is informed by Talmy (2000). The roles I assume are Agent, Instrument, Object and Result; they are defined in terms of a flow of "energy" or "force" originating from the Agent and culminating in the Result (if the event has a Result role). The hierarchy of roles is from most to least agentive:

$$\text{Agent} > \text{Instrument} > \text{Object} > \text{Result}$$

The distinction between *ACTOR* and *INST* compounds is thus based on the sentience of the more agentive participant: where the more agentive constituent is sentient *ACTOR* applies and where the more agentive constituent is non-sentient *INST* applies. Furthermore, the same criterion is used to assign directionality: the relation expressed by bee_A $honey_R$ is $ACTOR_1$ while that expressed by $honey_R$ bee_A is $ACTOR_2$. To reduce the burden on annotators and the probability of confusion (it is often difficult to tell Instruments from Objects), the annotation scheme does not require that the constituents of event-based compounds be annotated for their thematic roles. Only a decision about relative agentivity is necessary to assign the correct relation and directionality.

The relatively coarse granularity of these relations has the advantage of avoiding sparsity and reducing the need to define boundaries between categories that often overlap, such as causes, producers, agents and instruments. On the other hand, there is clearly a loss of information in the annotations; *headache pill* and *fertility pill* receive the same label (*INST*) despite their semantic differences, and *museum curator* and *bee sting* are both labelled *ACTOR* (though with different directionality markers).

[7] In my annotated sample of 2,000 compounds, 232 items were annotated with Rules 2.1.4.1 or 2.1.5.1 (Participant-Event) while 269 were annotated with Rules 2.1.4.2 or 2.1.5.2 (Participant-Participant).

3.4.7 ABOUT

The inclusion of a topicality relation is uncontroversial, as it occurs in all standard compound relation inventories. The prototypical instance involves an entity with descriptive, significative or propositional content, in other words an item that is *about* something. This is commonly a speech act or a physical manifestation of a speech act (*unification treaty*, *history book*, *drugs charge*) but also extends to non-verbal means of communication (*wiring diagram*, *love scene*). A special provision is made in the guidelines for entities which may not have descriptive/significative/propositional content themselves but rather house or contain entities with such content. This applies to museums and exhibitions (*waterways museum*) and to educational courses (*publishing course*).[8]

Also falling in the *ABOUT* category is the class of compounds expressing the focus of a mental state or an activity, e.g., *exam practice*, *siege mentality*, *pollution problem*, *cup match*. These are cases where the criterion of topical content does not hold but there is nonetheless a strong sense of "aboutness". The final non-archetypal subclass of *ABOUT* deals specifically with prices and charges, e.g., *house price*, *recording cost*, *case discount*. These are held to belong to the *ABOUT* category because prices are abstract entities that signify the amount of commodities that must be exchanged with regard to services or goods, as stipulated by an entitled party such as the seller or the government. Thus there is a fine line distinguishing prices from inherent properties of entities (*HAVE*, Rule 2.1.2.2) on one side and from the physical commodities exchanged in a commercial transaction (*INST*, Rule 2.1.5.2) on the other.

3.4.8 REL, LEX, UNKNOWN

Not all compound nouns can be classified with one of the six relations described above. Some compounds seem to encode a general sense of association rather than a specific semantic relation, for example *fashion essentials* (*essentials associated with fashion*) or *trade purposes* (*purposes associated with trade*). Others do have relational content, but the relation in question is distinct from the six named relations. Examples are found in the names of chemical compounds (*lithium hydroxide*, *calcium sulphide*) and names coined after the pattern N_2 *named after* N_1 (e.g., *diamond jubilee*, *Beacon Theatre*). These are not necessarily lexicalised compounds, as they are generated by productive patterns. However, it is impractical and less than fruitful to attempt to account for all possible relations, and the use of an "other" category such as *REL* is unavoidable.

The label *LEX* is applied to lexicalised or idiomatic compounds. These are compounds that cannot be understood by analogy or other standard strategies for compound interpretation. The relation between the constituents is not clear if the compound has not been encountered before, and it is often assumed that lexicalised compounds are listed in a speaker's lexicon.[9] It therefore seems appropriate that they be assigned to a distinct category. Defining a general concept of lexicalisation is not straightforward, nor

[8]This rule (2.1.6.2) can be seen as a clarification of the previous rule (2.1.6.1), rather than as an alternative semantic category.

[9]The tendency to lexicalisation is one that compounds share with other classes of multiword expressions (MWEs; Sag et al., (2002)). Morphosyntactic idiosyncracy is a second typical property of MWEs that can be observed in compounds: it is well-known that non-head constituents in English compounds are in general not inflected, i.e., we say *mouse trap* and *window cleaner* but not **mice trap* or **windows cleaner* (though there are many exceptions such as *arms race* and *trades unions*). On the other hand,

is disentangling it from related concepts such as non-compositionality, opacity and conventionality (Nunberg et al., 1994; Keysar and Bly, 1995). Here I describe the class of lexicalised compounds in terms of *semantic substitutability*, which has proven useful in computational research on idiomatic multiword expressions (Lin, 1999; McCarthy et al., 2003; Fazly and Stevenson, 2006) and has been proposed as a test for compound lexicalisation by Lehnert (1988).[10] The substitutability criterion states that a lexicalised compound is one whose semantics does not follow an analogical pattern; other compounds formed by substituting one of its constituents with a semantically similar word do not have a similar relational meaning. For example, the compound *monkey business* has an idiomatic meaning that is not shared by the lexically similar compounds *ape business* or *monkey activity*.

Finally, the scheme contains an *UNKNOWN* category for compounds which the annotator is unable to interpret. Sometimes this occurs because the compound is a technical term; other times the sentential context is insufficient to deduce the meaning.

3.4.9 MISTAG, NONCOMPOUND

Methods for automatically extracting compounds from corpora will not be error-free, and the annotation scheme makes provision for labelling items that are in fact not compounds. The *MISTAG* label is applied when one or more constituents should not have been tagged as common nouns. The *NONCOMPOUND* label is used when the constituents have been tagged correctly but do not constitute a true two-noun compound. This can occur when they are part of a larger noun phrase or when they appear adjacent in the context for reasons other than compounding. These classes of errors are described more fully in Section 4.2.

3.5 Conclusion

The desiderata listed in Section 3.2 offer a means of evaluating the new annotation scheme that I have introduced above. To estimate its coverage we require an annotated sample of compound data. Table 3.2 gives the distribution of categories over the sample of 2,000 compounds used in the machine learning experiments of Chapters 6 and 7. 92% of the valid compounds (i.e., of all those not labelled *MISTAG* or *NONCOMPOUND*) are assigned one of those six relations, which are the ones I use in the classification experiments. This indicates that the annotation scheme has good coverage. The generalisation criterion is satisfied as many of the category definitions are based on general linguistic principles such

compounds seem to be unlike other kinds of MWEs in allowing the productive combination of open-class words to express non-idiomatic semantic relations.

[10] Bannard et al. (2003) raise an issue that is problematic for corpus-based work relying on the substitutability criterion. The fact that lexically similar neighbours of a multiword expression do not appear in a corpus, or only appear with very different distributional properties, is not necessarily a sign of semantic lexicalisation. Rather, it may be due to facts about the world (the entities described by lexically similar terms do not exist) or about the language (previously established synonyms may block the use of multiword expressions); the example given by Bannard et al. is *frying pan*. However, this phenomenon is not problematic for the use of substitutability as an annotation criterion: an annotator does not need to ask whether a neighbour of a given compound is likely to be used, but rather whether the neighbour, if it were used, would have a similar relational meaning.

Relation	Distribution	
BE	191	(9.55%)
HAVE	199	(9.95%)
IN	308	(15.40%)
INST	266	(13.30%)
ACTOR	236	(11.80%)
ABOUT	243	(12.15%)
REL	81	(4.05%)
LEX	35	(1.75%)
UNKNOWN	9	(0.45%)
MISTAG	220	(11.00%)
NONCOMPOUND	212	(10.60%)

Table 3.2: Class frequencies in a sample of 2,000 compounds

as thematic roles, semantic substitutability and the event/object distinction. I have made the annotation guidelines accompanying the scheme publicly available to allow comparison and adoption by other researchers. In the next chapter I describe a multiple-annotator experiment that tests the coherence of the annotation categories and usability of the guidelines.

Chapter 4

Evaluating the annotation scheme

4.1 Introduction

This chapter describes an experiment in evaluating the annotation scheme developed in Chapter 3 on real data extracted from corpus text. I describe the construction of the dataset in Section 4.2 and the evaluation procedure in Section 4.3. This is followed by agreement results and analysis in Section 4.4. The raw agreement and Kappa scores on the test set of 500 compounds are 66.2% and 0.62 respectively, which compares very favourably to other results reported in the literature. The results underline both the difficulty of the compound annotation task and the need for rigorous annotation scheme development when working with semantic data.

4.2 Data

To compile a corpus of compound nouns I used the written component of the British National Corpus (Burnard, 1995). This source contains approximately 90 million words of text in British English, balanced across genre and type. Although the BNC may be small in comparison with some other corpora used in NLP, it compensates by its balanced nature, which arguably gives a more representative snapshot of the language than larger corpora generated from web or newswire text. The BNC's tractable size also offers the advantage that it can be tagged and parsed without significant resource requirements. In the compound extraction stage, I have used a version of the BNC tagged and parsed using the RASP toolkit (Briscoe and Carroll, 2002).[1] The procedure for identifying compounds is a simple heuristic similar to those used by Lauer (1995) and Lapata and Lascarides (2003). All sequences of two or more common nouns containing alphabetic characters only and flanked by sentence boundaries or by tokens not tagged as common nouns were extracted as candidate compounds. This heuristic excludes compounds written as a single word (*chairlift*, *bookshelf*) and compounds with hyphenated constituents (*bully-boy tactic*, *state decision-making*) as these are difficult to identify reliably and introduce their own particular problems. Applying the heuristic to the BNC produced a collection of almost

[1] This was carried out with the first release of RASP as the second release (Briscoe et al., 2006), which I use in the experiments of Chapters 6 and 7, was not yet available.

Length	Token Frequency	Type Frequency	Tokens/Types
2	1,590,518	430,555	3.7
3	142,553	96,013	1.5
4	11,348	9,635	1.2
5	1,074	925	1.2
6	129	113	1.1
7	30	29	1.0
8	1	1	1.0
9	3	3	1.0
>9	3	3	1.0
Total	1,745,659	537,277	3.2

Table 4.1: Distribution of noun sequences in the BNC

1.75 million candidate compound instances, with 537,277 unique types;[2] the distribution of tokens and types by length is given in Table 4.1. Very few of the longest extracted candidates are well-formed compounds; most include a mistagged word or proper name, or are just unstructured lists of nouns. The longest sequences satisfying my working definition of compoundhood contain six words, e.g., *metal oxide semiconductor field effect transistor* and *farm management microcomputer software service project*.

In the case of two-noun compounds, which are the focus of the experimental work in this thesis, there are 1,590,518 tokens and 430,555 types. The average token/type ratio is 3.7, corresponding to one appearance every 24.3 million words. However, the frequency spectrum of compounds exhibits the Zipfian or *power-law* behaviour common to word frequency distributions and many naturally occurring phenomena (Baayen, 2001; Newman, 2005).[3] The number of types observed with a certain frequency drops off rapidly for frequencies above 1, but there is nonetheless a "long tail" of tokens with frequency much higher than the average. In the two-noun compound data there are 296,137 types that occur once and just 2,514 types that occur 10 times, but 81 types that occur 500 times or more.[4] Figure 4.1 is a log-log plot of the frequency spectrum for two-noun compounds; the straight line plot that is characteristic of power laws is visible up to a point where sparsity makes the plot rather scrambled.

The simple method I have used to extract compounds is not error-free. It has a non-negligible false positive rate, whereby sequences are incorrectly identified as compound nouns. This can arise due to errors in part-of-speech tagging, both when constituents are falsely tagged as common nouns (*Reagan* in *the Reagan years*, *play* in *what role does the king play*) and when adjacent words are falsely tagged as non-nouns (*Machine* in *Machine Knitting Course* is tagged as a proper noun).[5] These cases are accounted for in

[2]Even allowing for extraction error, this suggests that close to 3% of all words in the BNC are constituents of a noun-noun compound.

[3]Power-law distributions are so-called because they have the form $P(X = k) \propto k^{-\alpha}$, where the exponent α determines the drop-off rate of the probability curve.

[4]The most frequent compounds are *interest rate* (2901), *world war* (2711), *subject area* (2325), *trade union* (1843) and *health service* (1824). This ranking is certainly affected to some degree by the selection of documents in the BNC (a "small sample" effect). Comparing Google counts, *world war* (94.5 million) is around 2.5 times more frequent than *interest rate* (38.1 million). However, the general power-law behaviour is expected to hold for larger corpora also, just as it does for unigram frequencies (Baayen, 2001).

[5]Briscoe and Carroll (2002) report "around 97%" tagging accuracy for RASP. This figure is comparable

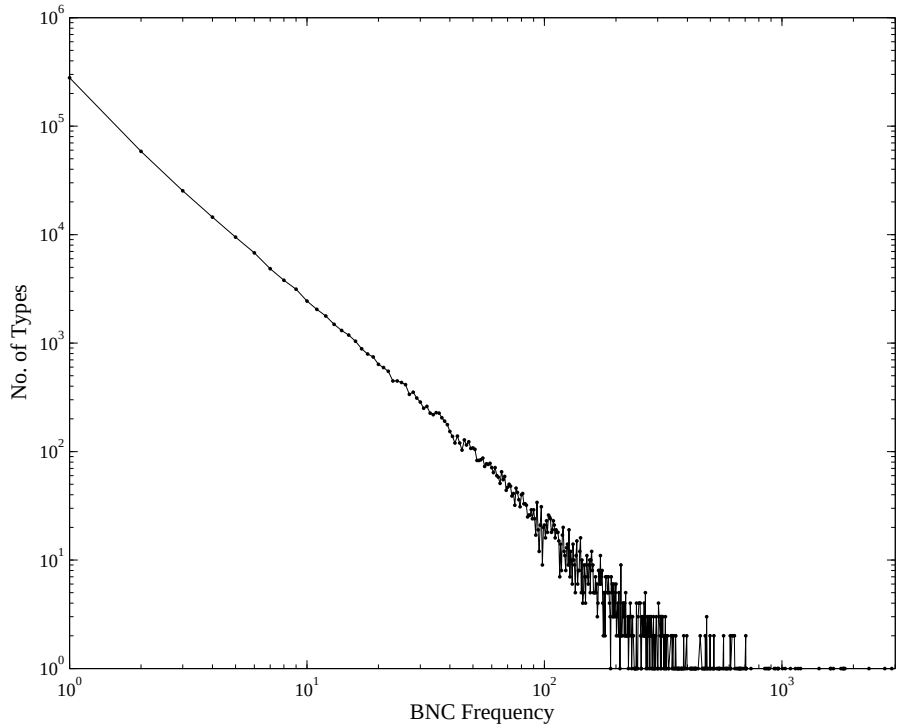


Figure 4.1: Log-log plot of the frequency spectrum for compounds of length 2

my annotation scheme by the *MISTAG* and *NONCOMPOUND* labels respectively. Even when the tagging is correct, false positives can arise because one constituent is in fact part of a larger noun phrase (*[scene of crime] officer*, *[stained glass] window*) or because the two nouns occur together for reasons other than compounding (*the skills people have*, *However in practice deterioration of stock inevitably sets in*). These are also labelled *NONCOMPOUND*. Using the frequencies for the *MISTAG* and *NONCOMPOUND* labels given in Table 3.2, we can estimate a false positive rate of 21.6% for my heuristic on the BNC.

False negatives also arise, whereby sequences that are in fact compounds are not extracted. For example, the modifier *orange* in the sequence *orange juice* is tagged as a common noun 44 times in the corpus but is tagged as an adjective 150 times. Most of the latter seem to be tagging errors, thus excluding them incorrectly from the collection of compounds. It is more difficult to estimate false negative rates than false positive rates, as the true set of compounds in a corpus is unknown. Also, false negative errors cannot be corrected at the manual annotation stage as false positives can. The only study I am aware of to address the issue of false negatives is that of Lapata and Lascarides (2003) discussed below, but they limited themselves to a sample of sequences tagged as nouns and did not consider the mistagged-constituent factor.

Lauer (1995) reports that his version of the heuristic attained 97.9% precision on a sample of 1,068 candidate compounds extracted from the Grolier Multimedia Encyclopedia. However, this figure was achieved by using a dictionary of words which can only be nouns as a proxy for part-of-speech tagging, leading to a presumably large number of false neg-

to state-of-the-art results in part-of-speech tagging (Shen et al., 2007), suggesting that the choice of tagger did not have a detrimental effect on compound extraction performance. That the RASP tagger has been trained on general British English text should also be an advantage when tagging the BNC.

atives. Lapata and Lascarides (2003) apply a method similar to mine to a tagged version of the BNC and report a true positive rate of 71% and a true negative rate of 98.8%; the latter is probably inflated due to the consideration of sequences tagged as nouns only. The estimate of 78.4% precision for my heuristic is slightly better; this method is stricter than that of Lapata and Lascarides due to the restriction that constituents must contain only alphabetic characters. Lapata and Lascarides also describe how compound identification can be improved through statistical measures, but this has not been investigated in this work as the simpler heuristic seems sufficient.

4.3 Procedure

Two annotators were used – the present author (Annotator 1) and an annotator experienced in lexicography but without any special knowledge of compounds or any role in the development of the annotation scheme (Annotator 2). Both are native speakers of English. The distance of the second annotator from the development phase is important as her judgements should be based only on the text of the annotation guidelines and a small amount of clarificatory email correspondance, not on shared knowledge that might have emerged during development but is not explicitly included in the guidelines. This adds rigour to claims of reproducibility regarding our agreement results.

Each compound was presented alongside the sentence in which it was found in the corpus. Each annotator labelled it with the appropriate semantic category, the rule licensing that label in the annotation guidelines, and the order of compound constituents with regard to the argument slots in that rule (directionality). The following is a representative example:

```
483883: air disaster
IN,2,2.1.3.2
In the country 's fifth air disaster in four months ,
the China Southern Airlines plane crashed as it
approached to land at the city of Guilin

|In_II| |the_AT| |country_NN1| |'s+_$| |fifth_MD|
|air-disaster_QNN1| |in_II| |four_MC| |month+s_NNT2| |,_,|
|the_AT| |China_NP1| |Southern_JJ| |Airline+s_NN2|
|plane_NN1| |crash+ed_VVN| |as_CSA| |it_PPH1|
|approach+ed_VVD| |to_TO| |land_VV0| |at_II| |the_AT|
|city_NN1| |of_IO| |Guilin_NN1|
```

Here the annotation states that the category is IN, it is *a disaster in the air* not *air in a disaster* and that the licensing rule is 2.1.3.2 *N1/N2 is an event or activity spatially located in N2/N1.*

I used a set of 2,000 compounds for my annotation and classification experiments. These were sampled randomly from the corpus of compounds extracted in Section 4.2, with the constraint that no compound type could occur more than once. Two trial batches of 100 compounds each were annotated to familiarise the second annotator with the guidelines and to confirm that adequate agreement could be reached without further revisions of the annotation scheme. The first trial resulted in agreement of 52% and the second in

agreement of 73%. The result of the second trial, corresponding to a Kappa beyond-chance agreement estimate (Cohen, 1960) of 0.693, was very impressive and it was decided to proceed to a larger-scale task.[6] 500 compounds not used in the trial runs were drawn from the 2,000-item set and annotated. As the data contained many rare and technical terms, the annotators were permitted to make use of resources including Google, the Oxford English Dictionary and Wikipedia so that the task would not be compromised by an inability to understand the data. The second annotator reported that the first 100 compounds took seven hours to annotate (4.2 minutes/compound), while the second trial batch took five hours (3.0 minutes/compound) and the 500-item test set took 15.5 hours (1.9 minutes/compound). This indicates that an initially untrained annotator can boost his/her labelling speed quite quickly through practice.

4.4 Analysis

4.4.1 Agreement

Agreement on the 500-item test set was 66.2%. corresponding to a Kappa score of 0.62. This is lower than the result of the second trial annotation, but may be a more accurate estimate of the "true" population Kappa score due to the larger sample size.[7] On the other hand the larger task size may have led to a decrease in agreement, as the test set annotation had to be done over the course of multiple days and inconsistencies may have resulted – the second annotator has endorsed this suggestion.

The granularity of the agreement analysis can be refined by considering the directionality and rule information included in the annotations. Agreement on category and directionality (order of the compound constituents with regard to the arguments listed in the rule) is similar to agreement on categories alone at 64% (Kappa = 0.606). Agreement on the 25 rules licensing category assignment is lower at 58.8% (Kappa = 0.562) but it should be borne in mind that the guidelines were not developed with the intention of maximising the distinctions between rules in the same category.

[6]Kappa is the most widely-used estimate of beyond-chance agreement but its correct application remains the subject of some controversy. A number of absolute scales have been proposed for its interpretation, but these scales are frequently contradictory and do not allow for task-specific factors or sampling variation (Di Eugenio, 2000; Krenn and Evert, 2004). For example, the fact that annotators often disagree about the basic meanings of compounds means this task will have a lower ceiling on possible agreement than tasks such as part-of-speech tagging. Rather than referring to a universal scale, it may be more informative to compare agreement results to other work on the same problem, as I do in Section 4.5. A second criticism of the use of Kappa in multiclass experiments is that presenting a single agreement figure is insufficient to describe the multiple interactions between data and annotators (Byrt et al., 1993; Kraemer et al., 2004). To address this, I present analyses of per-category and inter-category agreement alongside raw agreement and Kappa figures, following the recommendations of Kraemer et al.

[7]It is possible to calculate a confidence interval for the population Kappa value using a method described in Fleiss et al. (2003). The 95% confidence interval for the 100-item second trial set is 0.69±0.11, while that of the 500-item test set is a narrower 0.62±0.05. The calculation of these intervals is dependent on a large-sample normality assumption and its accuracy on small datasets is questionable; more accurate methods have been proposed, but they are restricted to the case of binary categories (Lee and Tu, 1994; Blackman and Koval, 2000). If sufficient resources were available, a more concrete evaluation of the reproducibility of the agreement figure for this dataset could be provided by a repetition of the annotation task at a subsequent time or the annotation of the data by a third annotator.

Unlike most other studies of compound annotation, this annotation task requires the annotator to distinguish syntactically valid compounds from non-compounds and lexicalised compounds from non-lexicalised ones in addition to assigning semantic relations to non-anomalous data items. To get a rough estimate of agreement on the six "semantic" categories that would be used in the classification experiments (*BE*, *HAVE*, *IN*, *ACTOR*, *INST*, *ABOUT*) and to aid comparison with studies that use cleaner pre-filtered data, an analysis was carried out using only those items which both annotators had labelled with one of those categories. This left 343 items with agreement of 73.6% and Kappa = 0.683. Of course, this is not a perfect estimate of agreement on these categories as it excludes items which one annotator labelled with a semantic category and the other did not but may have done if the alternative "non-semantic" categories were not available.

97 of the 500 test items occur just once in the BNC. If these mainly consist of unusual or technical terms, or novel compounds coined for a particular discourse context, we might expect lower agreement than on more frequent compounds. On the other hand, single-instance sequences are more likely to be extraction errors labelled with *MISTAG* or *NONCOMPOUND*, and agreement is relatively high on these labels (Section 4.4.2).[8] Splitting the test data into two sets of frequency = 1 and frequency > 1 and calculating separate agreement scores for each yields agreement of 61.9% (Kappa = 0.557) in the first case and 67.2% (Kappa = 0.630) in the second. When only the six "semantic" categories are considered, agreement is 64.3% (Kappa = 0.557, sample size = 43) for single-occurrence compounds and 74.9% (Kappa = 0.70, sample size = 291) for frequency > 1. This evidence is not conclusive due to the small sample sizes, but it certainly suggests that the productivity of compounding contributes to the difficulty of the annotation task. It also indicates that annotation results will be strongly influenced by the method used to sample data. If data is extracted from a dictionary, common compounds are likely to be oversampled and agreement will be higher. If data is sampled from a corpus with equal probability given to each compound type, rare compounds will be oversampled and agreement will be low. In this study data has been sampled from the BNC with probability proportional to corpus frequency (with a type constraint to prevent repeated types inflating agreement), and I would argue that this leads to a realistic estimate of agreement on the task of annotating compounds.

4.4.2 Causes of disagreement

It is interesting to investigate which categories caused the most disagreement, and which inter-category boundaries were least clear. One simple way of identifying category-specific differences between the annotators is to compare the number of items each annotator assigned to each category; this may indicate whether one annotator has a stronger preference for a given category than the other annotator has, but it does not tell us about actual agreement. One-against-all agreement scores and the corresponding Kappa values can highlight agreement problems concerning a single category C by measuring agreement on the binary task of classifying the data as either C or not-C (i.e., as belonging to one of the other categories). These measures are given for the test data in Table 4.2. The most striking disparities in the per-category counts show a bias for *INST* on the part of Anno-

[8]The distribution of the full 2000-item annotated dataset indicates that 40% of sequences with frequency 1 are labelled *MISTAG* or *NONCOMPOUND*, in contrast to 17.7% of more frequent sequences. This difference is confirmed to be significant by a χ^2 test ($p = 1.6e^{-10}$).

Category	Ann. 1	Ann. 2	Agreement	Kappa
BE	52	63	0.926	0.637
HAVE	59	77	0.888	0.525
IN	69	66	0.930	0.700
INST	73	42	0.902	0.523
ACTOR	52	48	0.948	0.711
ABOUT	55	83	0.908	0.616
REL	19	20	0.930	0.066
LEX	11	8	0.974	0.303
UNKNOWN	3	3	0.988	-0.006
MISTAG	57	52	0.966	0.825
NONCOMPOUND	50	38	0.964	0.776

Table 4.2: Per-category assignments for each annotator and one-against-all agreement measures

tator 1 and a bias for *ABOUT* on the part of Annotator 2; there are also smaller biases regarding *BE*, *HAVE* and *NONCOMPOUND*. The raw one-against-all agreement figures are universally high. This is not surprising as when there are many categories with a relatively balanced distribution, for any category C the majority of items will be clear-cut cases of the non-C category. More informative are the one-against-all Kappa values, which show agreement above 0.7 for *IN*, *ACTOR*, *MISTAG* and *NONCOMPOUND*, agreement close to 0.5 for *HAVE* and *INST*, and extremely low agreement (below 0.1) for *REL* and *UNKNOWN*.

Studying agreement between pairs of categories can explain which kinds of compounds are most difficult to label and can suggest where the annotation guidelines are in need of further refinement. Standardised Pearson Residuals (Haberman, 1973) give a chance-corrected estimate of between-category agreement. These residuals are defined on a confusion matrix or contingency table of assignments and the residual e_{ij} for two categories i and j is given by

$$e_{ij} = \frac{n_{ij} - \hat{\mu}_{ij}}{[\hat{p}_{i+}\hat{p}_{+j}(1-\hat{p}_{i+})(1-\hat{p}_{+j})]^{\frac{1}{2}}}$$

where n_{ij} is the observed value of cell ij and $\hat{p}_{i+}$, $\hat{p}_{+j}$ are row and column marginal probabilities estimated from the data. Intuitively, this residual compares the proportion of data items assigned by Annotator 1 to category i and by Annotator 2 to category j with the expected proportion given Annotator 1's overall proportion of assignments to i and Annotator 2's overall proportion of assignments to j, normalised by a variance term. The resulting table of residuals is therefore not symmetric, $e_{ij} \neq e_{ji}$. In the context of an annotation experiment it is expected that the observed data will diverge strongly from independence, giving large positive values on the same-category diagonals and negative off-diagonal values. Problematic boundaries can be identified where this pattern is not observed.

Residuals for the experimental results are given in Table 4.3. There are clear problems with *REL*, *LEX* and *UNKNOWN*, precisely because the borders of these categories are very difficult to pin down. In the case of *UNKNOWN* disagreement is unavoidable as different annotators will bring different background knowledge to the task and some annotators may be more willing than others to assign a possible relation in doubtful cases.

The only off-diagonal positive residual among the six semantic relations is between *INST* and *ABOUT*. Inspection of the data shows that this is due to a set of items such as *gas alarm* which can justifiably be interpreted as both an *alarm activated by the presence of gas* (*INST*) and an *alarm signalling the presence of gas* (*ABOUT*). In these cases Annotator 1 tended to assign *INST* and Annotator 2 tended to assign *ABOUT*. The low one-against-all Kappa score for *HAVE* seems to arise mainly from an interaction with *REL*; many of the problematic items here are borderline properties such as *pay rate* and *resource level*. Adding further examples to the annotation guidelines should clarify these cases. On the other hand, many disagreements fall into other patterns that are not common enough to show up in this analysis and thus constitute a "long tail" for which the provision of exhaustive guidelines is not practically feasible.

A different perspective on observed disagreements can be obtained through a qualitative analysis of the reasons why annotators give different labels to a data item. In some cases, one annotator simply makes a mistake; in others, the annotation guidelines are unclear; in others, there is genuine disagreement about the meaning of the compound. The distribution of these factors can inform us of the genuine upper bound that can be achieved even with a perfect annotation scheme and error-free annotators, and of the degree to which agreement could be improved by further refining the guidelines. To this end, a classification of disagreement types was produced and all disagreements in the annotated test corpus were attributed one of these types. In many cases the reason for disagreement was clear from the data; if not, it was identified by consultation among the annotators. The classification used and distribution of types were as follows:

1. True disagreement about the referent of the compound (10.1%). Examples are *peat boy*, which one annotator understood as a *boy who works with or sells peat* and the other understood as a *boy buried in peat*, and *school management*, which was understood both as the *personnel who manage the school* and as the *activity of managing the school*. It is possible that the number of such disagreements could be reduced by providing more context to the annotators, but they cannot be avoided completely.

2. Agreement about the compound referent, but disagreement about the relation between the nouns (20.1%). This often results from disagreement about the meaning of one of the compound's constituents; a *video phone* may be interpreted as a *phone that plays video (information)* (*INST*) or as a *phone that is also a video (player)* (*BE*), though both interpretations allow the compound to denote the same set of devices.[9] Likewise *sponsorship cash* can be *cash gained through sponsorship* (*INST*) or *sponsorship in the form of cash* (*BE*). Annotation practice for some recurring compound classes of this type could be stipulated in the guidelines, but it is probably impossible to produce an exhaustive listing that would eliminate all disagreements.

3. Disagreement about part of speech or bracketing, whereby both analyses are plausible (11.8%). Examples are *mass death* (*mass* could be adjective or noun) and *new technology applications* (*applications of new technology* or *technology applications which are new*). These disagreements are unavoidable where noisy data is used.

[9]There are many phenomena in natural language which exhibit clear ambiguity but do not usually lead to misunderstandings or breakdown in dialogue. Similar observations have been made about syntactic sentence structure by Poesio (1996) and Sampson and Babarczy (2006) and about "sloppy" anaphoric reference by Poesio et al. (2006).

Ann. 2 / Ann. 1	BE	HAVE	IN	ACTOR	INST	ABOUT	REL	LEX	UNK	MIS	NON
BE	14.32	-1.63	-2.54	-2.48	-1.78	-2.22	-1.56	**0.20**	-0.59	-1.64	-1.63
HAVE	-1.02	11.87	-1.14	-1.72	-1.98	-3.28	**1.87**	-1.04	-0.64	-2.79	-2.35
IN	-3.01	-0.58	15.66	-2.04	-2.71	-2.95	**0.16**	-0.11	-0.70	-3.05	-2.57
ACTOR	-1.57	-1.63	-2.54	15.92	-2.31	-2.61	**0.69**	-0.97	**3.20**	-2.60	-2.18
INST	-0.84	-1.14	-1.36	-3.01	12.27	**0.64**	-0.59	-0.17	-0.72	-2.32	-2.65
ABOUT	-2.98	-2.56	-2.64	-1.59	-2.38	14.16	-0.88	**0.14**	-0.61	-2.68	-1.18
REL	-0.98	**0.05**	-1.73	-1.45	**1.18**	**3.05**	1.48	**1.30**	-0.35	-0.75	-1.27
LEX	**0.56**	**0.26**	-0.41	-1.09	-1.02	-0.68	**0.87**	6.86	-0.26	-0.14	-0.96
UNK	-0.66	**0.86**	-0.68	-0.57	**1.56**	-0.78	**2.60**	-0.22	-0.13	-0.59	-0.50
MIS	-1.77	-3.03	-2.71	-1.18	-1.92	-3.58	-0.92	-1.02	**1.20**	18.47	-2.30
NON	-1.93	-1.94	-2.91	-1.42	-1.18	-1.32	-0.76	-0.95	-0.58	-2.54	17.55

Table 4.3: Standardised Pearson Residuals for the annotated test set; off-diagonal positive values are in bold

4. Mistakes: one annotation clearly contradicts the guidelines and no reasonable explanation can be given for the annotation (8.9%). Examples found in the test data are *cat owner* (annotated as *ACTOR*, should be *HAVE*), *credit facility* (annotated as *ABOUT*, should be *INST*) and *pearl brooch* (annotated as *BE*, in context this is part of the phrase *mother of pearl brooch* and should be *NONCOMPOUND*). As might have been expected, the majority of mistakes (but not all) were made by the annotator with less experience of the annotation scheme (Annotator 2).

5. Vague guidelines: there is probably agreement on the meaning of the compound but uncertainty about category boundaries leads to disagreement (20.7%). Many of these cases lie on the *INST*/*ABOUT* borderline discussed above. Others relate to vagueness in the distinction between common and proper nouns; one annotator labelled both *Christmas cake* and *Palace player* (*Palace* denoting Crystal Palace football club) as *MISTAG* while the other assigned *IN* and *REL* respectively, and the guidelines did not specify the correct annotation.

6. There is no evidence of disagreement about the compound's meaning but at least one annotator has assigned one of the categories *REL*, *LEX* and *UNKNOWN* (28.4%). As observed above, these categories are especially problematic. As they apply when no other category seems appropriate, some disagreements of this type could be reduced by clarifying the boundaries of the other categories. For example, disagreement about *football enthusiast* (one annotator has *ACTOR*, the other *REL*) and about *pay rate* (*HAVE* versus *REL*) might be avoided by improving the definitions of *ACTOR* and *HAVE* respectively. On the other hand, it is harder to solve the problem of distinguishing lexicalised compounds from non-lexicalised. The substitutability criterion used in the guidelines for *LEX* functions well much of the time, but different annotators will have different intuitions about substitutability and disagreements may be inevitable. Examples found in the test data include *platform game*, *rugby league* and *trace element*. As previously noted, the *UNKNOWN* category will always be likely to cause disagreements, though the overall number of assignments to this category might be reduced by the provision of more context.

It has been argued that for part of speech annotation (Babarczy et al., 2006) and for syntactic annotation of sentences (Sampson and Babarczy, 2006), the abilities of annotators to follow guidelines contribute more to annotation disagreements than imprecision in those guidelines does. Those studies use a highly-refined exhaustive set of annotation guidelines and expert annotators, so their results will be more conclusive than ones drawn from the current study. However, the breakdown of disagreement types presented here does suggest that even with a rigorously developed annotation scheme the division of responsibility is less clear in the case of compound semantics. If we attribute all cases of true disagreement and all mistakes (categories 1–4) to annotator issues, 50.86% of disagreements can be thus accounted for. Perhaps some of these could be resolved by expanding the guidelines and providing more context around the compound to the annotators. However, there are only a few obvious cases where a change in the guidelines would make a significant difference to the agreement rate. All category 5 cases (20.7%) are due to the annotation guidelines. It is less clear how to analyse the category 6 cases, which relate to the *REL*, *LEX* and *UNKNOWN* categories. In many of these, the annotators may agree on the compound semantics but be unclear whether or not it fits into one of the six semantic categories, or whether or not it is lexicalised. This suggests that the problem lies with the guidelines,

but beyond certain common disagreement types, it will be difficult to solve. The conclusion drawn from this analysis is that it may not be practically feasible to develop an annotation scheme for compound relations with the same precision as has been achieved for syntactic annotation tasks.

4.5 Prior work and discussion

This work appears to be the first reported study of annotating compounds in context.[10] This aspect is important, as in-context interpretation is closer to the way compounds are used and understood in the real world, and compound meanings are often context-dependent.[11] It is not clear whether in-context or out-of-context interpretation is easier, but they are indeed distinct tasks. Out-of-context interpretation relies on a compound having a single most frequent meaning and where this holds agreement should be higher. In-context interpretation allows even improbable interpretations to be considered (a *fish knife* could be a *knife that looks like a fish*) and where the intended meaning is not fully explicit in the context annotators may vary in their willingness to discard the most frequent meaning on the basis of partial evidence.

Some authors of compound annotation schemes and compound datasets do not describe any measurement of inter-annotator agreement, notably Lauer (1995) and Nastase and Szpakowicz (2003). Other authors have given out-of-context agreement figures for corpus data. Kim and Baldwin (2005) report an experiment using 2,169 compounds taken from newspaper text and the categories of Nastase and Szpakowicz. Their annotators could assign multiple labels in case of doubt and were judged to agree on an item if their annotations had any label in common. This less stringent measure yielded agreement of 52.3%. Girju et al. (2005) report agreement for annotation using both Lauer's (1995) 8 prepositional labels (Kappa = 0.8) and their own 35 semantic relations (Kappa = 0.58). These figures are difficult to interpret as annotators were again allowed to assign multiple labels (for the prepositions this occurred in "almost all" cases) and the multiply-labelled items were excluded from the calculation of Kappa. This entails discarding the items which are hardest to classify and thus most likely to cause disagreement.

Girju (2006) reports impressive agreement (Kappa = 0.66/0.67) on a noun phrase annotation task, but differences in experimental design preclude direct comparison with my results. The data used in that experiment consisted of both noun-noun (NN) and noun-preposition-noun (NPN) phrases taken from a multilingual dictionary and thus might be expected to contain more familiar terms than a balanced corpus containing many technical

[10] My results were first reported in Ó Séaghdha (2007b), which appeared at the same time as Girju's (2007a) paper on in-context multilingual noun phrase annotation. As described below there are important differences between these two tasks.

[11] Gagné et al. (2005a) have demonstrated this phenomenon experimentally by asking subjects to judge interpretations of compounds in contexts that either support or contradict their dominant out-of-context meaning. For example, the compound *bug spray* has the dominant meaning *spray for killing bugs* and a subdominant alternative meaning *spray produced by bugs*; the dominant meaning is supported by the context *Because it was a bad season for mosquitoes, Debbie made sure that every time she went out she wore plenty of bug spray* and the subdominant meaning is supported by *As a defence mechanism against predators, the Alaskan beetle can release a deadly bug spray.* Gagné et al. found that in the case of novel non-lexicalised compounds the meaning judged more likely in out-of-context trials could be overridden completely by a context supporting the subdominant meaning and in the case of familiar lexicalised compounds a subdominant-supporting context made both meanings equally competitive.

items and context-dependent usages. Compounds judged to be lexicalised were discarded and there was no noise in the data as it was not extracted from a corpus. Neither the proportions of NN and NPN phrases nor separate agreement figures for the two phrase types are reported, but the results of Girju (2007a) on other datasets suggest that NPN phrases give better agreement than NN phrases. Furthermore, each compound was presented alongside its translation in four Romance languages. Compounding is relatively rare in these languages and English compounds often have periphrastic translations that disambiguate their meaning – this was in fact the primary motivation for the multilingual experiment. On the other hand, the annotation involved a larger set of semantic categories than the six used in this work and the annotation task will therefore have been more difficult in one aspect; the author lists 22 categories, though only 10 occur in more than 2% of her data.

Girju (2007a) extends her multilingual annotation framework to NN and NPN phrases presented in sentential contexts extracted from parallel corpora. Again, each phrase was presented in English and four Romance languages, serving to clarify the meaning and aid annotation. For her set of 22 semantic relations, agreement of Kappa = 0.61 is reported for English NN phrases extracted from the Europarl corpus of European Parliament proceedings, and agreement of Kappa = 0.56 for English NN phrases extracted from literary texts in the CLUVI corpus. The annotated data used to derive these results is not publicly available, but the subset used in the classification experiments of Girju (2007b) has been made available.[12] Of the 437 Europarl NN compounds in this subset, 261 (59.7%) have the label *TYPE* and 228 (52.2%) are of the same type (*member state*). This is likely to be a major factor in the higher agreement figures on Europarl than on CLUVI, though not necessarily the sole factor.[13] Girju (2007a) also reports agreement on labelling with Lauer's prepositions, which is much better than with semantic relations: Europarl gives Kappa = 0.8, CLUVI gives Kappa = 0.77. As in Girju et al. (2005), "almost all" compounds were labelled with more than one preposition, but Girju (2007a) does not state whether these were excluded from the Kappa calculation.

It is clear from the results reported here and by other authors that the compound annotation task is a very difficult one. Why is this the case? A general problem in semantic annotation of text is that the annotator does not have access to all the information available to the author and his/her intended audience. Interpreting referring expressions in dialogue has been shown to be much harder for overhearers than for participants (Schober and Clark, 1989). In technical or specialist genres, an annotator may lack much of the background knowledge required to arrive at a full or correct interpretation. Even where the source of the data is written and intended for a general readership, it is not practical to read a large portion of the source text as may be necessary for accurate interpretation. This difficulty is exacerbated in the case of compounds, which are often regarded as compressed descriptions of their referents (Downing, 1977). To decompress the semantics of a compound, the hearer must share certain knowledge with the speaker either through mutual world knowledge or through common ground established in the preceding text. The use of compounds thus reflects the tendency of speakers to use shorter referring expressions as a discourse develops (Krauss and Weinheimer, 1964; Clark and Wilkes-Gibbs, 1986; Master, 1993) and the tendency to reduce redundant syntactic structures and main-

[12] `http://apfel.ai.uiuc.edu/resources.html`

[13] It may also explain why Girju's (2007b) classification results are better on the Europarl dataset than on the CLUVI dataset.

tain a constant information density (Levy and Jaeger, 2006). Much of the difficulty in annotation thus arises from the very nature of compounds and compound usage.

4.6 Conclusion

This chapter has presented an empirical evaluation of the semantic annotation scheme for compound nouns that was developed in Chapter 3. Having a measure of inter-annotator agreement is important, as it indicates whether the annotation reliably captures the semantic information shared between users of a language. The correlation of human judgements can also calibrate our expectations of the performance that automatic methods will achieve – it would be surprising if computers performed significantly better than humans at interpreting compounds. The agreement results reported in this chapter compare favourably with other results in the literature; one significant factor in this success is the rigorous development of the annotation schemes and guidelines, which appears to be necessary for reliable and reproducible annotation at a non-trivial semantic depth.

The dataset collected in the course of this annotation work is also used to produce a gold standard for classification experiments in later chapters of this thesis. The data for those experiments consist of all 2,000 compounds from which the annotation trial and test datasets were sampled. It is therefore possible that inter-annotator agreement for the classification data might differ from that for the subset on which agreement was evaluated.[14] If so, this would be a sample size effect and not a bias in the estimation of the "true" agreement, as the data for annotation was sampled randomly. A second point to note is that the classification dataset was labelled by a single annotator and not subject to a reconciliation phase where disagreements between annotators could be resolved. In this situtation the estimate of inter-annotator agreement, in addition to evaluating the learnability of the annotation scheme, provides evidence for the reproducibility of the classification gold standard by measuring whether an independent human annotator chooses the same labelling.

[14]Ideally, the entire set of 2,000 compounds would have been annotated by both annotators. However constraints on time and resources made this impossible in practice.

Chapter 5

Semantic similarity and kernel methods

5.1 Introduction

This chapter presents the theoretical background for the learning methods used in Chapters 6 and 7. In Section 5.2 I state the assumption underlying those methods, that semantic relations in compounds and between nouns in text can be identified through a process of analogical or similarity-based reasoning. Section 5.3 describes relevant prior work on measuring semantic similarity between words and pairs of words. Section 5.4 is an introduction to kernel methods for classification, in particular support vector machines. As well as providing state-of-the-art statistical classifiers, kernel methods allow the application of similarity measures tailored to a particular task. This flexibility will prove very useful in later chapters.

5.2 A similarity-based approach to relational classification

An analogical hypothesis for compound interpretation holds that the relational meaning of a given compound can be predicted, at least in part, by knowledge about the meanings of similar compounds. Or equivalently: the more similar two compounds are, the more likely they are to express similar semantic relations. As I described in Section 2.2.2, analogical reasoning plays a central role in Ryder's (1994) theory of compound semantics. Psycholinguistic studies by Gagné and Shoben (1997; 2002), Tagalakis and Keane (2005) and Devereux and Costello (2007) have found evidence for the reality of semantic priming by similar compounds, though the details of this effect are still the subject of contentious debate among researchers.[1] Analogical effects have also been observed in the choice of morphological linking elements in novel Dutch and German compounds (Krott et al., 2002; Krott et al., 2007) and the placement of stress in English compounds (Plag, 2006; Lappe and Plag, 2007). It is clear that the analogical hypothesis can only be of practical

[1] See, for example, the running debate in Estes (2003), Gagné et al. (2005b) and Estes and Jones (2006) for a flavour of the disagreements on this issue.

use if we possess an appropriate conception of similarity between compounds; the range of options that are available for computational implementation is the topic of Section 5.3.

Statistical methods for classification also make a fundamental assumption of analogy: that similar data items will be likely to belong to the same class. Here, the definition of similarity depends both on the representation chosen for the data items and on the method used to compare those items. A standard approach for choosing an appropriate representation is to extract a set of features that are expected to be informative for class prediction. The process of item comparison may be explicit, as in nearest-neighbour methods, or implicit, as in methods where a classification model is abstracted from training data and test items are classified by referring to that model. In either scenario, if the data violate the assumption that the class distribution correlates with the similarity between items, it will be difficult or impossible to classify unseen test data. Thus the notion of similarity used is of crucial importance. As will be shown in Section 5.4, kernel-based learning offers a flexible framework for engineering task-appropriate similarity measures in the form of kernel functions.

This insight shows that analogical models of relational semantics are not limited to the case of noun-noun compounds. Other tasks, such as *relation extraction*, can be analysed in similar terms. Relation extraction, a very prominent task in the field of information extraction, involves identifying occurrences of specific semantic relations in text. It is perhaps more accurate to describe it as a family of related tasks, as the same term is used to denote identification of relation instances at the sentence level and on a global level. In the former case, each test sentence must be classified according to whether it expresses a relation of interest; in the latter, the evidence provided by a corpus of text that a relation holds between pairs of entities must be evaluated. Compound noun interpretation can be seen as a particular case of the latter task, in which the very use of a compound indicates that some unspecified relation holds between its constituents. Just as a typical relation extraction system may be required to answer the question *what relation (if any) holds between Google and YouTube?*, a compound interpreter must answer questions such as *what relation holds between the concepts kitchen and knife?*. In Section 5.3, approaches to both tasks are outlined in a unified manner, and in Chapters 6 and 7 I show that related methods can be used to classify relations in compound noun data and a more typical relation extraction dataset, that used for the task on classifying relations between nominals at the 2007 SemEval competition (Girju et al., 2007).

5.3 Methods for computing noun pair similarity

In order to implement the similarity-based approach to relation classification, it will be necessary to define a suitable concept of similarity between pairs of words. While there is a long tradition of NLP research on methods for calculating semantic similarity between words, calculating similarity between pairs (or n-tuples) of words is a less well-understood problem. In fact, the problem has rarely been stated explicitly, though it is implicitly addressed by most work on compound noun interpretation and semantic relation extraction. This section describes two complementary approaches for calculating noun pair similarity. The *lexical similarity* approach is based on standard lexical similarity methods and derives a measure of similarity from pairwise similarities between constituents. Section 5.3.1 surveys some appropriate techniques for lexical similarity, with an emphasis on distributional methods that use co-occurrence information extracted from corpora. A second

approach to pair similarity is based on the hypothesis that pairs of words that co-occur in similar contexts will tend to partake in similar semantic relations. This paradigm can be termed *relational similarity*.

These two kinds of similarity are frequently used in NLP, and they are often combined for improved performance. Many approaches to relation extraction combine a measure of similarity between sentences (token relational similarity) with basic semantic information about the words that are candidate relation arguments (Miller et al., 2000; Culotta and Sorensen, 2004; Zhao and Grishman, 2005; Zhou et al., 2005; Zhang et al., 2006). Turney et al. (2003) combine a range of modules for solving SAT analogy questions, including a WordNet-based module for lexical similarity and a Web-based module for type-level relational similarity. Gliozzo et al. (2005) combine single-word analogues of word and token relational similarity for word sense disambiguation. The distinction between word and relational similarity for word pair comparison is recognised by Turney (2006) (he calls the former *attributional similarity*), though the methods he develops use only relational similarity (see Section 5.3.2). Jiang and Zhai (2007) draw a distinction between "properties of a single token" (e.g., unigram counts, entity types) and "relations between tokens" (subsequences, dependency relations) as features for relation extraction, but this distinction is orthogonal to that discussed here as it relates to a given feature's type as opposed to its source.

5.3.1 Constituent lexical similarity

5.3.1.1 Lexical similarity paradigms

Automatic methods for identifying semantically similar words have been studied since the earliest period of NLP research (Masterman, 1956; Spärck Jones, 1964; Harper, 1965), and they remain an active area of investigation. This longevity is in part due to the fundamental importance of lexical semantics for a wide range of language processing tasks, and furthermore because the problem of lexical similarity is a difficult one that is far from solved – the lexical representations used in current state-of-the-art approaches to semantics are quite impoverished in comparison to the multimodal, multirelational nature of human semantic memory. Nonetheless, the techniques that have been developed in this area prove very useful in many practical applications. Dagan et al. (1999) use lexical similarity measures to smooth word probabilities in a language model; Hirst and colleagues have investigated similarity-based techniques for spelling error detection and correction (Hirst and St-Onge, 1998; Hirst and Budanitsky, 2005); Slonim and Tishby (2000) and Bekkerman et al. (2003) demonstrate that word clustering produces powerful features for document clustering and categorisation. I will discuss in turn three distinct paradigms for computing lexical similarity: those that perform simple matching on semantic categories, those that use structural information from hand-built resources and those that extract distributional information from corpora. I will also describe how these paradigms have been applied to relational semantic tasks.

Given a set of semantic categories such as {PERSON, ORGANISATION, LOCATION,...}, the simplest method for comparing two words is to perform binary-valued matching on their categories, i.e., the two words have similarity 1 if they belong to the same category and otherwise have similarity 0. This conception of similarity is often implicit in relation classification methods which integrate context information and basic entity information.

For example, the ACE 2008 Local Relation Detection and Recognition task specifies seven entity types {FACILITY, GEO-POLITICAL ENTITY, LOCATION, ORGANISATION, PERSON} with 31 subtypes (ACE, 2008). This task involves identifying sentences in a test corpus that express one of a pre-defined set of relations.[2] Systems for ACE-style relation classification often make use of these entity types by adding a binary feature for each type to a set of context features (Kambhatla, 2004; Zhou et al., 2005; Jiang and Zhai, 2007), by defining a matching kernel that is summed with a kernel on syntactic structures (Zhao and Grishman, 2005), or by integrating a matching component into the calculation of a structural kernel by upweighting matching substructures that also match in the entity types they contain (Culotta and Sorensen, 2004).

A second approach to lexical similarity exploits the semantic information in manually constructed ontologies. Such ontologies offer many advantages; the accuracy and relevance of the content is guaranteed and the structured nature of the data can provide very rich information about lexical relations. These come at the cost of inflexibility in the face of constant language change and an inevitable lack of coverage of both lexical items and lexical relations; the time and effort required for ontology development means that adaptation to new situations and uses is often not feasible. The ontology most widely used for NLP research is WordNet (Fellbaum, 1998), though thesauri such as Roget's have been used as well.[3] A wide range of lexical similarity measures have been proposed which make use of the hypernymy (*IS-A*) structure of WordNet (Budanitsky and Hirst, 2006) and these measures have been applied to an even wider range of tasks (far too many to list here). For the compound noun interpretation task, Girju et al. (2005) and Ó Séaghdha (2007a) use the WordNet hypernymy hierarchy to derive features for SVM classification, while Kim and Baldwin (2005) directly employ WordNet similarity measures and a nearest neighbour technique to classify compound semantic relations. Kim and Baldwin (2006) use WordNet to generate new compounds from a small seed set of annotated data by substituting constituents of seed compounds with similar words. This bootstrapping method can increase the amount of training data available for machine learning. The WordNet hierarchy was also used by Giuliano et al. (2007) to derive feature vectors for the SemEval 2007 task on classifying semantic relations between nominals.[4] Surprisingly little attention has been paid to the development of kernel functions (see Section 5.4.1) that exploit ontology knowledge; the only such work I am aware of is by Siolas and d'Alche-Buc (2000) and Basili et al. (2006). The main complicating factor from a kernel perspective is that the kinds of functions most frequently used for WordNet similarity are not positive semi-definite;[5] identifying how WordNet kernels could be constructed is an important open research question.

In addition to broad-coverage thesauri, specialised term ontologies have been developed

[2]There are seven relations in ACE 2008 (*ARTIFACT*, *GENERAL AFFILIATION*, *METONYMY*, *ORG-AFFILIATION*, *PART-WHOLE*, *PER-SOC*, *PHYSICAL*), with 18 subtypes. In addition to the Local Relation Detection and Recognition task there is also a Global RDR task, which is a type-level relational task in the terminology of Section 5.3.2.

[3]Kilgarriff and Yallop (2000) give a comparative overview of manually and automatically constructed thesauri for NLP.

[4]This relation classification task is described in detail in Section 6.3.2.

[5]For example, many WordNet similarity measures use information about the most specific common subsumer of two senses, which cannot be expressed in terms of an inner product between vectors. On the other hand, measures based on the similarity between the glosses associated with each sense (such the adapted Lesk measure of Banerjee and Pedersen (2002)) should be more amenable to a kernel approach. One possible implementation of this idea would use string kernels to compare glosses.

for research in fields such as biomedicine, and the hierarchy-driven measures originally devised for WordNet are also suitable for exploiting these resources (Lord et al., 2003; Pedersen et al., 2007). Rosario and Hearst (2001) and Rosario et al. (2002) use the MeSH medical taxonomy to interpret noun compounds by mapping compound constituents into the taxonomy and learning associations between taxonomy elements and semantic relations; Rosario and Hearst (2004) apply a similar approach to classifying relations between treatments and diseases in medical texts. Structured resources other than ontologies that have been used for computing lexical similarity include machine-readable dictionaries (Lesk, 1985; Wilks et al., 1989; Vanderwende, 1994) and electronic encyclopaedias, most notably Wikipedia (Strube and Ponzetto, 2006; Gabrilovich and Markovitch, 2007).

The third paradigm for lexical similarity compares words on the basis of their observed behaviour in naturally occurring language. The core assumption underlying work in this paradigm is often called the *distributional hypothesis*: that two words are semantically similar if they have similar patterns of co-occurrence with other words in some set of contexts. For example, when we observe the word *dog* in a corpus we are more likely to observe it co-occurring with the words *loyal*, *bark* and *fetch* than with the words *infinite*, *juggle* or *dispute*, and as the word *hound* displays similar behaviour, the distributional hypothesis would predict that *dog* and *hound* are semantically similar or related. The word *table* will tend to co-occur with a different set of words and will correspondingly be judged semantically dissimilar to *dog*. The hypothesis was introduced as a theoretical principle by Firth (1957) and Harris (1968), and it motivates Rubenstein and Goodenough's (1965) study of human similarity judgements. The thesis of Spärck Jones (1964) and the paper of Harper (1965) were to my knowledge the first computational implementations of the distributional approach and contain many ideas that are now commonplace in modern data-driven NLP research, from the extraction of co-occurrence vectors from corpora to the use of the semantic similarity measures. Distributional methods lost favour in the mainstream linguistic community with the rise of generative grammar, though they have been readopted by psycholinguists and neurolinguists in recent years (Burgess and Lund, 1998; McDonald and Shillcock, 2001; Huettig et al., 2006; Mitchell et al., 2008).

5.3.1.2 The distributional model

Depending on the choice of a particular interpretation of "patterns of co-occurrence" and a particular notion of context, applying the distributional hypothesis can yield different kinds of similarity, from a tighter relation of synonymy to a looser relation of taxonomic similarity or just a notion of general relatedness. To retain maximal generality, we can formalise the measurement of distributional similarity between target words w_1, w_2 belonging to a vocabulary V_t in terms of a set of co-occurrence types C, a real-valued weighting function g and a similarity function $\text{sim} : \mathbb{R}^{|C|} \times \mathbb{R}^{|C|} \to \mathbb{R}$.[6] Each co-occurrence type $c \in C$ can be decomposed as the pairing of a relation r from a set R of admissible relations and a co-occurring word v from a vocabulary V_c (possibly the same as V_t), i.e., $C \subseteq R \times V_c$. The weighting $g(w, c)$ is some function of the co-occurrence frequency $f(w, c)$ (the number of times w and c were observed co-occurring in a corpus) that may also incorporate information about the marginal distributions of w and c. The overall similarity value $\text{sim}(w_1, w_2)$ will be defined as a combination of the values of a pointwise similarity function sim_0 at

[6] In what follows I will unapologetically abuse notation and write $\text{sim}(w_1, w_2)$ as if it were a function on pairs of words instead of a function on pairs of vectors.

each $c \in C$. In most cases, the overall similarity will be computed by summing over the pointwise similarity scores and those scores will be 0 for co-occurrence types c with $g(w_1, c) = g(w_2, c) = 0$. It follows that we need only count the co-occurrences in the support of $g(w_1, \cdot)$ and $g(w_2, \cdot)$, typically the set of co-occurrences that were observed for either w_1 or w_2 in the corpus, so even when the vocabulary of co-occurrence types is large the computation of $\text{sim}(w_1, w_2)$ can be efficient.

This general description leaves a number of important components undetermined. The size and character of the corpus used to produce frequency estimates will fundamentally affect the results.[7] The broad notion of "co-occurrence relation" admits a large variety of context and co-occurrence types. For example, the co-occurrence relation used may be such that $g(w_1, r, w) > 0$ whenever w_1 and w appear in the same document. This relation specification is often used in information retrieval and also in approaches to lexical semantics inspired by information retrieval (Wong et al., 1985) or Latent Semantic Analysis (Landauer and Dumais, 1997). The size of the admissible context can be reduced so that only co-occurrences in the same paragraph are considered, or co-occurrences within a window of n words around the target w_1. It is often the case that similarity judgements based on textual proximity retrieve word pairs that are semantically or topically related rather than truly similar. To see why, consider the example pair *referee* and *penalty*. Both of these words will appear in the same kinds of sentences and will therefore show a strong association with co-occurring words such as *foul*, *goal*, *blow* and *award*. Proximity information alone cannot identify whether two words perform similar functions in the sentences they appear in. Window-based techniques have been used by many authors, including Church and Hanks (1989), Schütze (1992), McDonald and Shillcock (2001) and Widdows (2003).

An alternative approach is to identify the set R with a set of admissible syntactic relations such as *verb-object*, *verb-subject* or *modifier-noun*. This is a stricter definition of co-occurrence, as only words entering into a syntactic relation with the target word contribute to its distributional profile. As a result, words that are judged similar do not just appear in similar sentences or clauses, they also perform similar functions in those sentences. Syntactic co-occurrences therefore yield a similarity measure that is closer to what we intuitively think of as "similarity". For example, verb-argument co-occurrences were used by Pereira et al. (1993) for clustering nouns, by Grefenstette (1994) and Curran (2003) for automatic thesaurus construction and by Grishman and Sterling (1994) for learning verbal selectional restrictions. Korhonen et al. (2003) perform semantic verb clustering using subcategorisation frame distributions. Lin (1998a; 1998b) uses all dependency relations associated with a word to calculate lexical distributional similarity. Padó and Lapata (2003) generalise the syntactic co-occurrence framework to handle combinations of syntactic relations with a model based on paths in a sentence's dependency graph; the similarity estimates produced by their model correlate well with human priming data. Grefenstette (1993) finds that while syntactic methods generally perform better than window-based methods in identifying semantic neighbours, the opposite behaviour is observed for rare terms. The intuitive explanation is that syntactic co-occurrences can be very sparse for low-frequency words and may not provide enough information to

[7] For example, one would not generally expect *rutabaga*, *clock* and *hedgehog* to be rated as similar words. However, a distributional model derived from a corpus of biomedical articles might indeed judge them similar, as they are all names of genes belonging to the organism *Drosophila* (Morgan et al., 2004). This is a frequent and non-trivial issue in the processing of specialised sublanguages which adopt common words for technical uses.

be useful, whereas the amount of window-derived data for any term is typically greater. Schulte im Walde (2008) also reports that syntactic co-occurrence features outperform window-based features on the task of clustering similar verbs, although the former may be more brittle in the sense that different sets of syntactic relations perform optimally on different datasets.

Whichever co-occurrence relations are selected, the result is a representation of each word in terms of the co-occurrence types observed for that word in the corpus. The standard means of representing a word w is as a vector in a normed vector space where each dimension corresponds to a particular co-occurrence type c and the coordinate value in each dimension is the weight $g(w, c)$.[8] The function g may simply count the number of times w co-occurs with c in the corpus ($g(w, c) = f(w, c)$), or it may be a binary function with value 1 if the co-occurrence (w, c) was observed at least once and value 0 otherwise. One problem that arises when raw frequency counts are used for computing similarity is that the influence of the distributions of the target word and of the co-occurrence type is not considered. A large co-occurrence count $f(w, c)$ may not provide useful information if c co-occurs frequently with many other target words, but its contribution to the similarity profile of w will nonetheless dominate the contribution of less frequent but possibly more discriminative co-occurrence types.

To avoid this problem, one of a number of more sophisticated weighting functions may be used as an alternative; those suggested in the literature include mutual information (Hindle, 1990; Lin, 1998a), the log-likelihood ratio (Dunning, 1993; Padó and Lapata, 2003), odds ratios (Lowe and McDonald, 2000), z-scores (Weeds and Weir, 2005), t-tests and χ^2-tests (Curran, 2003). These statistical measures compare the observed co-occurrence counts with the counts that would be expected to occur by chance.[9] Calculating the expected frequencies requires that we know the marginal frequencies $f(w)$ (how often w co-occurs with any type in the corpus) and $f(c)$ (how often any target word co-occurs with c), and usually the total number N of word-type co-occurrences in the corpus. With an appropriate choice of processing tools, this requirement can usually be met even for very large corpora (Curran, 2003). However, in some scenarios it is impossible or infeasible to compute the marginals, particularly when the co-occurrence relations are syntactically informed. This can be the case when co-occurrences are extracted only from a subset of the corpus known to contain all co-occurrences of certain target words but not all co-occurrences of other words, such as when a Web corpus is created by submitting targeted queries to a search engine. If the co-occurrence types are derived from full syntactic parses, as in the model of Padó and Lapata (2003), the amount of time available for parsing will limit the size of the corpus that can be used (without the need to compute the marginals, only those sentences containing a target word must be parsed).[10]

An alternative weighting strategy is to represent each target word w by its *co-occurrence probability distribution* $P(C|w)$. For a particular co-occurrence type c the value of $P(c|w)$ gives the conditional probability of observing c in a co-occurrence given that the target

[8] The condition that the space should be a normed space is equivalent to requiring that the space have a distance function.

[9] Evert (2004) gives a thorough overview of measures for estimating the strength of association between co-occurring terms, with a focus on their application in collocation extraction.

[10] One workaround to practical limits on corpus size is to use a sketching algorithm as in Li and Church (2007) to estimate the marginals from a representative sample of a larger corpus. As far as I am aware, this has not yet been tried for computing semantic similarity, but it could well be a productive direction of enquiry.

Dot product	$\text{sim}_{dot}(w_1, w_2)$	$= \sum_c P(c\|w_1)P(c\|w_2)$
Cosine	$\text{sim}_{cos}(w_1, w_2)$	$= \frac{\sum_c P(c\|w_1)P(c\|w_2)}{\sqrt{\sum_c P(c\|w_1)}\sqrt{\sum_c P(c\|w_2)}}$
Weeds and Weir *add*	$\text{sim}_{WW_{add}}(w_1, w_2)$	$= \beta \sum_{c \in S(w_1 \cap w_2)} P(c\|w_1) + (1-\beta) \sum_{c \in S(w_1 \cap w_2)} P(c\|w_2)$
Weeds and Weir *dw*	$\text{sim}_{WW_{dw}}(w_1, w_2)$	$= \sum_c \min(P(c\|w_1), P(c\|w_2))$
Jaccard	$\text{sim}_{Jaccard}(w_1, w_2)$	$= \frac{\sum_c \min(P(c\|w_1), P(c\|w_2))}{\sum_c \max(P(c\|w_1), P(c\|w_2))}$
Lin (1998b)	$\text{sim}_{Lin}(w_1, w_2)$	$= \frac{\sum_{c \in S(w_1) \cap S(w_2)} IC(c)}{\sum_{c \in S(w_1)} IC(c) + \sum_{c \in S(w_2)} IC(c)}$

Table 5.1: Similarity measures for co-occurrence probability distributions. $S(w)$ is the support of $P(C|w)$, $P(c)$ is the probability of observing any target word co-occurring with c, and $IC(c) = -\log_2(P(c))$ is the information content of that event.

word is w. The co-occurrence types for which $P(c|w)$ is high can be called the preferred co-occurrences of w, and it follows from the distributional hypothesis that two words whose preferred co-occurrences overlap are likely to be semantically similar. Note that a word cannot give a high co-occurrence probability to a large number of types, due to the constraint that its co-occurrence probabilities must sum to 1.

The maximum likelihood estimate of the "true" population value of $P(c|w)$ is simply $\frac{f(w,c)}{\sum_c f(w,c)}$, the number of times w and c co-occur in the corpus divided by the total number of times w co-occurs with any co-occurrence type. Given an arbitrary ordering $c_1, \ldots, c_d$ of co-occurrence types the vector $\mathbf{p}_w = (P(c_1|w), \ldots, P(c_d|w))$ parameterises a multinomial or categorical probability distribution. Using these parameter vectors as a data representation guarantees a sound probabilistic interpretation for our model and allows us to profit from methods that have been designed for the specific purpose of comparing and discriminating probability distributions. With regard to the problem of chance co-occurrences mentioned above, the conditional probability representation should not be affected by the marginal probability of the target word as each vector must by definition sum to 1, but it does not in itself dispel the effect of the co-occurrence type marginals. We might hope to use similarity measures that are relatively robust to this effect, and in practice this does seem to be possible (see Chapter 6). The use of conditional distributions is implicit in much work on semantic similarity and is explicitly treated by Pereira et al. (1993), Dagan et al. (1999) and Lee (1999), among others.

5.3.1.3 Measures of similarity and distance

Similarities

The final component required for computing semantic similarity is the semantic similarity function itself. Table 5.1 lists the best-known lexical similarity measures that are suitable for comparing arbitrary (conditional) probability distributions over the same

event space.[11] The discussion in this section will assume that co-occurrence vectors are weighted with an unspecified function g (Section 5.3.1.2), of which the conditional probability weighting in Table 5.1 is a particular case.

The *dot product* or *scalar product* is a fundamental concept in linear algebra but it is rarely used as a lexical similarity measure as it is sensitive to the magnitudes (L_2 norms) of the vectors $\mathbf{g}_{w_1}$ and $\mathbf{g}_{w_2}$. This is made clear by restating the dot product as

$$\text{sim}_{dot}(w_1, w_2) = \|\mathbf{g}_{w_1}\|\|\mathbf{g}_{w_2}\| \cos\theta \tag{5.1}$$

where θ is the angle between the two vectors. However, the dot product is often used with support vector machines for a variety of classification tasks; in that context, it is named the *linear kernel* (see Section 5.4.1). The *cosine* similarity measure gives the cosine of the angle between $\mathbf{p}_{w_1}$ and $\mathbf{p}_{w_2}$; as suggested by (5.1), this corresponds to the dot product when $\mathbf{g}_{w_1}$ and $\mathbf{g}_{w_2}$ are normalised to have unit magnitude. The cosine measure has a long history of use in information retrieval and is as close to a standard similarity measure as exists in NLP.

Another class of similarity measures derives from measures originally designed for comparing sets or, equivalently, binary vectors. One such measure is that of Jaccard (1901), for arbitrary sets A and B:

$$\text{sim}_{SetJaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.2}$$

There are a number of ways in which the Jaccard coefficient can be generalised to compare vectors in $\mathbb{R}^n$. The version stated in Table 5.1, used by Grefenstette (1994) and Curran (2003), takes the min function as the real-valued analogue of set intersection and the max function as the analogue of union. Curran also considers the formulation

$$\text{sim}_{AltJaccard}(w_1, w_2) = \frac{\sum_c g(w_1, c) g(w_2, c)}{\sum_c g(w_1, c) + g(w_2, c)} \tag{5.3}$$

When the weighing function g is the conditional probability $P(c|w)$, the denominator in (5.3) will always equal 2 and $\text{sim}_{AltJaccard}$ will reduce to 0.5 times sim_{dot}. An alternative measure of set similarity is the Dice coefficient (Dice, 1945):

$$\text{sim}_{SetDice}(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{5.4}$$

$\text{sim}_{SetDice}$ increases monotonically with $\text{sim}_{SetJaccard}$; the relationship between them is given by $\text{sim}_{SetJaccard}(A, B) = \frac{\text{sim}_{SetDice}(A,B)}{2 - \text{sim}_{SetDice}(A,B)}$. For any word w, the two measures will always give the same ranking of most similar words to w. If we generalise the Dice coefficient by replacing the intersection in (5.4) with the min function and the denominator

[11] This definition excludes the mutual information measure, which computes an information-theoretic notion of similarity between random variables by comparing their joint probability distribution to a factorised distribution but is not suitable for comparing arbitrary distributions. Also excluded are similarity measures on individual events, such as pointwise mutual information, t-test and the log likelihood ratio; as described above, these are often applied as weighting functions prior to the calculation of lexical similarity. I also omit discussion of the confusion probability measure (Essen and Steinbiss, 1992); while this measure does compare arbitrary distributions, it has the surprising property that the most similar word to a given w_1 may not be w_1 itself, and Lee (1999) finds that it performs relatively poorly at estimating semantic similarity.

with $\sum_c P(c|w_1) + P(c|w_2)$ the resulting measure is equal to $\sum_c \min(P(c|w_1), P(c|w_2))$, which is the same as Weeds and Weir's (2005) $\text{sim}_{WW_{dw}}$. Replacing the intersection with a product yields twice $\text{sim}_{AltJaccard}$, which is equal to sim_{dot} for conditional probability vectors.

Lin (1998a; 1998b) presents two similarity measures that can be viewed as variants of the Dice coefficient, in that they divide a measure of the intersected support of w_1 and w_2 by the sum of the corresponding measures of the individual supports. Both measures are motivated by the same information-theoretic principle, that an appropriate measure of similarity between two items should quantify the amount of information the items share divided by the sum of the information each item possesses individually. Lin's (1998a) measure takes the pointwise mutual information between a target word w and co-occurrence type c as the weighting function and replaces the set cardinality measure in (5.4) with a sum over set member weights:

$$\text{sim}_{Lin98a}(w_1, w_2) = \frac{\sum_{c \in S(w_1) \cap S(w_2)} MI(w_1, c) + MI(w_2, c)}{\sum_{c \in S(w_1)} MI(w_1, c) + \sum_{c \in S(w_2)} MI(w_2, c)} \tag{5.5}$$

Here $S(w) = \{c \,|\, g(w, c) > 0\}$ is the support of $g(w, \cdot)$ and $S(w_1) \cap S(w_2)$ is the intersection of the supports of $g(w_1, \cdot)$ and $g(w_2, \cdot)$. $MI(w, c) = \log \frac{f(w,c)}{f(w)f(c)}$ is the mutual information between w and c. If conditional probability is used instead of the mutual information weighting, then $\text{sim}_{Lin98a}(w_1, w_2)$ reduces to $\text{sim}_{WW_{add}}$ with $\beta = 0.5$. The measure of Lin (1998b), stated in Table 5.1, is similar to $\text{sim}_{Lin98a}(w_1, w_2)$ except that the weighting function used is not conditioned on the words being compared. Instead, a global information weight is assigned to each co-occurrence type such that the weight $IC(c)$ of each co-occurrence is the information content of the event that c co-occurs with any target word in the corpus.

Weeds and Weir (2005) describe a general framework for deriving distributional similarity measures. They cast the task of calculating similarities as *co-occurrence retrieval* and extend the analogy with information retrieval by defining the similarity between w_1 and w_2 in terms of precision (the degree to which w_2 retrieves the preferred co-occurrences of w_1) and recall (the degree to which w_1 retrieves the preferred co-occurrences of w_2). For one class of similarity measure, the *additive models*, these quantities are defined as follows:

$$P_{add}(w_1, w_2) = \frac{\sum_{c \in S(w_1) \cap S(w_2)} g(w_1, c)}{\sum_{c \in S(w_1)} g(w_1, c)} \qquad R_{add}(w_1, w_2) = \frac{\sum_{c \in S(w_1) \cap S(w_2)} g(w_2, c)}{\sum_{c \in S(w_2)} g(w_2, c)} \tag{5.6}$$

When $g(w, c)$ is the probability $P(c|w)$, the denominators in both definitions will sum to 1. A second class of measures, the *difference-weighted models*, have the following definitions:[12]

$$P_{dw}(w_1, w_2) = \frac{\sum_{c \in S(w_1) \cap S(w_2)} \min(g(w_1, c), g(w_2, c))}{\sum_{c \in S(w_1)} g(w_1, c)} \tag{5.7}$$

$$R_{dw}(w_1, w_2) = \frac{\sum_{c \in S(w_1) \cap S(w_2)} \min(g(w_1, c), g(w_2, c))}{\sum_{c \in S(w_2)} g(w_2, c)} \tag{5.8}$$

[12]Weeds and Weir actually define the difference-weighted models in terms of an *extent function*. This gives the same definitions as the weighting function g in all cases except the *difference-weighted type-based model*. As I am not discussing this model, and in the interests of clarity, I will overlook the distinction.

L_1 distance	$\text{dist}_{L_1}(w_1, w_2)$	$= \sum_c \lvert P(c\lvert w_1) - P(c\lvert w_2)\rvert$
L_2 distance	$\text{dist}_{L_2}(w_1, w_2)$	$= \sqrt{\sum_c (P(c\lvert w_1) - P(c\lvert w_2))^2}$
Kullback-Leibler divergence	$\text{dist}_{KL}(w_1, w_2)$	$= \sum_c P(c\lvert w_1) \log_2 \frac{P(c\lvert w_1)}{P(c\lvert w_2)}$
α-skew divergence	$\text{dist}_{\alpha}(w_1, w_2)$	$= \sum_c P(c\lvert w_1) \log_2 \frac{P(c\lvert w_1)}{\alpha P(c\lvert w_2) + (1-\alpha) P(c\lvert w_1)}$
Jensen-Shannon divergence	$\text{dist}_{JS}(w_1, w_2)$	$= \sum_c P(c\lvert w_1) \log_2 \frac{2P(c\lvert w_1)}{P(c\lvert w_1) + P(c\lvert w_2)} +$ $P(c\lvert w_2) \log_2 \frac{2P(c\lvert w_2)}{P(c\lvert w_1) + P(c\lvert w_2)}$

Table 5.2: Distance measures for co-occurrence distributions

Again the denominators sum to 1 when the distributional probability weighting is used. Weeds and Weir discuss methods for combining precision and recall using a weighted arithmetic mean, the harmonic mean (F-measure) and a weighted sum of the arithmetic and harmonic means. The most general statement of their model is:

$$\text{sim}_{WW}(w_1, w_2) = \gamma \left[\frac{2P(w_1, w_2)R(w_1, w_2)}{P(w_1, w_2) + R(w_1, w_2)} \right] + (1 - \gamma) \left[\beta P(w_1, w_2) + (1 - \beta) R(w_1, w_2) \right] \tag{5.9}$$

The definition in (5.9) is shown by Weeds and Weir to have many known similarity measures as special cases, including sim_{Lin98a}, $\text{sim}_{SetDice}$ and a transformed dist_{L_1} (see below). The definitions in Table 5.1 are based on the arithmetic mean ($\gamma = 0$) as this formulation facilitates comparison with other measures.

Distances

It is also possible to arrive at a similarity measure by starting with a notion of distance or dissimilarity. Intuitively, the more distant the representations of two words are, the less similar the words should be. Table 5.2 lists the distance measures that have previously been applied to co-occurrence distributions. They can be grouped in two classes: distances appropriate for $\mathbb{R}^n$ and distances appropriate for $\mathcal{M}_+^1(C)$, the space of probability measures on C. If the task at hand is one of ranking words by similarity to a given w, a distance measure can be used as is and a ranking of most similar to least similar can be produced by ranking from least distant to most distant. On the other hand, if the degree of similarity between two words is to be quantified, measures of distance must be transformed into measures of similarity. The approaches described in the literature use heuristic transformations; in Section 5.4.1 we will see one theoretically motivated method for deriving similarities from distances.

The L_1 and L_2 distances are instances of Minkowski or L_p distances. For $p \geq 1$, the L_p

distance between vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is defined as:[13]

$$\text{dist}_{L_p}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{5.10}$$

As p increases, the distance becomes dominated by dimensions i for which $|x_i - y_i|$ is largest. The limiting distance P_∞ is the maximum of all the dimension-wise differences. Previous work on semantic similarity indicates that the L_1 distance performs better than the L_2 distance (Lee, 1999). Lee suggests this difference is due to the fact that the contribution of co-occurrence types outside of the intersected support $S(w_1) \cap S(w_2)$ is squared in L_2 compared to L_1. Another factor may be that unless the co-occurrence types are normalised for variance, the difference $|P(c|w_1) - P(c|w_2)|$ will be greater for more frequent c and higher values of p exacerbate this effect.[14] To derive a similarity measure from dist_{L_1}, Lee (1999) uses

$$\text{sim}_{L_1}(w_1, w_2) = 2 - \text{dist}_{L_1}(w_1, w_2) \tag{5.11}$$

Dagan et al. (1999) use the same similarity function, but raised to an parameterised power β. The constant term 2 arises in (5.11) because $0 \leq \text{dist}_{L_1}(w_1, w_2) \leq 2$ when w_1 and w_2 are represented by probability distributions. Weeds and Weir (2005) also derive the formula (5.11) as a special case of their difference-weighted model via the identity $\text{dist}_{L_1}(w_1, w_2) = 2 - 2\sum_c \min(P(c|w_1), P(c|w_2))$.

The development of distance measures between probability distributions has been pursued most energetically in the field of information theory. It is often of interest to quantify the degree to which one distribution captures the information content of another. The L_1 distance is often known as the *variational distance* in this context. Probably the most important measure of distance between probability measures on a set C is the *Kullback-Leibler divergence* (Kullback and Leibler, 1951):

$$\text{dist}_{KL}(P, Q) = \sum_{c \in C} P(c) \log_2 \frac{P(c)}{Q(c)} \tag{5.12}$$

The KL divergence can be interpreted as the expected information loss incurred by approximating a "true" distribution P with a distribution Q, or equivalently as the expected increase in word length (in bits) when a code that is optimal for the distribution Q is used to describe data produced by the distribution P. It is a fundamental concept in many areas of statistical NLP. For example: it has been used for clustering by Pereira et al. (1993); the popular maximum entropy modelling paradigm is based on minimising the KL divergence between the model distribution and a uniform distribution subject to empirically derived constraints (Berger et al., 1996; Della Pietra et al., 1997); the G^2 log-likelihood measure of association can be viewed as the KL divergence between observed and expected cell values in a contingency table (Evert, 2004). The KL divergence is only defined if $Q(c) > 0$ whenever $P(c) > 0$, i.e., P must be *absolutely continuous* with regard to Q. This condition is rarely met for pairs of co-occurrence distributions. Lee (1999) has introduced the *α-skew divergence* to address this problem. The α-skew divergence

[13] The condition $p \geq 1$ is necessary for L_p to be a metric, as otherwise the triangle inequality does not hold.

[14] This topic is taken up again in Section 6.8.1.

between distributions P and Q is defined as the KL divergence between P and a mixture $\alpha Q + (1 - \alpha)P$, with the smoothing parameter α typically set very close to 1 (Lee uses 0.99).

The KL and α-skew divergences are clearly asymmetric. This property is not necessarily undesirable for semantic modelling, as some aspects of semantic similarity are often argued to function asymmetrically.[15] However, a symmetric distance measure is required for many applications, including the distributional kernels derived in Chapter 6. Kullback and Leibler (1951) describe a symmetric divergence which they call J:

$$\text{dist}_J(P, Q) = \text{dist}_{KL}(P, Q) + \text{dist}_{KL}(Q, P) \tag{5.13}$$

$$= \sum_{c \in C} (P(c) - Q(c)) \log_2 \frac{P(c)}{Q(c)} \tag{5.14}$$

Like the standard KL divergence, J also suffers from the problem of undefinedness when the absolute continuity condition is not met. An alternative is the *Jensen-Shannon divergence* (Rao, 1982; Lin, 1991):[16]

$$\text{dist}_{JS}(P, Q) = \frac{1}{2} \text{dist}_{KL}\left(P, \frac{P+Q}{2}\right) + \frac{1}{2} \text{dist}_{KL}\left(Q, \frac{P+Q}{2}\right) \tag{5.15}$$

The information theory literature contains numerous interpretations of the Jensen-Shannon divergence. It can be understood as the information transmission rate in a communication channel with an equiprobable binary input and output generated half the time by P and half the time by Q (Topsøe, 2000). It is also the expected information gain from a single sample on the task of deciding between generating models P or Q, with priors for both hypotheses equal to 0.5 (Endres and Schindelin, 2003). Unlike Kullback and Leibler's J, the Jensen-Shannon divergence is always defined. A further property that will prove useful later is that it is a squared metric (Endres and Schindelin, 2003; Fuglede, 2005). Jensen-Shannon divergence has been successfully applied to many NLP tasks, including word and document clustering (Slonim and Tishby, 2000), word sense disambiguation (Dagan et al., 1997; Niu et al., 2005) and the analysis of statistical parsing models (Bikel, 2004). When a quantification of similarity is required, most authors follow Dagan et al. (1999) in using the transformation $\text{sim}_{JS} = 10^{-\beta \text{dist}_{JS}}$, though Lin (1999) uses $2 - \text{dist}_{JS}$.

5.3.1.4 From words to word pairs

Once we have specified a model of lexical similarity, some adaptation is necessary in order to obtain a model of word pair similarity. One possible approach for compounds is to treat them as "words with spaces" and calculate similarity between compounds in exactly the same way as similarity between words. As compounds tend to be far more sparsely distributed than single words it will be very difficult to attain accurate probability estimates even from large corpora. It might be possible to accumulate sufficient data by submitting targeted queries to a Web search engine. However, there is also a conceptual

[15] For example, Tversky (1977) presents evidence that the prototypicality and discourse context of similarity stimuli can affect subjects' judgements. In comparisons of prominent and less-prominent countries, Tversky's subjects overwhelmingly preferred statements such as "North Korea is similar to Red China" to the reverse statement ("Red China is similar to North Korea").

[16] It is also known as *capacitory discrimination* in the information theory literature.

problem with this approach: although the context of a compound may contain information about the referent of the compound, it is less likely to contain information about the implicit semantic relation. For example, the following compounds all encode different relational meanings but are likely to appear in similar contexts:

- John cut the bread with the *kitchen knife*.
- John cut the bread with the *steel knife*.
- John cut the bread with the *bread knife*.

Ó Séaghdha and Copestake (2007) report that the "words with spaces" approach performs very poorly for compound interpretation, using co-occurrence information from both the BNC and the 2 billion-word English Gigaword Corpus (Graff et al., 2005). As a result, I am not considering it further here.

A more fruitful approach is to calculate the similarity of two compounds from the pairwise lexical similarities of their constituents: pairs (N_1, N_2) and (N_3, N_4) are judged similar if N_1 is similar to N_3 and N_2 is similar to N_4. The lexical similarities $\text{sim}(N_1, N_3)$ and $\text{sim}(N_2, N_4)$ can be combined linearly, i.e.:

$$\text{sim}_{pair}((N_1, N_2), (N_3, N_4)) = \alpha[\text{sim}(N_1, N_3)] + \beta[\text{sim}(N_2, N_4)] \tag{5.16}$$

Alternatively, a co-occurrence probability vector can be constructed for each compound by appending the distributional vectors of its two constituents and, if desired, rescaling by 0.5 to ensure that the compound vector sums to 1. Lexical similarity measures or feature-based machine learning methods can be applied directly to the joint vector. This second approach has given better results in preliminary experiments and is the method adopted in the following chapters.

5.3.2 Relational similarity

5.3.2.1 The relational distributional hypothesis

Whereas measures of distributional lexical similarity consider the co-occurrences of each constituent of a word pair separately, relational similarity is based on the contexts in which both constituents appear together. The underlying intuition is that when nouns N_1 and N_2 are mentioned in the same context, that context is likely to yield information about the relations that hold between those nouns' referents in the world. For example, the sentences in 1 and 2 below provide evidence about the relations between *bear* and *forest* and between *fish* and *reef*, respectively.

1. (a) *Bears* still inhabit the *forests* of Italy.
   (b) Wandering in the *forest*, I encountered a *bear*.
2. (a) These brightly-coloured *fish* inhabit the coastal *reefs*.
   (b) Diving in the *reef*, I saw many *fish*.

Sentences 1a and 2a are similar in that they share the subject-verb-object triple N_1-*inhabit*-N_2. Sentences 1b and 2b appear quite different lexically, but they have identical syntactic structures and both match the lexico-syntactic pattern *V-ing in the* N_2, *Pro V Det* N_1. In both cases, the shared patterns are clues that a *LOCATED-IN* or *LIVES-IN* relation holds between each noun pair. If we knew the correct semantic relation label for (*bear*, *forest*), we could justifiably predict the same relation for (*fish*, *reef*).

A *relational distributional hypothesis* would therefore state that two word pairs are semantically similar if their members appear together in similar contexts. This definition leaves a number of free parameters. First of all, a definition of "context" is required. A plausible starting point is to identify the context of a (N_1,N_2) pair with the sentence in which the nouns co-occur. Some alternatives are to take only the substring between N_1 and N_2, or to take this middle context plus some number of words outside the two nouns. These two alternatives are motivated by the hypothesis that the contexts close to and between N_1 and N_2 are more likely to contain information about their relationship than contexts that are more distant. As in the lexical case, each context fitting the chosen definition can be represented as an unordered bag of words, as an ordered sequence or string, or as a tree or graph structure derived from a parse of the context sentence. The range of measures available for comparing contexts is also similar to those used for lexical similarity, though in practice most researchers seem to use the cosine measure or, when using support vector machines, kernels based on the L_2 distance; Sections 5.3.2.2 and 5.3.2.3 describe some previously reported methods.

Given a suitable model of relational similarity, there are two kinds of questions to which it can be applied. The first kind asks whether the relation expressed in one context is likely to be the same as that in another; e.g. if we know that sentence 1a expresses a *LOCATED-IN* relation between *bear* and *forest*, does sentence 2a also express a *LOCATED-IN* relation between *fish* and *reef*? This is a problem of *token-level* relational similarity, as it involves comparing two instances or tokens of the noun pairs (*bear*, *forest*) and (*fish*, *reef*). The second kind of question is not specific to a pair of contexts, but asks about the general similarity of two noun pairs; e.g. if we know that the typical relation between *bear* and *forest* is a locative one, is the relation between *fish* and *reef* also typically locative? This can be called a problem of *type-level* relational similarity. Both kinds of problems are frequently encountered in NLP research. A token-level perspective is implicit in any task requiring the identification of relations between constituents of a sentence, from semantic role labelling to anaphora resolution. Turney (2006; 2008) has argued that a wide range of semantic processing tasks, including compound interpretation, synonym/antonym identification and modelling lexical associations, can be treated as analogical problems based on type-level relational similarity.

Although compound noun interpretation is usually treated as a type-level problem and methods for relation classification usually take a token-level approach, both tasks combine token-level and type-level aspects to different degrees. It is well-known that the context in which a compound noun is used can modify its conventional type-level meaning; this is a token-level effect. Leveraging token-level information for compound interpretation is difficult, and to my knowledge no previous research has done so. Likewise, the relation between two entities in a sentence is primarily indicated by the context, but prior knowledge about the typical or probable relations between them can also steer interpretation. This dynamic has not been widely adopted for biomedical or ACE-style relation extraction, but in the 2007 SemEval task on identifying semantic relations between nominals,

a number of competitors used type-level information (Nakov and Hearst, 2007b; Nulty, 2007b).

5.3.2.2 Methods for token-level relational similarity

The notion of token-level similarity encompasses all approaches that use information about an item's context to classify it. A wide range of semantic classification tasks are amenable to such an approach, including semantic role labelling (Pradhan et al., 2004; Zhang et al., 2007) and word sense disambiguation, both supervised (Gliozzo et al., 2005) and unsupervised (Mihalcea, 2005). Token-level approaches are also standard in semantic relation classification, whereby the context in which a word pair appears is used to extract features for supervised learning. Kambhatla (2004) uses what can be considered a standard set of features for the ACE 2004 Relation Detection and Classification task: the pair words themselves, all words appearing between them, the ACE entity type of the pair words, whether they are referred to by a name, a nominal or a pronoun, the number of words and entity mentions between them, the words with which they enter a grammatical relation, and the path between them in the sentence parse tree. Additional features have been proposed, such as word bigrams (Zhao and Grishman, 2005) and syntactic chunks (Zhou et al., 2005). Maximum entropy classifiers and support vector machines have been the most popular machine learning techniques for this task, but it is possible to apply any statistical classifier. Ray and Craven (2001) use hidden Markov models for protein-localisation extraction from biomedical abstracts; Goadrich et al. (2004) apply an Inductive Logic Programming approach to the same task. Miller et al. (2000) integrate relation detection into a statistical parsing model, in such a way that the relation detection can inform the syntactic parsing and vice versa. The main drawback to this interesting approach is that it requires rich semantic annotation of parse trees to provide training data. In contrast, Chen et al. (2006a) have investigated graph-based semi-supervised methods which can still perform well when very little labelled data is available.

An alternative to the standard feature engineering approach is the use of kernel methods for structured data such as strings and trees. Convolution kernels (discussed in Chapter 7) facilitate the use of standard feature-based classification methods with non-vectorial data by implicitly mapping each data item into a feature space whose dimensions correspond to its substructures. Thus strings are mapped onto vectors of substring counts, and trees are mapped onto vectors of subtree counts; however, it is not necessary to explicitly represent these high-dimensional vectors due to the properties of kernel functions (Section 5.4.1). Bunescu and Mooney (2005b) apply string kernels to biomedical and ACE relation extraction. Each pair instance is represented by three substrings of the context sentence: up to four words before and between the pair words, up to four words between the pair words, and up to four words between and after the pair words. The kernel function used for classification is the sum of three string kernels each dedicated to one of these substring types. The limit of four words improves the efficiency of kernel calculations and prevents overfitting. Bunescu and Mooney demonstrate that their string kernel used with a support vector machine outperform rule-based and other statistical classifiers. Giuliano et al. (2007) use Bunescu and Mooney's string kernel as one of five heterogeneous kernels for the SemEval task on classifying relations between nominals.

While string kernels incorporate the linear structure of language, they do not capture syntactic structure. Tree kernels, on the other hand, do make use of syntax through a

parse tree representation of sentences. The use of tree kernels for relation classification was first suggested by Zelenko et al. (2003), who use a representation based on shallow parse trees to identify *PERSON-AFFILIATION* and *ORGANISATION-LOCATION* relations in newspaper text. Culotta and Sorensen (2004) take a similar approach based on feature-rich dependency parses rather than traditional constituent parse trees. Subsequent research has refined the design of syntactic kernels for relation classification by identifying more precisely which area of the sentence parse tree should be used for the kernel calculation (Bunescu and Mooney, 2005a; Zhang et al., 2007; Zhou et al., 2007). Using a combination of a "context-sensitive" tree kernel and Zhou et al.'s (2005) standard feature-based linear kernel, Zhou et al. (2007) attain state-of-the-art performance on the ACE 2003 and 2004 Relation Detection and Classification datasets.

5.3.2.3 Methods for type-level relational similarity

Type-level information can be applied to a number of related problems. One problem is to decide whether or not a certain relation holds between two nouns N_1 and N_2. For many relations *REL*, we can make the assumption that (N_1, N_2) is a positive example of *REL* if any instance of the type pair provides reliable evidence that *REL* holds. This can be called a *multiple instance problem*, by analogy to the multiple instance learning paradigm (Dietterich et al., 1997) where this assumption is central. An example of this approach is the hyponym identification task considered by Hearst (1992), where the goal is to find pairs (N_1, N_2) such that N_1 is a hyponym of N_2, e.g. (*carrot*, *vegetable*), (*hammer*, *tool*). The technique applied by Hearst is to search a corpus for lexical patterns such as N_1 *and other* N_2 and N_2 *such as* N_1. If any instance of this pattern is found for a given pair, the hyponymy relation is assumed to hold for that pair. This pattern-matching approach tends to achieve high precision but low recall. It seems best suited to domains and relations which are conventionally described in a restricted number of ways, such as are often found in scientific writing. For example, Pustejovsky et al. (2002) describe a system that identifies pairs of proteins satisfying an *INHIBITS* relation, by extracting the subject and object arguments taken by instances of the verb *inhibit* in biomedical abstracts.

In general, a relation can be expressed in many different ways and it is impractical to manually specify a sufficient set of patterns for high-recall extraction. The bootstrapping approach introduced by Brin (1998) and Agichtein and Gravano (2000) addresses this issue by automatically identifying reliable patterns with minimal user input.[17] Bootstrapping algorithms take a user-supplied initial seed set of positive example pairs for the relation of interest and iteratively discover new patterns and new positive examples. A number of authors have recently extended this method by applying it to the large amount of text available on the World Wide Web (Feldman and Rosenfeld, 2006; Tomita et al., 2006), and by using syntactic patterns rather than word sequences (Stevenson and Greenwood, 2005; Greenwood and Stevenson, 2006).

A further, distinct approach to multiple instance relation extraction is to classify each instance of a word pair using token-level techniques and to then classify the pair as a positive relation example if any of its instances is positive. Bunescu and Mooney (2007) use a support vector machine classifier with a string kernel function to identify person-birthplace pairs and pairs relating to corporate acquisitions. A small training set of

[17] Bootstrapping was suggested earlier by Hearst (1992), but not implemented.

positive and negative example pairs is used to extract sets of training instances, each of which is labelled with the class of its type; a test pair is classified as positive if the SVM labels any of its instances as positive. This method is shown to work quite well, though Bunescu and Mooney observe that care must be taken in weighting the effects of each instance to compensate for the over-general assumption that every instance of a positive (or negative) pair is a positive (or negative) example of the relation.

Instead of focusing on a single relation and searching for positive examples of that relation, it is often of interest to take two nouns and study the distribution of possible and probable relations between them. This is the case when interpreting noun compounds. The first type-level approach to compound interpretation was Lebowitz' (1988) RESEARCHER system for processing patent abstracts. To understand the compound *motor spindle*, RESEARCHER maps each constituent onto its conceptual definition in a semantic dictionary (here DRIVE-SHAFT# and MOTOR#) and searches its "memory" for prior instances of this concept pair. If RESEARCHER has previously encountered and assigned a semantic relation to the concept pair, it assumes that the compound also expresses that relation. For example, if the sentence *The motor includes a spindle* has been seen and interpreted as expressing a *HAS-PART* relation, RESEARCHER will identify the compound *motor spindle* as expressing a *HAS-PART* relation as well. The utility of this approach is limited by the assumptions that at most one relation applies to any concept pair, and that the relation will have been encountered before the compound. As a result, it is only appropriate to closed domains such as the patent abstracts studied by Lebowitz.

Type-level information is applied in a different way by Lauer (1995) to generate prepositional paraphrases for noun compounds. As described in Section 2.3.2, Lauer uses a set of eight prepositions as proxies for semantic relations; the compound interpretation task thus becomes one of identifying the preposition most likely to join the two constituents. To counter the problem of sparsity, Lauer assumes a probabilistic model with independent contributions from each constituent; the preposition predicted for a pair (N_1, N_2) is that P maximising the probability $P(P|N_1, N_2) = P(P|N_1)P(P|N_2)$.[18] As the conditional probabilities $P(P|N_1)$ and $P(P|N_2)$ can easily be estimated from any corpus, this model is simple and general in its application. Furthermore, it is an unsupervised method and does not require annotated training data. Lapata and Keller (2004) use Web counts to estimate the model probabilities and show that this gives significantly better results than models estimated from smaller corpora. Despite its advantages, Lauer's model can only be used when we are willing to identify the set of semantic relations we are interested in with a set of lexical items. As noted in Chapter 2, this is often not desirable.

A more general model replaces the assumption that lexical items or other surface forms map unambiguously onto semantic relations with the weaker but more realistic assumptions that they provide evidence for semantic relations. This model shares the advantages of efficiency and general applicability with Lauer's model, while deepening the kind of semantics it can provide. Rosario and Hearst (2005) consider the task of identifying the relation between two proteins that is supported by a biomedical article; this is not an archetypal type-level task as it distinguishes occurrences of the same pair in different doc-

[18] Lauer also investigates a model based on the concepts provided for N_1 and N_2 by Roget's Thesaurus. This model performs worse than the purely lexical model; the explanation given is that the associations between constituent nouns and prepositions are primarily ones of lexical collocation, which are certainly influenced by semantics but are also affected by non-semantic factors.

uments, but it shares the fundamental notion of a representation level above the token level. In the training set of protein-protein-document triples, each document sentence mentioning both proteins is labelled with the relation supported by the document as a whole, whether or not the sentence itself is evidence for the relation. For the test triples, sentence-level classifiers are used to predict the relation label of each sentence in the test document and these predictions are combined to assign a label to the entire document. The best-performing combination method is a majority vote strategy – the document is labelled with the relation assigned to the most sentences. Bunescu and Mooney's (2007) model for extracting relational pairs is similar to that of Rosario and Heart, the main differences being the use of binary labels, a purely type-level approach without the intermediate document representation, and the combination method (predict +1 for a pair if any sentence is labelled +1) dictated by the multiple-instance learning assumption.

Turney and Littman (2005) and Turney (2006) develop a framework based on *joining terms* which can be used to construct co-occurrence vectors for each noun pair (N_1, N_2). Unlike the co-occurrence types used when computing lexical similarity, these joining terms are only counted when they co-occur between the two constituents of a pair. Turney and Littman (2005) use 64 joining terms such as *after*, *at*, *of the* and *like*; these are used to generate query strings N_1 *after* N_2, N_2 *after* N_1, N_1 *at* N_2, N_2 *at* N_1,... which are submitted to a Web search engine. The counts returned by the search engine define a 128-dimensional co-occurrence vector which can be compared to other vectors through standard distributional similarity measures (in this case, the cosine measure). Applying this technique to a SAT analogy test and to Nastase and Szpakowicz' (2003) compound noun dataset, Turney and Littman achieve reasonable performance outperforming standard WordNet-based methods with a nearest-neighbour classifier. Nulty (2007a) has implemented a co-occurrence-based model based on joining terms similar to that of Turney and Littman, and his results on the same compound noun dataset indicate that using a support vector machine classifier instead of the nearest-neighbour classifier is an efficient way of boosting performance.

Turney (2006) builds on this work by introducing a method he calls *Latent Relational Analysis (LRA)*. In LRA, two generalisation steps are performed to extend the recall and the richness of the co-occurrence model. Firstly, an automatically constructed semantic thesaurus is used to generate new query pairs; secondly, the set of co-occurrence patterns is not limited to prespecified joining terms, but is based on the contexts observed in a large corpus for each query pair (in a manner similar to Brin (1998) and Agichtein and Gravano's (2000) relation extraction systems). To prevent the system overgeneralising, unreliable pairs and patterns are filtered out after each stage. In a third processing step, the co-occurrence matrix is mapped onto a maximally information-preserving linear subspace spanned by its principal eigenvectors. As in Latent Semantic Analysis (Landauer and Dumais, 1997), this dimensionality reduction step should enhance the robustness of the co-occurrence patterns by identifying combinations of features that best explain the observed distribution of data. The LRA method has very high space and time requirements, but it does perform significantly better than the basic vector space model of Turney and Littman (2005). Turney (2008) introduces a simpler method called *PairClass*, which is similar to LRA but does not use thesaurus-based query expansion or dimensionality reduction and replaces the nearest-neighbour classifier with a Gaussian-kernel support vector machine. PairClass is more efficient than LRA but does not achieve the same level of performance.

Although SemEval Task 4 (Section 6.3.2) is formulated as a token-level problem of classify-

ing whether or not a specified relation holds in a given sentence, a number of the task competitors successfully used a type-level joining terms approach. Nulty (2007b) combines the method used in Nulty (2007a) with additional features based on the WordNet hierarchy. Nakov and Hearst (2007b) apply a joining terms approach based on co-occurring verbs and prepositions. Adopting an event-based conception of relational semantics, Nakov and Hearst generate a corpus from the snippets returned by Google for queries such as N_1 *that* * N_2, N_2 *that* * N_1, N_1 * N_2 and N_2 * N_1, where the * symbol can match up to 8 intervening words. All verbs, prepositions, verb-preposition pairs and conjunctions taking N_1 and N_2 as arguments are counted and the resulting co-occurrence vectors are weighted with the Dice measure. A nearest-neighbour classifier using this representation achieved 67.0% accuracy and 65.1% on the SemEval task, which was the best performance in the competition achieved by a WordNet-free method. Nakov (2007) applies the same method to Nastase and Szpakowicz' (2003) dataset, achieving close to the accuracy of Turney's (2006) LRA but without the need for extensive computational resources.

A related problem that is defined in type-level terms is the task of *relation discovery* (Hasegawa et al., 2004). In this unsupervised learning scenario, the goal is to discover the distribution of relations that can obtain between two nouns of given types. A clustering framework and ACE entity types are standardly used; a sample task is to cluster the contexts that appear between entities of type PERSON and GEOPOLITICAL ENTITY. Although the motivation for this task is to let the data guide the type of relations that are discovered, it has been necessary to evaluate performance by taking the ACE relations as a gold standard. This implies that the development of relation discovery techniques will be biased towards those that discover a particular kind of relation. In their paper defining the task, Hasegawa et al. use a simple bag-of-words representation and a hierarchical clustering method. Hachey (2006) compares the effects of various distance measures and dimensionality reduction algorithms, while Chen et al. (2006b) demonstrate that a graph-based spectral clustering technique achieves superior performance. In view of the long-running debate on the range of relations that can underlie compound nouns (discussed in Chapter 2), it would be interesting to perform a relation discovery analysis of attested compound constituent pairs. To my knowledge, this has not yet been investigated.

Davidov and Rappoport (2008) use relation clustering as part of a supervised classification system. They extract a large set of relation-encoding lexical patterns for a randomly sampled vocabulary and cluster these patterns to discover prototypical semantic relationships. These clusters are used to generate features for word pairs based on the distribution of patterns observed for each individual word pair; these features can then be used for classification with standard tools such as support vector machines. Davidov and Rappoport show that their method works very well on the SemEval Task 4 dataset, achieving state-of-the-art results (see Section 6.3.2).

5.4 Kernel methods and support vector machines

This section gives a brief theoretical overview of kernels and classification with kernel machines, which are the tools used in the learning experiments of Chapters 6 and 7. More comprehensive general treatments of these topics can be found in the tutorial by Burges (1998) or the book by Shawe-Taylor and Cristianini (2004).

5.4.1 Kernels

A *kernel* is a function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_+$ which is equivalent to an inner product $\langle \cdot, \cdot \rangle$ in some inner product space $\mathcal{F}$ (the *feature space*):[19]

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}} \tag{5.17}$$

where ϕ is a mapping from $\mathcal{X}$ to $\mathcal{F}$. For example, the polynomial kernel of degree $l = 3$, $k_{P3}(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + R)^3$ defined on input vectors of dimension $d = 2$ corresponds to an inner product in the space whose $\binom{d+l}{l} = 10$ dimensions are the monomials of degree i, $1 \leq i \leq 3$, hence $\phi(\mathbf{x}) = \phi(x_1, x_2) = [x_1^3, x_2^3, x_1^2 x_2, x_1 x_2^2, x_1^2, x_2^2, x_1 x_2, x_1, x_2]$. Given a set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathcal{X}\}$ and a kernel k, the $n \times n$ matrix K with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is called the *kernel matrix* or *Gram matrix*. It follows from Mercer's Theorem (Mercer, 1909) that a valid kernel on a set $\mathcal{X}$ is defined by any symmetric finitely positive semi-definite function, i.e., a function for which the Gram matrix of function values calculated on any finite set $X \subseteq \mathcal{X}$ satisfies

$$\mathbf{v}' K \mathbf{v} \geq 0, \forall \mathbf{v} \in \mathbb{R}^n \tag{5.18}$$

We therefore do not need to state the feature mapping ϕ in order to use a kernel (though it may sometimes be informative to do so), so long as the positive-definite property can be proven. This will be useful when we consider derived kernels, where one kernel is defined in terms of another and the associated feature space may be opaque.

An alternative interpretation of kernels arises through defining the feature mapping ϕ as a mapping from elements of $\mathcal{X}$ to functions on $\mathcal{X}$. Specifically, the image $\phi(\mathbf{x})$ of $\mathbf{x}$ is defined as the function $k_{\mathbf{x}}(\cdot) = k(\mathbf{x}, \cdot)$ that gives the value of the kernel function for $\mathbf{x}$ and its argument. We are interested in functions inside the linear span of the images of the items in our dataset X, as the classification function that solves the SVM optimisation problem will be located in this space. Let $f, g \in \mathcal{F}$ be two such functions, so that

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{5.19}$$

$$g(\mathbf{x}) = \sum_{i=1}^{n'} \beta_i k(\mathbf{x}'_i, \mathbf{x}) \tag{5.20}$$

An inner product in this space can be defined as a linear combination of kernel functions:

$$\begin{aligned} \langle f, g \rangle &= \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}'_j) \\ &= \sum_{i=1}^{n} \alpha_i g(\mathbf{x}_i) \\ &= \sum_{i=1}^{n'} \beta_i f(\mathbf{x}'_i) \end{aligned} \tag{5.21}$$

[19] Some authors also consider complex-valued kernels, but this more general definition is not relevant to the methods described here.

This is a valid inner product as it by definition satisfies the conditions of symmetry and bilinearity, and the condition $\langle f, f\rangle \geq 0 \ \forall f \in \mathcal{F}$ is satisfied due to the positive semi-definiteness of the kernel function.

The class of kernels has a number of closure properties that will be useful in subsequent sections. If $k_{\mathcal{X}},\ k_{\mathcal{S}}$ are kernels on sets $\mathcal{X}$ and $\mathcal{S}$ respectively, then all k_{new} satisfying the following definitions are also kernels:

$$k_{new}(\mathbf{x}_i, \mathbf{x}_j) = a k_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j) \ \forall a \in \mathbb{R}_+ \tag{5.22}$$

$$k_{new}(\{\mathbf{x}_i, \mathbf{s}_i\}, \{\mathbf{x}_j, \mathbf{s}_j\}) = k_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j) + k_{\mathcal{S}}(\mathbf{s}_i, \mathbf{s}_j) \tag{5.23}$$

$$k_{new}(\{\mathbf{x}_i, \mathbf{s}_i\}, \{\mathbf{x}_j, \mathbf{s}_j\}) = k_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j) k_{\mathcal{S}}(\mathbf{s}_i, \mathbf{s}_j) \tag{5.24}$$

The notation $\{\mathbf{x}_i, \mathbf{s}_i\}$ in (5.23) and (5.24) reflects the fact that we may wish to combine kernels defined on different sets, in which case the composite kernel is properly described as a kernel on the Cartesian product $\mathcal{X} \times \mathcal{S}$. (5.22) and (5.23) follow from the positive semi-definite property $\mathbf{v}'K\mathbf{v} \geq 0 \ \forall \mathbf{v} \in \mathbb{R}^n$ and standard laws of commutativity and distributivity. Positive semi-definiteness of the kernel in (5.24) is a consequence of the fact that the pointwise product (also called the Schur or Hadamard product) of two positive semi-definite matrices is also positive semi-definite (Schur, 1911). Further details of these proofs are given by Shawe-Taylor and Cristianini (2004), p. 75.[20]

A further piece of mathematical theory which will prove useful below concerns the class of *negative semi-definite kernels*. These are symmetric functions $\tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for all $n \times n$ finite sets $X \subseteq \mathcal{X}$ and for all vectors $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{R}^n$ with $\sum_i \mathbf{v}_i = 0$

$$\mathbf{v}'\tilde{K}\mathbf{v} \leq 0 \tag{5.25}$$

Whereas positive semi-definite kernels correspond to inner products in a Hilbert space $\mathcal{F}$, negative semi-definite kernels correspond to squared distances. In particular, if $\tilde{k}(\mathbf{x}, \mathbf{x}) = 0$ then $\sqrt{\tilde{k}}$ is a semi-metric in the feature space and if also $\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = 0$ only when $\mathbf{x}_i = \mathbf{x}_j$ then $\sqrt{\tilde{k}}$ is a metric (Schoenberg, 1938).[21] If a function k is positive semi-definite, then $-k$ is negative semi-definite, but the converse does not hold.[22] However, Berg et al. (1984) describe two simple methods for inducing a positive semi-definite function k from negative semi-definite $\tilde{k}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\alpha \tilde{k}(\mathbf{x}_i, \mathbf{x}_j)),\ \alpha > 0 \tag{5.26a}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tilde{k}(\mathbf{x}_i, \mathbf{x}_0) + \tilde{k}(\mathbf{x}_j, \mathbf{x}_0) - \tilde{k}(\mathbf{x}_i, \mathbf{x}_j) - \tilde{k}(\mathbf{x}_0, \mathbf{x}_0),\ \mathbf{x}_0 \in \mathcal{X} \tag{5.26b}$$

The point $\mathbf{x}_0$ in (5.26b) can be viewed as providing an origin in $\mathcal{F}$ that is the image of some point in the input space $\mathcal{X}$ (Schölkopf, 2000). When we come to using kernels

[20] To be precise, Shawe-Taylor and Cristianini give proofs for the cases $k_{new}(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j) + k_2(\mathbf{x}_i, \mathbf{x}_j)$ and $k_{new}(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j)k_2(\mathbf{x}_i, \mathbf{x}_j)$, where the kernels to be combined are defined on the same set and take the same arguments. As the closure proofs depend only on the positive semi-definiteness property satisfied by all kernel matrices they also apply to the more general cases stated here.

[21] It is desirable to use a metric distance in most cases where one is available. Hein et al. (2005) observe that kernels derived from semi-metrics assume an invariance in $\mathcal{X}$, for example the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$ treats each point and its reflection through the origin as identical. If $\mathcal{X}$ does not possess the assumed invariance and points assumed identical may actually belong to different classes, then SVM classification performance will suffer. Non-metrics may also be negative semi-definite, but as they do not satisfy the triangle inequality their use may lead to counter-intuitive results.

[22] Negated negative semi-definite functions are sometimes called *conditionally positive semi-definite functions*; they constitute a superset of the positive semi-definite functions.

derived by (5.26b) for SVM classification, the choice of $\mathbf{x}_0$ has no effect on the solution found (Hein et al., 2005), and it is usually convenient to set it to the zero element (where $\mathcal{X}$ has such an element). A familiar example of these transformations arises for $\mathcal{X} = \mathbb{R}^n$ if we take $\tilde{k}$ to be the squared Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \sum_l (x_{il} - x_{jl})^2$. Applying (5.26a) we derive the Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\alpha\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. Applying (5.26b) and setting $\mathbf{x}_0$ to be the zero vector, we obtain a quantity that is twice the linear kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \sum_l x_{il} x_{jl}$.

Kernel functions had been studied in the mathematical literature since the early 20th century but were first applied to machine learning problems by Aizerman et al. (1964), who combined a Gaussian kernel with a perceptron classifier. Their recent popularity is primarily due to their very successful use in classification with maximum margin classifiers (Boser et al., 1992) and subsequently in a variety of pattern recognition applications. One advantage of kernel methods is that they allow linear classifiers to learn non-linear classification functions through a mapping to a space of higher or even infinite dimension, without the requirement that the higher-dimensional mappings be explicitly represented. A further advantage is that efficient and well-understood methods for vectorial classification can be applied to non-vectorial objects such as strings, trees and sets by defining a kernel on those objects; the kernel function definition does not place any restrictions on the nature of the input space $\mathcal{X}$.

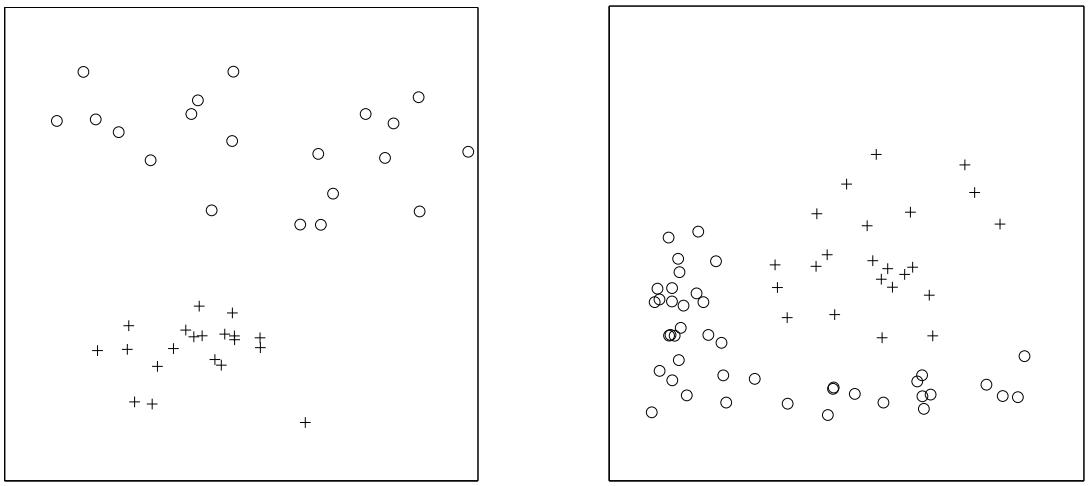
5.4.2 Classification with support vector machines

Given training data $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ drawn identically and independently from some set $\mathcal{X}$, with corresponding labels $Y = \{y_1, \ldots, y_n\}$ belonging to a set $\mathcal{Y}$, the *supervised classification* task is to learn a function $f(x) : \mathcal{X} \mapsto \mathcal{Y}$ that best predicts the unknown label y_j for an as yet unseen data point $\mathbf{x}_j \in \mathcal{X}$. The notion "best predicts" can be formalised in terms of a loss function $L(y, f(\mathbf{x}))$ that quantifies the penalty incurred by assigning the label $f(\mathbf{x})$ to an item which actually has the label y. A standard loss function for classification is the *zero-one loss*, which as the name suggests takes value 0 in the case of misclassification ($f(\mathbf{x}) \neq y$) and value 1 when the predicted label is correct ($f(\mathbf{x}) = y$). The best solution to the classification problem is the function that minimises the generalisation error Err, the expectation of the loss function over all possible new points in $\mathcal{X}$:

$$\mathrm{Err} = E_{\mathcal{X},\mathcal{Y}} L(y, f(\mathbf{x})) = \int_{\mathcal{X},\mathcal{Y}} p(\mathbf{x}, y) L(y, f(\mathbf{x}))\, \mathrm{d}\mathbf{x}\, \mathrm{d}y \tag{5.27}$$

In practice, of course, finding an optimal classifier is a complicated task. Firstly, the data density $p(\mathbf{x}, y)$ is generally unknown so the expectation in (5.27) cannot be calculated exactly. A common strategy is to separate the data into a training set and a test set. The training set is used to select the classifier function $f(\mathbf{x})$. The test set is then used to estimate the generalisation error of $f(\mathbf{x})$ on unseen data. Another strategy, called *k-fold cross-validation*, is to split the data in k different ways and estimate the generalisation error from the average test error of the k splits; this can give better estimates of Err than a single train-test split when the size of the dataset is small.

A second complication is that it is impossible to consider all functions consistent with a finite training set (there are infinitely many), and training consistency may not be a guarantee of generalisation performance when the function overfits the data. Typically, the set of functions considered is restricted to a particular functional class based on its



(a) Separable (b) Non-separable

Figure 5.1: Linearly separable and non-separable datasets

complexity, i.e., its ability to fit many different patterns of data, and on prior knowledge about the structure of the problem. One such class is the class of *linear functions*:

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{5.28}$$

Well-known classifiers with this form include the perceptron (Rosenblatt, 1958) and linear regression of X on Y. The absolute value of $g(\mathbf{x})$ is proportional to the distance from the point $\mathbf{x}$ to the hyperplane $b + \sum_{i=1}^{d} w_i x_i = 0$. The sign of $g(\mathbf{x})$ indicates on which side of the hyperplane it is located; hence, an appropriate classification rule is $f(\mathbf{x}) = \text{sign}(g(\mathbf{x}))$ for binary classes, i.e., when $\mathcal{Y} = \{-1, 1\}$. The quantity $y_i g(\mathbf{x}_i)$ is positive if $g(\mathbf{x}_i)$ is correctly classified, as y_i and $g(\mathbf{x}_i)$ then have the same sign.

Ideally, a classification algorithm should find a hyperplane that divides the training data perfectly in accordance with their class labels. However, for many problems of interest it is not possible to separate the classes with a linear decision boundary. Difficulties also arise when the data *is* linearly separable, as there are then infinitely many separating hyperplanes. These two scenarios are illustrated in Figure 5.1. The *support vector machine* or *SVM* (Cortes and Vapnik, 1995) is a classifier that addresses both problems. This classifier finds a solution for non-separable data by tolerating a number of misclassified training examples; it therefore learns a *soft* decision boundary. It also finds the optimal separating hyperplane in the sense of maximising the distance of both classes from the hyperplane.[23] The SVM solution is defined by the optimisation problem

$$\begin{aligned} \min_{\mathbf{w},b,\gamma,\boldsymbol{\xi}} \quad & -\gamma + c \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \geq \gamma - \xi_i, \\ & \xi_i \geq 0 \text{ for } i = 1, \ldots, n, \text{ and } \|\mathbf{w}\|^2 = 1 \end{aligned} \tag{5.29}$$

The quantity γ in (5.29) is the *margin*, the smallest observed value of $y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) + \xi_i$. The quantities ξ_i are the *slack variables* corresponding to how far the points $\phi(\mathbf{x}_i)$ are

[23] This is not the only way of defining an optimal solution; for example, the Bayes point machine classifier (Herbrich et al., 2001) estimates the average of all separating hyperplanes.

allowed to be closer than γ to the hyperplane. $\xi_i > \gamma$ implies that $\phi(\mathbf{x}_i)$ may be on the "wrong" side of the decision boundary; in this way, misclassifications of the training data are tolerated. The parameter c controls the tradeoff between maximising the margin and tolerating errors; an increase in c entails an increase in the cost of non-zero slack variables.

The methods used to optimise (5.29) will not be detailed here, but some important points will be noted.[24] By introducing Lagrange multipliers a dual objective function W can be obtained that has the same solution as (5.29) but is simpler to optimise. The learning problem is to find the item weight vector $\boldsymbol{\alpha}^*$ that maximise the objective $W(\boldsymbol{\alpha})$:

$$\begin{aligned} \boldsymbol{\alpha}^* = \operatorname*{argmax}_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = -\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)\phi(\mathbf{x}_j) \qquad (5.30) \\ \text{subject to } \sum_{i=1}^{n} y_i \alpha_i = 0,\ \sum_{i=1}^{n} \alpha_i = 1, \\ 0 \le \alpha_i \le C, \text{ for all } i = 1, \ldots, l \end{aligned}$$

The vector of coordinate weights $\mathbf{w}$ is given by $\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i^* \phi(\mathbf{x}_i)$, a linear combination of the training examples. The slack variables ξ_i have disappeared in the dual formulation, but their effect is to limit the permitted size of the item weights through the constraint $\alpha_i \le c$ (the *box constraint*). The solution to (5.30) has a number of pleasing properties. It is a convex optimisation problem, and the solution found will always be a global optimum. Different optimisation methods and repeated runs of the same method are guaranteed to give the same answer, modulo stopping tolerances. The vector of item weights $\boldsymbol{\alpha}^*$ will be sparse, in that many of the values will be 0. This can be seen from a necessary property of the SVM solution:[25]

$$\alpha_i^* [y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - \gamma + \xi_i] = 0, \text{ for all } i = 1, \ldots, l \qquad (5.31)$$

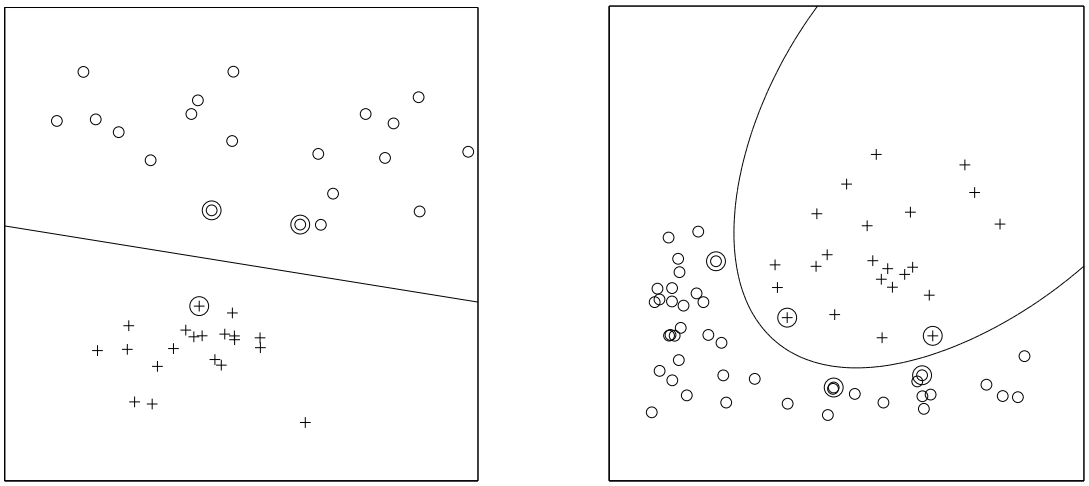
It follows that $\alpha_i^* > 0$ only when $y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = \gamma - \xi_i$; points falling on the correct side of the margin do not contribute to the solution $\boldsymbol{\alpha}^*$. Those points with non-zero α_i^* are known as the *support vectors*.

The dual objective function $W(\boldsymbol{\alpha})$ in (5.30) depends on the training examples $\mathbf{x}_i$ only through their inner products $\phi(\mathbf{x}_i)\phi(\mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\mathcal{F}}$. This suggests that we can use support vector machines to do classification in a kernel feature space by rewriting the objective as

$$W(\boldsymbol{\alpha}) = -\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \qquad (5.32)$$

[24] The derivation of the Lagrangian dual and the SVM objective function (5.30) is stated in many texts, including Hastie et al. (2001) and Shawe-Taylor and Cristianini (2004). The problem is often reformulated as that of minimising the norm of the weight vector w with the functional margin γ set to 1, yielding a dual objective slightly different to that of (5.30). However, the solutions obtained by the two formulations are equivalent up to rescaling.
Specialised methods have been developed for solving the SVM optimisation quickly, including sequential minimal optimisation (Platt, 1999); Bottou and Lin (2007) is a recent survey of this area.

[25] The equality (5.31) belongs to the *Karush-Kuhn-Tucker (KKT) conditions* (Karush, 1939; Kuhn and Tucker, 1951) for the SVM optimisation problem. The set of KKT conditions for any convex optimisation state necessary and sufficient properties of the optimal solution. They are described in most textbooks on optimisation, e.g. Baldick (2006).



(a) Linear kernel (b) Gaussian kernel

Figure 5.2: Decision boundaries found by an SVM with linear kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \sum_l x_{il} x_{jl}$ and non-linear Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2)$. The circled points are the support vectors.

where k is a valid kernel on $\mathcal{X}$.[26] The SVM optimising this objective will learn a decision function $f(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b)$ that is linear in the codomain $\mathcal{F}$ of ϕ but nonlinear in the input space $\mathcal{X}$. Figure 5.2 shows examples of SVM solutions with linear and non-linear mappings.

Support vector machines are binary classifiers that assign one of two labels to a test point according to which side of the decision hyperplane $g(\mathbf{x}) = 0$ it lies on. However, many classification tasks involve more than two labels. In these cases it is possible to modify the SVM objective function (5.30) in order to obtain a true multiclass classifier (Vapnik, 1998; Crammer and Singer, 2001; Lee et al., 2004), but such solutions typically yield more complex optimisation problems and can have very large time requirements (Hsu and Lin, 2002). A simpler and more popular approach is to train a number of standard SVM classifiers on binary subtasks and integrate their predictions on each test example to give a multiclass prediction. In the *one-against-all* approach one binary SVM is learned for each of the K classes. The training data for the class k classifier consists of all training examples, with examples belonging to class k given the label 1 and examples of all other classes given the label -1. The predicted label for a test example $\mathbf{x}_i$ is then the largest of the k decision values:

$$f(\mathbf{x}_i) = \underset{k}{\text{argmax}} \langle \mathbf{w}_k, \phi(\mathbf{x}_i) \rangle + b_k \tag{5.33}$$

where $(\mathbf{w}_k, b_k)$ define the solution of the class-k classifier. This corresponds to assigning the label which is predicted most confidently for $\mathbf{x}_i$ (or in the case where none of the classifiers give a positive prediction, the label that gives the least negative prediction). Another popular method is *one-against-one*, whereby a binary SVM is trained for each pair of labels. This involves $K(K-1)/2$ binary classifiers. Each i-against-j classifier is trained on just the subset of training examples belonging to class i or j. For a test example the prediction of each binary classifier is counted as a vote for either i or j, and the class with the most votes is predicted for that example.

[26] If k is not a kernel but rather some function which is not positive semi-definite, the SVM is not guaranteed to converge. However, good results can sometimes be achieved with such functions and a geometric interpretation of the resulting classifier has been provided by Haasdonk (2005).

Both one-against-all and one-against-one methods have advantages. The one-against-one method can be quicker to train as the training dataset for each classifier is smaller than the entire training set for the problem, and it can learn more complex distinctions between classes. However, the smaller training set sizes for one-against-one can be a disadvantage when little training data is available or the number of classes is large, as there may not be enough data for the i-against-j classifiers to learn reliable decision rules; one-against-all classification should be more robust to the data size factor. Hsu and Lin (2002) recommend one-against-one over one-against-all, but this is largely on the basis of training time efficiency, as their experimental results show little difference in classification performance. In contrast, Rifkin and Klautau (2004) mount a robust defence of one-against-all, claiming that it is at least as accurate as other methods when comparison experiments are carried out rigorously.

5.4.3 Combining heterogeneous information for classification

It is a common scenario that different sources of information are available for tackling a problem, where each source captures a different aspect or "view" of the data. In such cases, it is often useful to combine the information sources in a way that produces an integrated view more suitable for the task than any of the individual views. For example, combining lexical and contextual information can give improved performance on word sense disambiguation (Gliozzo et al., 2005), and complementarily exploiting bag-of-words and syntactic representations is an effective approach to question and answer classification (Moschitti et al., 2007). As described above, combining different levels of syntactic and lexical information is a standard methodology for relation classification. The idea of combination can be realised in many ways in many classification frameworks; here I focus on kernel methods, which provide simple and flexible tools for my purposes. Equations (5.23) and (5.24) state that the sum or product of two positive semi-definite kernel functions is itself a positive semi-definite kernel. One advantage of these kernel combination methods is that facilitates the integration of heterogeneous data representations: we can combine kernels on vectors, kernels on sets, kernels on strings or any other kind of kernel. In Chapter 7 I show how combining lexical and relational information by summing lexical and relational similarity kernels leads to improved performance on the tasks of compound noun interpretation and relation identification.

When is kernel combination beneficial? Joachims et al. (2001) relate the effect of combining kernels to the resulting change in the item margin $g(x_i)$ for each member x_i of a given training set. If the sets of support vectors induced by the two individual kernels on the training set have a low degree of overlap, the item margins will tend to be larger for the combined kernel than the individual kernels. So long as the individual kernels have similar training error, classification performance should be improved. If the training error of the individual kernels are very different, combination tends to give an intermediate level of performance.

Cristianini et al. (2001) describe a similarity measure between kernels which they call *alignment*. The true alignment A between two kernels is an inner product on functions that is not calculable in practice, but it can be estimated by the empirical alignment $\hat{A}$:

$$\hat{A}(k_1, k_2) = \frac{\langle K_1, K_2 \rangle_F}{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F} \tag{5.34}$$

where $\langle K_1, K_2 \rangle_F = \sum_{i=1}^{n} \sum_{j=1}^{n} K_{1ij} K_{2ij}$ is the Frobenius inner product between the Gram matrices K_1 and K_2. By considering the target matrix $K_{\mathbf{y}}$ defined as the outer product of the label vector $\mathbf{y}$, i.e., $K_{\mathbf{y}ij} = \text{sign}(y_i = y_j)$, the empirical target alignment $\hat{A}(K_1, K_{\mathbf{y}})$ quantifies the degree to which the representation of the data induced by k_1 matches the "true" label-derived clustering of the data. The utility of a high target alignment for SVM learning is elucidated by its interpretation as a measure of between-class distance with respect to the true classes $\mathbf{y}$ (Xiong et al., 2005). Among the applications of the alignment measure is a criterion for optimal kernel combination; Cristianini et al. show that the target alignment of a summed kernel $k_+ = k_1 + k_2$ will be increased if k_1 and k_2 are both well-aligned with the target $k_{\mathbf{y}}$ but not aligned with each other. Both this analysis and that of Joachims et al. (2001) highlight the intuition that a suitable guiding principle for kernel combination is to seek kernels that capture distinct but discriminative information about the data.

It follows from the property of closure under scaling (5.22) that the contributions of the individual kernels in a sum can be weighted differentially. That is, any linear combination of kernels is a valid kernel:

$$k_+(\mathbf{x}_i, \mathbf{x}_j) = \sum_l \mu_l k_l(\mathbf{x}_i, \mathbf{x}_j) \tag{5.35}$$

with an appropriate extension for kernels defined on different sets as in (5.23). Joachims et al. (2001) state that the weight parameters μ_l are relatively unimportant, and my own trial experiments have also indicated that the effects of optimising these are generally smaller than those of optimising other parameters. In order to reduce the number of parameters to be estimated in training, the experiments with combined kernels described in Chapter 7 will use equal weights throughout.

5.5 Conclusion

The unifying theme of this chapter has been the importance of similarity in relational classification tasks. Consonant with the hypothesis that semantic relations can be identified through a process of analogy, I have given an overview of prior NLP research on lexical and relational similarity that can be exploited to define measures of similarity between pairs of nouns. Positive semi-definite kernel functions can also be seen as similarity functions whose mathematical properties ensure that they can be used for classification with support vector machines. In the next two chapters I bring these two themes together by introducing kernel methods that are suitable for implementing models of constituent (Chapter 6) and relational (Chapter 7) similarity.

Chapter 6

Learning with co-occurrence vectors

6.1 Introduction

In this chapter I develop a model that uses lexical distributional similarity to perform semantic classification. Working in a kernel framework (Section 5.4), I describe a family of kernel functions on co-occurrence distributions whose connection to well-known lexical similarity measures strongly suggests their appropriateness for semantic tasks. These *distributional kernels* perform very well on compound noun interpretation and semantic relation identification datasets, attaining state-of-the-art results on both. In Section 6.8 I consider explanations for the superior performance of distributional kernels compared to the popular linear and Gaussian kernels derived from the L_2 distance. I propose that it can be related to the distributional kernels' robustness to large variances in co-occurrence type marginal distributions, and demonstrate that the application of a suitable co-occurrence reweighting function can sometimes allow the L_2 kernels to approach the performance level of the distributional kernels by emulating this robustness.

6.2 Distributional kernels for semantic similarity

Good performance with support vector machines is dependent on the choice of a suitable kernel. If a kernel function induces a mapping into a feature space where the data classes are well separated, then learning a decision boundary in that space will be easy. Conversely, if the feature space mapping of the data does not contain discriminative information, SVM classification will perform poorly. Hence if we can use a kernel function tailored to the prior knowledge we have about our classification problem, we expect to do better than we would with a less appropriate kernel. In NLP, the development of new kernels has tended to focus on kernels for structured linguistic data such as strings, trees and graphs. For classification with numerical features the standard linear, polynomial and Gaussian kernels are almost always used. As described in Section 5.4.1, the linear and Gaussian kernels are related to the Euclidean L_2 distance, yet this distance has been shown by Lee (1999) and others to perform relatively poorly when applied to distributional similarity. It therefore seems worthwhile to investigate other kernels for learning with co-occurrence distributions.

The starting point I take is a parameterised family of functions on positive measures described by Hein and Bousquet (2005). Building on work by Topsøe (2003) and Fuglede

(2005), Hein and Bousquet define the function $d^2_{\alpha|\beta} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ as

$$d^2_{\alpha|\beta}(x_i, x_j) = \frac{2^{\frac{1}{\beta}}(x_i^\alpha + x_j^\alpha)^{\frac{1}{\alpha}} - 2^{\frac{1}{\alpha}}(x_i^\beta + x_j^\beta)^{\frac{1}{\beta}}}{2^{\frac{1}{\alpha}} - 2^{\frac{1}{\beta}}} \tag{6.1}$$

for any $\alpha \in [1, \infty]$, and any $\beta \in [\frac{1}{2}, \alpha]$ or $\beta \in [-\infty, -1]$. All such functions are shown to be squared Hilbertian metrics on $\mathbb{R}_+$, i.e., $d_{\alpha|\beta}$ is a metric distance that can be isometrically embedded in a Hilbert space (which is not true of all metrics). $d_{\alpha|\beta}$ also has the property of $\frac{1}{2}$-*homogeneity*, meaning that $d_{\alpha|\beta}(cx_i, cx_j) = c^{\frac{1}{2}} d_{\alpha|\beta}(x_i, x_j)$ for all $c \in \mathbb{R}_+$. Using the pointwise distance $d^2_{\alpha|\beta}$ as a building block, a distance function on probability distributions $D^2_{\alpha|\beta} : \mathcal{M}^1_+(C) \times \mathcal{M}^1_+(C) \to \mathbb{R}$ can be constructed by integrating over the event space C.[1] Where the probability distributions are discrete, as in the case of co-occurrence distributions, this entails a simple pointwise summation:

$$D^2_{\alpha|\beta}(P, Q) = \sum_{c \in C} d^2_{\alpha|\beta}(P(c), Q(c)) \tag{6.2}$$

$D^2_{\alpha|\beta}$ is also a squared Hilbertian metric. Due to the $\frac{1}{2}$-homogeneity of $d^2_{\alpha|\beta}$, $D^2_{\alpha|\beta}$ is independent of the dominating measure on the event space C and hence invariant to bijective transformations of that space. Hein and Bousquet argue that this is a useful property in the context of image histogram classification, where colours $c \in C$ can be represented in one of a number of colour spaces and $D^2_{\alpha|\beta}$ is invariant to the choice of space.

Different values for the parameters α and β give different functions $d^2_{\alpha|\beta}$ and $D^2_{\alpha|\beta}$, including some distances on distributions that are well known in the statistical and information theoretical literature. In particular, $D^2_{\infty|1}$ is the L_1 distance, $D^2_{1|1}$ is the Jensen-Shannon divergence and $D^2_{\frac{1}{2}|1}$ is the Hellinger divergence.[2] As these are squared Hilbertian metrics, it follows from a theorem of Schoenberg (1938) that they are also negative semi-definite kernel functions. Thus, equations (5.26a) and (5.26b) provide a means of deriving positive semi-definite kernels from these distances, in the same way that the standard linear and Gaussian kernels are derived from the squared L_2 distance. Table 6.1 lists the distributional distances considered in this thesis and the positive semi-definite kernels obtained by using (5.26b) with the origin $\mathbf{x}_0$ set to the zero measure. These kernels will be called the *linear* kernels for the corresponding distance. The *RBF* kernels will be those obtained through (5.26a), i.e., $k_{rbf}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\alpha D^2(\mathbf{x}_i, \mathbf{x}_j))$.[3] For example, the L_1 RBF kernel is defined as:

$$k_{L_1\_RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\alpha \sum_c |P(c|w_1) - P(c|w_2)|\right) \tag{6.3}$$

[1] $\mathcal{M}^1_+(C)$ is the space of positive measures summing to 1 on some set C, i.e., the space of probability distributions on C.

[2] These distributional distances also belong to the family of *f-divergences*, whose behaviour is well-studied in the information theory literature (Liese and Vajda, 2006), and which mutually define upper and lower bounds on each other (Topsøe, 2000). Furthermore, they share important geometric properties; in particular, they yield slightly different approximations to the Fisher information metric (Rao, 1987). For these reasons, it is to be expected that kernels derived from these distances will exhibit broadly similar performance.

[3] The use of α to denote both the RBF kernel width in (5.26a) and one of the parameters in Hein and Bousquet's function definition (6.2) may be confusing. In the remainder of this thesis, α will denote the kernel width exclusively.

Distance	Definition	Derived linear kernel
$(L_2$ distance$)^2$	$\sum_c (P(c\|w_1) - P(c\|w_2))^2$	$\sum_c P(c\|w_1)P(c\|w_2)$
L_1 distance	$\sum_c \|P(c\|w_1) - P(c\|w_2)\|$	$\sum_c \min(P(c\|w_1), P(c\|w_2))$
Jensen-Shannon divergence	$\sum_c P(c\|w_1)\log_2(\frac{2P(c\|w_1)}{P(c\|w_1)+P(c\|w_2)}) + P(c\|w_2)\log_2(\frac{2P(c\|w_2)}{P(c\|w_1)+P(c\|w_2)})$	$-\sum_c P(c\|w_1)\log_2(\frac{P(c\|w_1)}{P(c\|w_1)+P(c\|w_2)}) + P(c\|w_2)\log_2(\frac{P(c\|w_2)}{P(c\|w_1)+P(c\|w_2)})$
Hellinger distance	$\sum_c \left(\sqrt{P(c\|w_1)} - \sqrt{P(c\|w_2)}\right)^2$	$\sum_c \sqrt{P(c\|w_1)P(c\|w_2)}$

Table 6.1: Squared metric distances on co-occurrence distributions and derived linear kernels

I will refer to the kernels derived from the L_1 distance, Jensen-Shannon divergence and Hellinger divergence as *distributional kernels*. For consistency I will also refer to the standard linear and Gaussian kernels as the L_2 linear and L_2 RBF kernels respectively.

The suitability of these distributional kernels for semantic classification is suggested by their connections with popular distributional similarity measures. As described in Section 5.3.1.3, the Jensen-Shannon and L_1 distances have been successfully applied as distributional distance measures. The L_1 linear kernel is the same as the difference-weighted token-based similarity measure of Weeds and Weir (2005). Lin (1999) uses the transformation $\text{sim}_{JSD} = 2 - \text{dist}_{JSD}$ to derive a similarity measure from the Jensen-Shannon divergence; this can be shown to equal the Jensen-Shannon linear kernel. Dagan et al. (1999) use a heuristic transformation $\text{sim}_{JSD} = 10^{-\alpha \text{dist}_{JSD}}$; the Jensen-Shannon RBF kernel $k_{JSD\_RBF} = \exp(-\alpha \text{dist}_{JSD})$ provides a theoretically motivated alternative when positive semi-definiteness is required. Thus these proven distributional similarity measures are also valid kernel functions that can be directly used for SVM classification.

Of the other distributional measures surveyed in Section 5.3.1.3, some can be shown to be valid kernels and some can be shown not to be. The cosine similarity is provably positive semi-definite, as it is the L_2 linear kernel calculated between L_2-normalised vectors. Distributional vectors are by definition L_1-normalised (they sum to 1), but there is evidence that L_2 normalisation is optimal when using L_2 kernels for tasks such as text categorisation (Leopold and Kindermann, 2002). Indeed, in the experiments described here the L_2 kernels performed better with L_2-normalised feature vectors. In this case the L_2 linear kernel function then becomes identical to the cosine similarity.

It follows from the definitions of positive and negative semi-definite kernels that non-symmetric measures cannot be kernels. This rules the confusion probability, Kullback-Leibler divergence and α-skew divergence out of consideration.[4] Other similarities, such as that of Lin (1998b), can be shown not to be positive semi-definite by calculating similarity matrices from real or artificial data and showing that their eigenvalues are not all non-negative, as is required by positive semi-definite functions.

[4]Confusingly, Kullback and Leibler (1951) state that the KL divergence is "almost positive definite". However, this seems to be a different usage of the term, the intended meaning being that the value of the divergence is always greater than or equal to zero.

Other kernels on probability distributions or on positive measures (which can be normalised to give distributions) have been proposed by researchers in machine learning. Chapelle et al. (1999) consider kernels of the forms

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_c x_i^a x_j^a \tag{6.4a}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\alpha \sum_c |x_{ic}^a - x_{jc}^a|^b\right) \tag{6.4b}$$

which are positive semi-definite for all a and all $0 \leq b \leq 2$. The family defined by (6.4a) includes the $L_2(a = 1)$ and Hellinger ($a = 0.5$) linear kernels, while that defined by (6.4b) includes the L_2 ($a = 1, b = 2$), L_1 ($a = 1, b = 1$) and Hellinger ($a = 0.5, b = 2$) RBF kernels. Chapelle et al. find that kernels with values of a less than 1 or b less than 2 significantly outperform the L_2 linear and RBF kernels for histogram-based image classification. The Jensen-Shannon RBF kernel is discussed under the name *entropy kernel* by Cuturi et al. (2005), who offer the interpretation that it quantifies the difference between the average of the entropy of two distributions and the entropy of the distributions' average. While Cuturi et al. fail to obtain good performance with the entropy kernel on a digit classification task, they hypothesise that it will perform better when dealing with multinomial distributions. The same authors also describe a kernel based on the *inverse generalised variance* of a measure, which I study in Chapter 7. Generalisations of the Hellinger linear kernel for parametric models, including Gaussian distributions and hidden Markov models, are described by Jebara et al. (2004). Another kind of distributional kernel is introduced by Lafferty and Lebanon (2005), based on a model of heat diffusion on statistical manifolds. This *heat kernel* is not always guaranteed to be positive semi-definite, but the authors report very impressive results on text classification tasks, where the "bag of words" representation naturally defines a multinomial distribution for each document.

6.3 Datasets

6.3.1 1,443 Compounds

The dataset used for compound noun interpretation is derived from the sample of 2,000 BNC noun sequences described in Section 4.3. These include the 700 items that were used in the trial and test stages of the dual-annotator experiment and an additional 1,300 items that were labelled by a single annotator. Ideally, the entire dataset would have been annotated by two annotators, but this was not possible due to constraints on resources and annotator availability. As the data for the dual-annotator experiment were sampled randomly, I have no reason to believe that the agreement results observed for the test data is unrepresentative of the dataset as a whole.

In order to focus on the task of classifying semantic relations, as opposed to identifying non-compositional compounds or extraction errors, only those compounds annotated with one of the six relations *BE*, *HAVE*, *IN*, *INST*, *ACTOR* and *ABOUT* were used in my experiments. This leads to a classification dataset of 1,443 compounds with the class distribution given in Table 6.2. Always choosing the most frequent class (*IN*) gives a baseline accuracy of 21.3%, while the random baseline is 16.7%.

Relation	Frequency	Proportion
BE	191	13.2%
HAVE	199	13.8%
IN	308	21.3%
INST	266	18.4%
ACTOR	236	16.4%
ABOUT	243	16.8%

Table 6.2: Class distribution for the compounds dataset

6.3.2 SemEval Task 4

Task 4 at the 2007 SemEval Competition (*Classification of Semantic Relations between Nominals*) consisted of seven relation identification subtasks (Girju et al., 2007). Each subtask concentrated on a single semantic relation, the goal being to distinguish sentences that express that relation from sentences that do not. The seven relations used were *CAUSE-EFFECT*, *INSTRUMENT-AGENCY*, *PRODUCT-PRODUCER*, *ORIGIN-ENTITY*, *THEME-TOOL*, *PART-WHOLE* and *CONTENT-CONTAINER*. Girju et al. explain that this binary classification framework was chosen over a multiclass framework in order to avoid the inevitable complications involved in constructing a comprehensive set of mutually exclusive semantic relations.[5]

A corpus of example sentences was collected for each relation by submitting targeted queries to the Google search engine. For example, sentences for the *PRODUCT-PRODUCER* relation were obtained through queries such as *the * * produces*, *the * maker* and *this * company*; queries for *CONTENT-CONTAINER* included *the * contains*, *contents of the * included* and *kept in a box*. The sentences returned for these queries were annotated by two annotators, and only sentences on which the annotators agreed were used for the datasets.[6] Average inter-annotator agreement is reported as 70.3%. Because all examples were retrieved with these targeted queries, the negative examples tend to be "near-misses" that narrowly fail to satisfy the definition of the relevant relation. This ensures that the classification task is a challenging one and seems likely to have depressed the inter-annotator agreement figure.

An example of a positive instance of the *CONTENT-CONTAINER* relation is:

> Put <e1>tea</e1> in a <e2>heat-resistant jug</e2> and add the boiling water.

The <e1></e1> and <e2></e2> tags denote the candidate relation arguments. A clearcut negative example for the same relation is:

> <e1>Batteries</e1> stored in <e2>contact</e2> with one another can generate heat and hydrogen gas.

It is obvious that *batteries* and *contact* do not enter into a *CONTENT-CONTAINER* relation in this sentence, as *contact* does not refer to an object. A more involved negative example is:

[5] I have given a taste of these complications in Chapter 3.

[6] The relation definitions, as well as the annotated datasets, are available from `http://nlp.cs.swarthmore.edu/semeval/tasks/task04/data.shtml`.

I am installing lights under the <e1>cabinets</e1> in my <e2>kitchen</e2>.

The annotation for this sentence contains the comment that "cabinets are normally affixed, so this is Part-Whole", referring to condition (3) in the definition of *CONTENT-CONTAINER*: *common sense dictates that X may be removed from Y without significantly changing the nature of Y; more precisely, X is not affixed to Y, nor is it usually considered to be a component of Y.* This contrasts with the following sentence, which seems very similar but is labelled positive:

The <e1>kitchen</e1> holds a <e2>cooker</e2>, fridge, microwave oven, in short: everything you need if you want to prepare a light meal.

Here the annotator has commented "a cooker is not attached (only plugged in), and a kitchen without a cooker is possible, so the definition holds". The distinction here is very subtle, and it may be difficult for any automatic classifier to learn the appropriate behaviour.

The data for each relation consists of 140 training examples and at least 70 test examples. As well as the context sentence and its label, the directionality of the candidate relation and WordNet senses for the two arguments are provided. Girju et al. describe three baselines for performance comparison. Labelling every test example as +1 (*alltrue* baseline) gives a baseline of 48.5% accuracy, 64.8% F-score.[7] Always choosing the majority class in the test set (*majority* baseline) gives 57.0% accuracy, 30.8% F-score. Random guessing in accordance with the label distribution in the test set (*probmatch* baseline) achieves 51.7% accuracy and 48.5% F-score. Most of the systems competing in the SemEval task were able to outperform these baselines, though only 8 of the 24 systems could match or exceed the *alltrue* F-score baseline of 64.8%.[8] Systems are categorised according to whether they used WordNet information and whether they used Google queries.

The overall highest scores were attained by systems using WordNet; the best of these used a variety of manually annotated resources, including WordNet, the NomLex-Plus nominalisation database and thousands of additional annotated example sentences, and achieved 76.3% accuracy, 72.4% F-score (Beamer et al., 2007). The best WordNet-free approach was that of Nakov and Hearst (2007b), whose Web query-based system scored 67.0% accuracy and 65.1% F-score. Very recently, Davidov and Rappoport (2008) have reported impressive results (Accuracy = 70.1%, F-score = 70.6%) achieved with a method based on pattern clustering. The systems of Nakov and Hearst and of Davidov and Rappoport are described in more detail in Section 5.3.2.3.

6.4 Co-occurrence corpora

Two very different corpora were used to extract co-occurrence information: the British National Corpus (Burnard, 1995) and the Web 1T 5-Gram Corpus (Brants and Franz, 2006). The former is a medium-sized corpus of texts manually compiled with a concern for balance of genre; the latter contains frequency counts for n-grams up to length 5 extracted from Google's index of approximately 1 trillion words of Web text. These differences entail different co-occurrence detection methods, as detailed below.

[7] Definitions of the performance measures used for this task are given in Section 6.5.

[8] I count as separate entries those systems which use multiple sets of information sources and for which multiple results are reported, for example results with and without WordNet information.

6.4.1 British National Corpus

As in the compound extraction experiment (Section 4.2), I use the 90 million word written component of the BNC. The corpus was tagged, lemmatised and parsed with the RASP toolkit (Briscoe et al., 2006). The co-occurrence relation I count to extract distributional vectors is the conjunction relation. This relation is a high-precision indicator of semantic similarity between its arguments, and has been successfully used in automatic thesaurus and taxonomy construction (Roark and Charniak, 1998; Widdows and Dorow, 2002). It is not the similarity between conjuncts that is of interest here, but rather the distributional similarity between nouns based on the conjunction arguments observed for each noun in the corpus, as in Caraballo (1999). I demonstrated in Ó Séaghdha (2007a) that conjunction co-occurrence information alone outperforms a number of other relations for compound interpretation. Furthermore, the co-occurrence vectors extracted from conjunction information are very sparse, leading to very quick learning and prediction performance.

Conjunctions are assigned the `conj` grammatical relation (GR) by RASP.[9] This binary relation holds between a conjunction and each of its arguments, rather than between the arguments themselves. For example, the GR output for the sentence *Tom and Jerry chased the dish and the spoon* is:

```
(|ncsubj| |chase+ed:4_VVD| |and:2_CC| _)
(|dobj| |chase+ed:4_VVD| |and:7_CC|)
(|conj| |and:7_CC| |dish:6_NN1|)
(|conj| |and:7_CC| |spoon:9_NN1|)
(|det| |spoon:9_NN1| |the:8_AT|)
(|det| |dish:6_NN1| |the:5_AT|)
(|conj| |and:2_CC| |Tom:1_NP1|)
(|conj| |and:2_CC| |Jerry:3_NP1|)
```

It is straightforward to distribute the dependencies of each conjunction over its conjuncts by adding the appropriate GRs:

```
(|ncsubj| |chase+ed:4_VVD| |Tom:1_NP1| _)
(|ncsubj| |chase+ed:4_VVD| |Jerry:3_NP1| _)
(|dobj| |chase+ed:4_VVD| |dish:6_NN1|)
(|dobj| |chase+ed:4_VVD| |spoon:9_NN1|)
(|conj| |dish:6_NN1| |spoon:9_NN1|)
(|conj| |Tom:1_NP1| |Jerry:3_NP1|)
```

To extract a co-occurrence vector for a noun N_i, we count occurrences of the relation $\texttt{conj}(N_i, N_j)$ where N_j belongs to the target vocabulary and both N_i and N_j are tagged as nouns. The target vocabulary V_c is defined as the 10,000 nouns most frequently entering into a conjunction relation in the corpus; in practice, this restricts the set of

[9] Briscoe et al. (2006) report 72.3% F-score on identifying `conj` relations in the DepBank parser evaluation corpus of 700 annotated Wall Street Journal sentences. It is possible that the quality of the co-occurrence vectors extracted from the BNC would be improved by using a different parser; for example, Clark and Curran (2007) report that their CCG parser attains 78.8% F-score on DepBank `conj` relations. However, one advantage of RASP is that it is unlexicalised and can therefore handle text from diverse sources, as in the BNC, without the need for retraining or other domain adaptation.

admissible co-occurrence types to those occurring at least 42 times in the BNC. Each co-occurrence vector $\mathbf{x}$ is normalised to have either unit L_2 norm ($\sum_i x_i^2 = 1$) or unit L_1 norm ($\sum_i x_i = 1$), for input to the L_2 kernels or distributional kernels respectively. The feature vector for each compound or word pair (N_1, N_2) in the dataset is constructed by appending the normalised co-occurrence vectors of the words N_1 and N_2. The application of the normalisation step before combining the constituent vectors, giving equal weight to each vector, proves to be very important – this step alone accounts for the four-point improvement in results with the L_2 linear kernel and BNC features over those reported in Ó Séaghdha (2007a). A further normalisation procedure is applied to the combined vector, again using L_2 or L_1 normalisation as appropriate to the kernel.

6.4.2 Web 1T 5-Gram Corpus

The Google 5-Gram Corpus (Brants and Franz, 2006) consists of n-gram counts of length up to 5 generated from Google's index of publicly available webpages. This allows us to use frequency data from about a trillion words of text, though we cannot access that text directly. The corpus contains all n-grams that occur 40 times or more in the index, after filtering to remove non-English text and rare words (words with frequency less than 200 are replaced with an <UNK> token). The following sample from the 5-gram section gives a flavour of the data:

```
channel is software compatible with 47
channel is software programmable for 56
channel is somehow unable to 47
channel is sometimes referred to 67
channel is specified ) ; 71
channel is specified , a 48
channel is specified , all 40
channel is specified , that 195
channel is specified , the 194
channel is specified , then 140
```

Because the data do not consist of full sentences, it is not possible to extract grammatical relations through parsing. Instead, I use a more heuristic method similar to the "joining terms" approach of Turney and Littman (2005). This involves searching the corpus for patterns of the form $N_i\ J\ (\neg N)^*\ N_j\ \neg N$, where N_i and N_j are nouns, J is a joining term and $(\neg N)^*$ matches some number (possibly zero) of non-nouns. N_j is not permitted to match the last word in an n-gram, as we cannot know whether it was followed by a non-noun in the original text. If N_i (resp., N_j) is a target word, the cell corresponding to the co-occurrence type (J, N_j) (resp., (J, N_i)) in N_i's co-occurrence vector is incremented by the frequency listed for that n-gram in the corpus. The co-occurrence type representation can be refined by indicating whether the target word comes before or after the joining term, i.e., contexts $N_i\ J\ N_j$ and $N_j\ J\ N_i$ would count as distinct co-occurrence types for a target word N_i. This is done for all joining terms except *and* and *or*, which clearly have symmetric semantics.[10] A noun dictionary automatically constructed from WordNet 2.1

[10]It is arguable that *is* and *like* should also be treated as symmetric, but my intuition is that they often have an asymmetric nature (see also Tversky (1977)). Experiments indicate that it actually makes little difference for these joining terms.

and an electronic version of Webster's 1913 Unabridged Dictionary determines the sets of admissible nouns $\{N\}$ and non-nouns $\{\neg N\}$.[11] To reduce the number of false positive noun matches, such as in *cat and sees the fish* (*see* can be a noun), I use a stop list adapted from van Rijsbergen (1979) that includes the most common falsely identified terms. Webster's dictionary also provides information about irregular plurals, enabling a simple form of lemmatisation by mapping singular and plural forms onto the same co-occurrence type. The following joining terms are used: *and*, *or*, *about*, *at*, *by*, *for*, *from*, *in*, *is*, *of*, *to*, *with*, *like*. As in Section 6.4.1 the co-occurrence vocabulary for each joining term is limited to the 10,000 most frequent co-occurrence types for that term.

I also consider a second kind of co-occurrence pattern based on verbal information. This is in line with event-based theories of semantic relations, in particular frame-based theories of compound interpretation (Section 2.2.2). It is directly inspired by the methods of Nakov (2007) and Nakov and Hearst (2007b) that were discussed in Section 5.3.2.3. For this technique the corpus is searched for n-grams matching N_i *that*$|$*which*$|$*who* V $\neg N$ or N_i *that*$|$*which*$|$*who* V $(\neg N)^*$ N_j $\neg N$, where N_i or N_j is a target word. A dictionary of verbs was created by taking all verbs listed in WordNet 2.1 and Webster's and using the `morphg` morphological generation software of Minnen et al. (2000) to generate inflected forms. For each matching co-occurrence of a target N_i with a verb V, the co-occurrence count for the feature V is incremented by the n-gram frequency, taking into account whether N_i appears before V (assumed subject) or after V (assumed object). For the transitive case where target N_i co-occurs with V and another noun N_j, the count for the combined feature (V, N_j) is also incremented, again distinguishing between subject and object co-occurrences.

6.5 Methodology

All classification experiments were performed with the LIBSVM support vector machine library (Chang and Lin, 2001). The standard LIBSVM implementation was modified to perform one-against-all multiclass classification instead of one-against-one, as the compound dataset is relatively small (Section 5.4.2). For all datasets and all training-test splits the SVM cost parameter c was optimised in the range $(2^{-6}, 2^{-4}, \ldots, 2^{12})$ through cross-validation on the training set. In addition, the width parameter α was optimised in the same range for the RBF kernels. The features were not normalised to have the same range, although this is sometimes recommended (e.g., by Hsu et al. (2008)); feature normalisation was in fact observed to decrease performance. All kernel values were calculated before running the SVM algorithm. This can speed up training and prediction, especially for less efficient kernel functions, and when the dataset is small only a moderate amount of space is required to store the kernel matrix – around 20MB for each 1443×1443 compound task matrix and 500KB for the 210×210 matrix of one SemEval relation, encoded to double precision and compressed with `gzip`.

Classifier performance on the two datasets is measured in terms of accuracy and macro-averaged F-score. Accuracy measures the proportion of items that are classified correctly. F-score complements accuracy by rewarding classifiers that perform well across all relations and balancing out the effect of class distribution skew. I use the standard (F_1)

[11] The electronic version of Webster's is available from `http://msowww.anu.edu.au/~ralph/OPTED/`.

formulation for each class k:

$$F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.5}$$

The precision of a classifier on class k is given by the proportion of items for which k was predicted that were correctly predicted. Recall is given by the proportion of items actually belonging to class k that were correctly predicted. In the compound task, an F-score value is calculated for each of the six classes and the average of those scores is presented. In the SemEval task, an F-score is calculated for each of the seven relations, precision and recall being measured in each case on the positive class only, as is standard in retrieval-style scenarios.

As well as comparing performance on a particular dataset, we can ask what can be inferred about the performance of a classifier on other similarly-sized datasets sampled from the same source. One method for addressing variability in the training set is to use *k-fold cross-validation*: each item in the dataset is assigned to one of k folds having approximately equal size. This gives k different training-testing splits; the i^{th} split is made by keeping the i^{th} fold for testing and training on all other folds. The sample variance or standard error of the classification results across folds gives an indication of the performance range of a given method over different datasets of the same size: the smaller the variance, the more certain that the cross-validation average is a good representation of true performance, i.e., "performance on the compound interpretation task", not just of "performance on this particular dataset with these particular training-test splits". However, using variance to compare classifiers can give misleading results. In the compound interpretation experiments described below, the standard errors observed are frequently quite large (1.5–2.5%) considering the inter-classifier performance differences, and confidence intervals based on these errors will suggest that there is no significant difference between classifiers.[12] Yet this approach discards valuable information by ignoring the fact that the observations for different classifiers are not independent. The results for each classifier are obtained on the same cross-validation folds and a sound comparison should take account of this; if classifier A consistently outperforms classifier B over all folds, this can be taken as strong evidence for classifier A's superiority even if the difficulty of individual folds has high variance.

Dietterich (1998) describes a number of suitable statistical tests for dealing with variance in the training and test data and with randomness in the classification algorithm.[13] To compare two classifiers we can take the difference p_i in accuracy (or F-score) on each cross-validation fold and calculate the paired t-test statistic:

$$t = \frac{\bar{p}\sqrt{k}}{\sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(p_i - \bar{p})^2}} \tag{6.6}$$

where $\bar{p}$ is the average of the p_i differences. This statistic has a Student's t distribution with $k-1$ degrees of freedom. When $k = 5$ as in the compound interpretation experiments described below, the critical values for significance are 2.776445 at the $p < 0.05$ level and

[12] This issue has been recognised as pervasive in NLP tasks (Carpenter, 2008), though I am not aware of any detailed published analysis.

[13] As the SVM algorithm always converges to a unique global optimum it does not display randomness, unlike a neural network which can be very sensitive to the initial setting of weights. However, there is some variability in the cross-validation procedure used here for parameter optimisation.

4.604095 at the $p < 0.01$ level. This test is preferable to alternative tests such as the resampled t-test as each test set is independent; and it also has relatively high power. It is not without its problems, however; Dietterich observes that it can sometimes have an inflated Type I Error rate, rejecting the null hypothesis when it is in fact true.

For comparing two classifiers on a task with a fixed training-testing split, Dietterich recommends McNemar's test. This test involves calculating the statistic

$$m = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{6.7}$$

where n_{01} is the number of items misclassified by the first classifier and classified correctly by the second, and n_{10} is the number of items classified correctly by the second classifier and misclassified by the second. The m statistic has a χ^2 distribution with 1 degree of freedom; the null hypothesis that there is no difference between the classifiers holds with probability $p < 0.05$ if $m > 3.841459$ and with probability $p < 0.01$ if $m > 6.634897$. McNemar's test is a conservative test with low error rate, but it assumes no variance in the training set and does not permit inference about performance with new training data. I use this test to measure significance on the SemEval Task 4 dataset where the training and testing subsets are defined in advance; to count n_{01} and n_{10} the predictions for all relations are pooled to create a 549-element set of predictions. This conflation of the individual subtasks means that McNemar's test does not take the balance of performance across relations into account. An alternative method is to perform paired t-tests on the accuracy and F-score differences for each relation, as in the cross-validation case.[14]

In the following sections I present results on the compound interpretation and SemEval relation classification tasks in terms of accuracy, macro-averaged F-score and the appropriate significance test(s). The significance test is applied to compare distributional kernels with the corresponding standard L_2 kernel; linear kernels are compared with the L_2 linear kernel and RBF kernels are compared with the L_2 RBF kernel, i.e., the Gaussian kernel. No single measure suffices to judge that one classifier is superior to another, but by analysing the three measures across tasks, kernels and feature sets we can come to an informed conclusion.

6.6 Compound noun experiments

Performance on the compound noun dataset was measured using 5-fold cross-validation; for each fold the c and (where appropriate) α parameters were optimised through 10-fold

[14] Demšar (2006) discourages the use of paired t-tests for comparing classifiers across multiple datasets and suggests the Wilcoxon signed-ranks test instead. His argument partly relies on the problems that arise when datasets are not commensurable; this is not a significant concern with the SemEval subtasks. His other concerns about non-normality and the skewing effect of outliers are salient, and with this in mind I have applied the Wilcoxon test to the SemEval results presented in Table 6.6. The patterns of significance and non-significance found are very similar to the t-test results. The only exceptions are that the accuracy improvement of JSD linear kernel with BNC features is found to be significant at the $p < 0.05$ level by the Wilcoxon test, but not by paired t-tests, while the accuracy improvement of the L_1 linear kernel with 5-Gram *and* features is found to be significant by paired t-tests but not by the Wilcoxon test. The Wilcoxon signed-ranks test can also be applied in the cross-validation case; with $k = 5$ as in the compound experiments the lowest attainable value of p is 0.0625. This significance level was reached by all feature-kernel combinations marked significant in Table 6.3 in accuracy and F-score, except for the accuracy improvement of the JSD linear kernel with 5-Gram *all* features.

	BNC		5-Gram (*and*)		5-Gram (*all*)	
LINEAR	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
L_2	57.9	55.8	55.0	52.5	58.1	55.6
L_1	59.2	56.7	58.7**	56.1**	58.3	56.0
JSD	**59.9**	**57.8**	**60.2****	**58.1****	59.9*	57.8**
H	59.8	57.3	59.9**	57.2**	**60.6***	**58.0***
RBF	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
L_2	58.0	56.2	53.5	50.8	56.1	54.0
L_1	58.5	56.2	58.6**	56.5*	58.1	56.0
JSD	**59.8**	**57.9**	**61.0****	**58.8****	59.5*	56.9*
H	55.9	53.0	58.8**	55.7*	**60.6****	**58.3****

Table 6.3: Results for compound interpretation. */** indicate significant improvement over the corresponding L_2 kernel at the 0.05/0.01 level with paired t-tests.

cross-validation on the training set. Table 6.3 gives results for eight kernels and three feature sets. The kernels are the L_2, L_1, Jensen-Shannon and Hellinger (H) linear and RBF kernels described in Section 6.2. The feature sets are the BNC conjunction features (Section 6.4.1), 5-Gram *and* co-occurrences and a 280,000-feature set consisting of the features for all 5-Gram joining terms and verb co-occurrences (Section 6.4.2). The entire optimisation, training and testing procedure for the linear kernels took between about 20 minutes for the sparsest feature set (BNC) and 45 minutes for the least sparse (5-Gram *all*) on a single 2.4 GHz 64-bit processor. For the RBF kernels the procedure took longer, between 45 minutes and three hours, as the additional α parameter had to be optimised, and the kernel matrix was recomputed for each value of α.[15]

Results are presented in Table 6.3. The distributional kernels clearly outperform the L_2 kernels, scoring higher on every kernel-feature combination with just one exception. The JSD and Hellinger kernels perform best and are in general evenly matched, with the exception of the BNC features where the Hellinger RBF kernel does relatively poorly. The best overall classifier is the the JSD RBF kernel computed on the 5-Gram *and* features, which reaches 61.0% accuracy and 58.8% F-score. Significance testing endorses the superiority of the distributional kernels for the two 5-Gram feature sets, but the difference between distributional and L_2 kernels is not significant on the BNC feature set. The reason for this latter finding is that while the distributional kernels do much better on average, the L_2 kernels actually perform slightly better on one or two cross-validation folds. Indeed the L_2-BNC kernel-feature combinations are not confirmed to be significantly worse than any other combination, though their consistently lower performance is suggestive at the least.

Table 6.4 provides a detailed breakdown of performance with the JSD linear kernel across classes and feature sets, including the individual 5-Gram joining terms. The relations classified most successfully are *IN*, *ACTOR*, *INST* and *ABOUT*, with the BNC, 5-Gram *and* and 5-Gram *all* features scoring well above 60% recall and F-score on each of these. The most difficult relation to classify is *HAVE*, on which the 5-Gram *and* classifier achieves the best results of 37.2% recall and 42.4% F-score. The reasons for this difficulty are not

[15]The kernel matrix recomputations were done in a space-saving but time-inefficient manner, with the co-occurrence vector files being read and preprocessed anew for each value of α. This could be done significantly more quickly in cases where speed is required.

	BE		HAVE		IN		ACTOR		INST		ABOUT		OVERALL	
	Recall	F	Recall	F	Recall	F	Recall	F	Recall	F	Recall	F	Acc	F
about	25.7	29.4	29.6	33.1	64.9	56.1	57.6	53.0	44.0	47.1	42.4	43.5	46.0	43.7
and	39.3	44.9	**37.2**	**42.4**	69.2	66.5	69.5	66.4	**66.2**	**63.8**	**68.7**	**64.6**	**60.2**	**58.1**
at	33.0	38.0	29.1	33.4	68.5	63.7	58.1	56.4	55.3	55.0	56.8	52.8	52.3	49.9
by	39.8	43.2	28.1	33.7	62.3	59.5	67.4	62.7	56.4	54.7	53.5	51.8	52.9	51.0
for	35.6	40.1	30.7	37.2	67.5	63.3	69.1	62.5	58.6	56.8	57.6	57.0	55.2	52.8
from	35.1	41.1	24.6	31.1	69.2	63.2	69.1	62.9	58.6	56.1	56.0	54.7	54.3	51.5
in	**46.1**	**50.6**	30.2	35.7	68.2	63.5	66.5	61.9	57.9	57.9	60.5	58.6	56.5	54.7
is	32.5	39.5	27.6	32.9	69.2	64.1	68.2	63.8	60.2	58.9	66.7	61.7	56.3	53.5
like	36.6	40.6	28.6	32.9	62.7	57.8	61.0	58.7	50.4	47.8	44.4	45.5	48.9	47.2
of	40.8	46.2	34.2	38.7	69.2	65.2	70.8	65.5	63.2	62.1	60.5	59.6	58.3	56.2
or	37.7	43.8	27.6	33.5	69.2	65.2	71.6	65.6	62.0	60.4	65.4	61.7	57.7	55.1
to	37.2	43.4	29.6	37.3	69.2	63.7	73.3	66.5	61.7	59.3	60.1	58.3	57.2	54.8
with	36.6	41.3	24.1	31.1	**70.8**	**66.8**	**73.7**	64.3	55.6	54.7	62.6	60.4	56.1	53.1
verb	27.7	34.0	29.1	34.9	61.4	58.0	70.8	61.9	51.5	50.0	58.4	56.6	51.7	49.2
all	41.4	47.3	33.2	39.3	68.2	64.8	**73.7**	68.6	65.0	63.5	67.1	63.2	59.9	57.8
BNC	45.0	49.4	31.6	38.0	69.5	66.3	**73.7**	**68.9**	63.2	61.8	65.8	62.6	59.9	57.8

Table 6.4: Compound interpretation results for individual joining terms and other feature sets with the linear JSD kernel

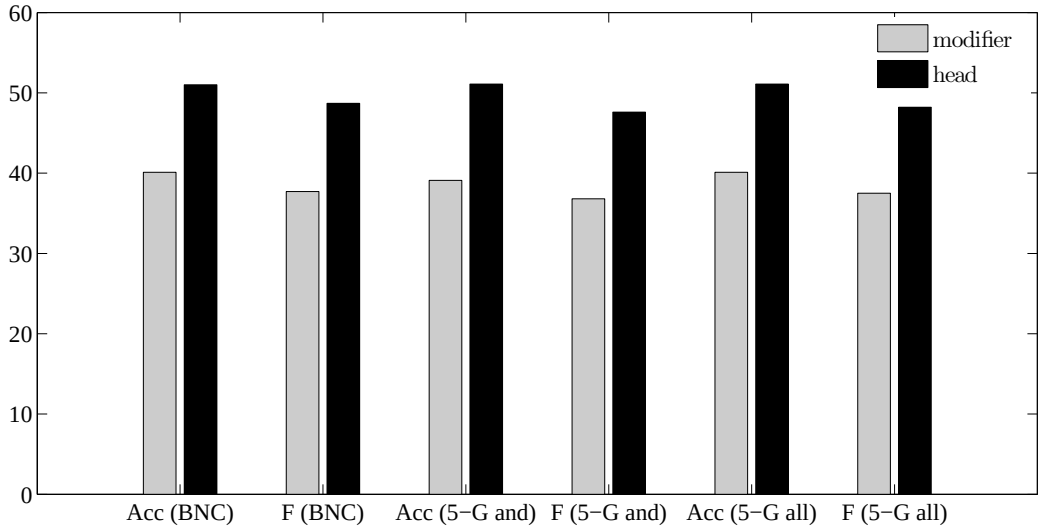


Figure 6.1: Comparison of modifier and head feature performance using the JSD linear kernel and BNC, 5-Gram *and* and 5-Gram *all* feature sets

clear; it may be that the class of *HAVE* relations is semantically heterogeneous or at least that the range of arguments for this relation is heterogeneous.[16] None of the other individual 5-Gram joining terms perform as well as *and*, though some do better on certain relations. The expanded 5-Gram *all* feature set seems truly beneficial only on the *BE* and *ACTOR* relations, where a number of other joining terms give superior performance compared to *and*, and *all* reflects this.

While there is some variation in the strengths and weaknesses of the various feature sets, it is not clear how they can be optimally combined to create improved classifiers. I have experimented with summing combinations of the best-performing kernels, which generally gives a slight boost in performance. The best result was obtained by summing the JSD RBF kernel computed on the BNC features with the Hellinger linear kernel computed on the 5-Gram *and* features, which gave 62.0% accuracy and 60.4% F-score. Though this improvement over the previous best results is small, the difference in F-score is found to be significant when compared to each of the results in Table 6.3, using both paired t-tests and the Wilcoxon signed-ranks test.

There has been some debate among researchers in psycholinguistics regarding the relative importance of modifier and head items in compound comprehension. For example, Gagné and Shoben's (1997) CARIN (*Competition Among Relations In Nominals*) model affords a privileged role to modifiers in determining the range of possible interpretations, and Gagné (2002) finds that meanings can be primed by compounds with semantically similar modifiers but not by similar heads. However, other authors have challenged these findings, including Devereux and Costello (2005), Estes and Jones (2006) and Raffray et al. (2007). While not proposing that human methods of interpretation are directly comparable to machine methods, I think it is interesting to test how informative heads or modifiers are for classifiers when taken separately. Figure 6.1 illustrates results using the JSD linear kernel with just head co-occurrence information or just modifier co-occurrence information. For each feature set, the performance of the head-only classifier is about 10 points above

[16]Interestingly, the *HAVE* relation had the lowest agreement figure in my annotation experiment (Table 4.2). It did not have the lowest one-against-all Kappa score, but it was only 0.002 away.

	Modifier-only		Head-only	
Relation	Accuracy	F-Score	Accuracy	F-Score
BE	**50.8****	**49.4****	22.0	26.5
HAVE	10.1	13.1	**35.7***	**39.7***
IN	**56.5**	50.9	53.9	**53.2**
ACTOR	39.4	38.4	**65.3****	**60.5****
INST	47.0	44.9	**54.1**	**52.4**
ABOUT	28.4	29.9	**65.4****	**60.1****
Overall	40.1	37.7	**51.0****	**48.7****

Table 6.5: Compound interpretation results with the JSD linear kernel with BNC features using modifier-only and head-only co-occurrence information. */** indicate significant positive differences at the 0.05/0.01 level, estimated by paired t-tests.

	BNC			5-Gram (*and*)			5-Gram (*all*)		
LINEAR	Accuracy	F-Score		Accuracy	F-Score		Accuracy	F-Score	
L_2	67.6	57.1		65.4	63.3		69.8	65.8	
L_1	69.0	63.6		67.9	64.0		70.1	65.6	
JSD	**71.4**	**68.8**	†	69.6	65.8	†	70.9	66.8	
H	66.7	61.5		**70.8**	**66.1**	††	**71.2**	**67.9**	
RBF	Accuracy	F-Score		Accuracy	F-Score		Accuracy	F-Score	
L_2	66.8	60.7		65.6	62.9		69.0	65.0	
L_1	67.9	63.0		68.1*	64.1		69.4	65.0	
JSD	**69.9**	**66.7****		**70.7**	**67.5**	††	**72.1**	**68.6**	††
H	65.6	60.5		68.5	66.0		69.9	65.4	

Table 6.6: Results for SemEval Task 4. */** and †/†† indicate significant improvement over the corresponding L_2 kernel at the 0.05 and 0.01 level with paired t-tests and McNemar's test respectively.

the modifier-only classifier in accuracy and F-score; accuracy with head-only information breaks 50%, surprisingly close to the performance achieved by the full combined model. Table 6.5 presents the performance of the JSD linear kernel with BNC features using modifier-only and head-only co-occurrences. The head-only classifier outperforms the modifier-only classifier on all relations except *BE*, where the former performs extremely poorly. On the other hand, modifier information is very weak at recognising instances of *HAVE*, *ACTOR* and *ABOUT*, which seem to be predominantly signalled by the head constituent – for example, compounds headed by *book*, *story* and *film* are very likely to encode a topic relation. The same patterns were observed with all other kernels and feature sets, suggesting that knowledge about compound heads is more informative for compound interpretation, at least when classifying with distributional information.

6.7 SemEval Task 4 experiments

The same eight kernels and three feature sets used in Section 6.6 were applied to the SemEval Task 4 data. Due to the relatively small training set sizes (140 examples for each relation), leave-one-out cross-validation was used to optimise the c and α parameters.

Optimisation, training and testing are very quick, taking 1–3 minutes in total for the linear kernels and 5–37 minutes for the RBF kernels. Results are presented in Table 6.6.

The distributional kernels outperform the corresponding L_2 kernels on almost every kernel-feature combinations, only once scoring lower in both accuracy and F-score (Hellinger RBF kernel with BNC features). The most consistently strong results are obtained with the Jensen-Shannon kernels, whose superiority attains statistical significance with McNemar's test in four out of six cases. Few kernel-feature combinations are found to be significant by paired t-tests; this is because the L_2 kernels tend to do better on one or sometimes two relations. However, a number of kernels come close to significance: the JSD RBF kernel with BNC features ($p = 0.056$) and with 5-Gram *and* features ($p = 0.092$) for accuracy, and the JSD RBF kernel with 5-Gram *all* features for F-score ($p = 0.062$).

The highest accuracy is achieved by the JSD RBF kernel with the large 5-Gram *all* feature set (accuracy = 72.1%, F-score = 68.6%). The highest F-score is achieved by the JSD linear kernel with BNC conjunction features (accuracy = 71.4%, F-score = 68.8%). Both of these kernel-feature combinations surpass the best WordNet-free entry in the SemEval competition by a considerable margin (Nakov and Hearst 2007; accuracy = 67.0%, F-score = 65.1%), and score higher than all but three of the entries that did use WordNet. They also attain slightly better accuracy and slightly lower F-score than the best reported WordNet-free result for this dataset (Davidov and Rappoport 2008; accuracy = 70.1%, F-score = 70.6%). The L_1 and Hellinger kernels, while also performing very well, are slightly less consistent than the JSD kernels. Most of the kernels achieve their best results with the 5-Gram *all* features, though the improvement over the other much sparser and more efficient feature sets is not always large.

Table 6.7 gives a detailed breakdown of the results for the JSD linear kernel with each feature set and each individual joining term. In general, the most difficult relations to identify are *ORIGIN-ENTITY* and *PART-WHOLE*. This tallies with Girju et al.'s (2007) report that *ORIGIN-ENTITY* and *THEME-TOOL* were most problematic for the SemEval participants, followed by *PART-WHOLE* (my system does relatively well on *THEME-TOOL*). The best five feature sets (*and*, *is*, *of*, *all*, BNC) exceed the baseline for each relation, apart from the tough *alltrue* F-score baselines for *PRODUCER-PRODUCT*, *ORIGIN-ENTITY* and *CONTENT-CONTAINER*. Interestingly, *of* is the best-performing single joining term; however, with all other kernels *and* was superior. There is some diversity among the optimal joining terms for each relation. In some cases, the connection between a joining term and a relation is intuitive, as between *for* and *PRODUCER-PRODUCT* and between *with* and *PART-WHOLE*. Other cases are less clear, as between *at* and *THEME-TOOL*. These observations suggest that while the best feature sets have very good overall performance, it should be possible to do even better by selecting the most appropriate feature set for each relation. If one could automatically select the optimal feature set for each relation from the 16 listed in Table 6.7, performance could be improved as far as 75.2% accuracy and 73.8% F-score. However, trial experiments using cross-validation on the training set to select an appropriate kernel have not given good results; it seems that good training performance is not a guarantee of good test performance here. The small training set sizes may be a contributing factor, as they make overfitting more likely. A further generalisation of this approach would be to select optimal linear combinations of kernels (rather than a single kernel) for each relation, in the framework of *multiple kernel learning* (Sonnenburg et al., 2006).

	Cause		Instrument		Product		Origin		Theme		Part		Content		Overall	
	Acc	F	Acc	F	Acc	F	Acc	F	Acc	F	Acc	F	Acc	F	Acc	F
about	55.0	59.1	52.6	50.7	63.4	75.7	50.6	33.3	71.8	64.3	55.6	36.0	67.6	62.5	59.4	54.5
and	72.5	74.4	66.7	66.7	72.0	80.6	64.2	52.5	71.8	66.7	70.8	53.3	68.9	66.7	69.6	65.8
at	52.5	68.3	64.1	66.7	65.6	76.5	59.3	42.1	**80.3**	**75.9**	52.8	41.4	62.2	56.3	62.3	61.0
by	61.3	64.4	60.3	61.7	65.6	75.8	60.5	40.7	73.2	62.7	63.9	0.0	64.9	63.9	64.1	52.7
for	68.8	69.9	65.4	69.0	**74.2**	**82.4**	66.7	50.9	70.4	61.8	59.7	45.3	**68.9**	**67.6**	67.9	63.8
from	63.8	65.9	64.1	66.7	73.1	81.5	64.2	49.1	76.1	69.1	66.7	50.0	**68.9**	64.6	68.1	63.8
in	71.3	74.2	66.7	69.8	68.8	78.5	58.0	32.0	71.8	64.3	69.4	54.2	**68.9**	**67.6**	67.8	62.9
is	68.8	69.1	66.7	68.3	64.5	74.8	**67.9**	55.2	67.6	63.5	70.8	61.8	64.9	63.9	67.2	65.2
like	55.0	60.9	51.3	50.0	62.4	73.7	59.3	37.7	69.0	56.0	52.8	41.4	60.8	58.0	58.7	53.9
of	**77.5**	**79.5**	**70.5**	**71.6**	69.9	78.8	65.4	56.3	69.0	56.0	73.6	62.7	66.2	61.5	**70.3**	**66.6**
or	61.3	64.4	65.4	64.9	65.6	73.3	64.2	49.1	76.1	67.9	68.1	56.6	62.2	54.8	65.9	61.6
to	66.3	67.5	69.2	72.1	64.5	75.6	61.7	49.2	70.4	65.6	66.7	53.8	59.5	57.1	65.4	63.0
with	65.0	66.7	66.7	69.0	63.4	74.2	60.5	38.5	77.5	74.2	**76.4**	**65.3**	67.6	65.7	67.8	64.8
verb	62.5	66.7	66.7	67.5	68.8	79.4	**67.9**	**61.8**	64.8	57.6	63.9	0.0	66.2	65.8	65.9	56.9
all	**77.5**	79.1	67.9	69.9	69.9	79.1	65.4	50.0	76.1	71.2	73.6	55.8	66.2	62.7	**70.9**	**66.8**
BNC	67.5	69.8	**73.1**	**74.1**	71.0	80.3	60.5	42.9	74.6	71.9	**79.2**	**69.4**	**75.7**	**73.5**	**71.4**	**68.8**
Baseline	51.2	67.8	51.3	65.5	66.7	80.0	55.6	61.5	59.2	58.0	63.9	53.1	51.4	67.9	57.0	64.8

Table 6.7: SemEval Task 4 results for individual joining terms and other feature sets with the linear JSD kernel. The baseline figures are the *majority* accuracy and *alltrue* F-score baselines described in Section 6.3.2.

6.8 Further analysis

6.8.1 Investigating the behaviour of distributional kernels

It is clear from the preceding sections that distributional kernels perform much better than the popular L_2 kernels on the two semantic classification tasks described. It is natural to ask why this is so. One answer might be that just as information theory provides the "correct" notion of information for many purposes, it also provides the "correct" notion of distance between probability distributions. Hein and Bousquet (2005) suggest that the property of invariance to bijective transformations of the event space C is a valuable one for image classification,[17] but it is not clear that this confers an advantage in the present setting. When transformations are performed on the space of co-occurrence types, they are generally not information-conserving, for example lemmatisation or stemming.

A more practical explanation is that the distributional kernels and distances are less sensitive than the (squared) L_2 distance and its derived kernels to the marginal frequencies of co-occurrence types.[18] When a type c has high frequency we expect that it will have higher variance, i.e., the differences $|P(c|w_1) - P(c|w_2)|$ will tend to be greater even if c is not a more important signifier of similarity. These differences contribute quadratically to the L_2 distance and hence also to the associated RBF kernel.[19] It is also easy to see that types c for which $P(c|w_i)$ tends to be large will dominate the value of the linear kernel. In contrast, the differences $|P(c|w_1) - P(c|w_2)|$ are not squared in the L_1 distance formula, and the minimum function in the L_1 linear kernel also dampens the effect of high-variance co-occurrence types. The square root terms in the Hellinger formulae similarly "squashes" the range of differences. The difference terms do not directly appear in the formula for Jensen-Shannon divergence, but we can see that while co-occurrence types with large $P(c|w_1)$ and $P(c|w_2)$ do contribute more to the distance and kernel values, it is the proportional size of the difference that appears in the log term rather than its magnitude. Thus the largest contribution to the JSD kernel value is made by frequent co-occurrence types for which large relative differences in co-occurrence frequency is observed. It is plausible that these types are indeed the most valuable for estimating similarity, as rare co-occurrence types may not give good estimates of relative frequency differences.

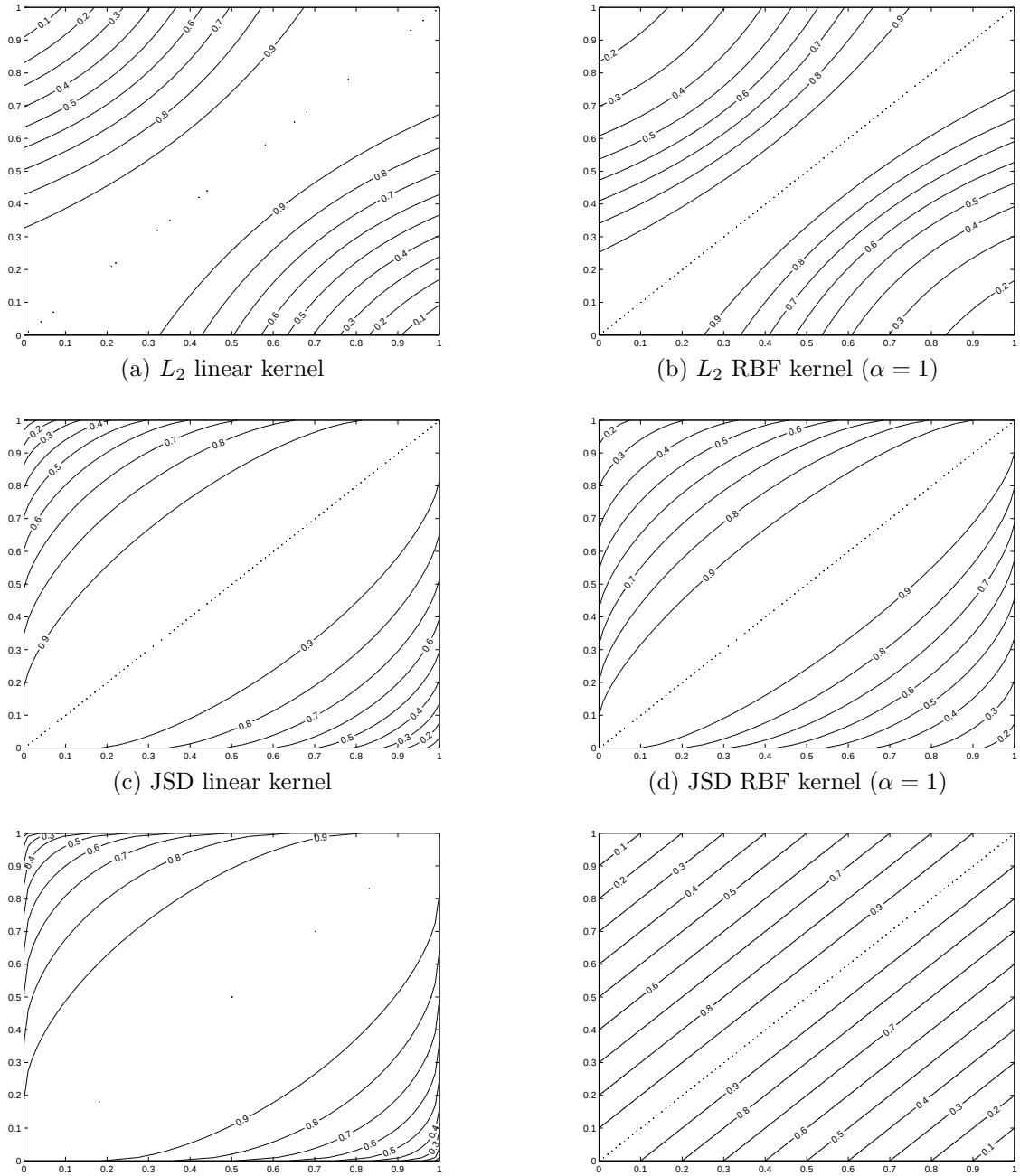
Another perspective on the behaviour of the L_2 and distributional kernels is given by tracking how their value responds to changes in the skewedness of their argument vectors. The contour plots in Figure 6.2 show the kernel values over the space of binomial distributions (i.e., 2-dimensional vectors summing to 1). As each distribution is fully parameterised by the first coordinate value p – the second coordinate is necessarily $1 - p$ – only two dimensions are necessary to plot all relevant information.

Contrasting the L_2 kernel contours with the Jensen-Shannon and Hellinger kernel contours, the clearest difference is that the former are narrowest in the centre of the plot while the latter are narrowest at the extremes. Intuitively, this means that the L_2 kernels are

[17] The same authors make a stronger version of this claim in an earlier paper (Hein et al., 2004).

[18] Chapelle et al. (1999) offer a similar explanation for the efficacy of their distributional kernels for histogram classification.

[19] As noted above, it is sometimes recommended that the range of values for each feature is normalised before applying the Gaussian kernel. One effect of this is to make the feature variances more similar, which smooths the effect of the feature marginals but also "over-smooths" other aspects of variance that may be useful for discrimination.



(a) L_2 linear kernel

(b) L_2 RBF kernel ($\alpha = 1$)

(c) JSD linear kernel

(d) JSD RBF kernel ($\alpha = 1$)

(e) Hellinger linear kernel

(f) L_1 linear kernel

Figure 6.2: Contour plots of kernel functions on binomial distributions with p parameters ranging from 0 to 1. The inputs to the L_2 kernels are L_2-normalised.

more likely to assign high similarity to skewed distributions and the Jensen-Shannon and Hellinger kernels are more likely to assign high similarity to balanced distributions. The RBF kernel contours have the same shape as the corresponding linear kernels; altering the α parameter has the effect of changing the contour steepness. The L_1 kernel does not "bulge" at any point and seems to be invariant to the degree of skew. This behaviour places the L_1 linear kernels at a mid-point between the L_2 and Jensen-Shannon/Hellinger kernels; suggestively, the same could be said of the experimental results with this kernel. It is not certain that these observations directly bear on the performance differences between the L_2 and distributional kernels, nor why a preference for balanced distributions would give better results than a preference for skewed distributions. One possibility is that distributions where most probability mass is placed on a small number of co-occurrence types are very often ones affected by sparsity and therefore not a reliable basis for inference.

6.8.2 Experiments with co-occurrence weighting

As noted in Section 5.3.1.1, co-occurrence vectors are often weighted in order to better identify significant statistical associations in the data. Weighting functions typically compare the observed co-occurrence counts $f(w, c)$ with the values that would be expected from the target marginals $f(w)$ and co-occurrence type marginals $f(c)$ under an independence assumption. Given the discussion of marginal effects in Section 6.8.1, we might therefore expect that this effect of compensating for the effects of marginal values would be beneficial to the linear and Gaussian kernels. A wide range of measures have been proposed for estimating co-occurrence association; Curran (2003) and Weeds and Weir (2005) compare weighting functions in the context of lexical similarity, while Evert (2004) presents a comprehensive analysis of association measures for identifying collocations. I have experimented with a number of popular association measures, including z-score, log-likelihood ratio, mutual information and chi-squared, but the only weighting function that gave positive results is the t-score measure, defined as:

$$g_t(w, c) = \frac{f(w, c) - \frac{1}{N} f(w) f(c)}{\sqrt{f(w, c)}} \tag{6.8}$$

The t-score function can take negative values when the observed co-occurrence count $f(w, c)$ is less than the expected count $\frac{1}{N} f(w) f(c)$, but the Jensen-Shannon and Hellinger kernels are only defined on positive measures. I have observed that simply deleting all negative entries in the weighted co-occurrence vectors is a good solution in this case, and it is indeed highly beneficial to the L_1 and L_2 kernels as well. Weeds and Weir find the t-score is the best-performing weighting function in comparative similarity and pseudo-disambiguation experiments. Curran (2003) reports that his "t-test" measure outperforms other measures on a semantic similarity task, but this is actually the measure called "z-score" by Evert (2004) and Weeds and Weir (2005). This other measure did not perform as well on my compound interpretation and SemEval Task 4 experiments, possibly because of its unreliability when observed frequencies are low and the assumption of normality can cause errors (Evert, 2004).

Results for the t-score reweighted vectors on the compound interpretation task and SemEval Task 4 are given in Tables 6.8 and 6.9 respectively. In the case of the L_2 kernels, the

	BNC		5-Gram (*and*)		5-Gram (*all*)	
LINEAR	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
L_2	60.2 (+2.2)*	58.0 (+2.2)*	58.6 (+3.6)*	56.0 (+3.5)*	**60.6** (+2.5)	**58.6** (+3.0)
L_1	58.6 (−0.6)	56.2 (−0.5)	59.2 (+0.5)	57.0 (+0.9)	59.1 (+0.8)	56.7 (+0.7)
JSD	**60.5** (+0.6)	**58.6** (+0.8)	**59.1** (−1.1)	**57.4** (−0.7)	59.7 (−0.2)	57.2 (−0.6)
H	59.9 (+0.1)	57.8 (+0.5)	59.4 (−0.5)	56.9 (−0.3)	60.0 (−0.6)	57.2 (−0.8)
RBF	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
L_2	**60.6** (+2.6)**	**58.6** (+1.4)*	**60.4** (+6.9)**	**58.6** (+8.6)**	59.5 (+3.4)	57.1 (+2.9)
L_1	59.2 (+0.7)	57.1 (+0.9)	59.8 (+1.2)	57.7 (+1.2)	58.7 (+0.6)	56.0 (0.0)
JSD	59.4 (−0.4)	56.7 (−1.2)	60.3 (−0.7)	58.1 (−0.7)	**60.0** (+0.5)	**57.3** (+0.4)
H	55.1 (−0.8)	52.0 (−1.0)	56.1 (−2.7)	53.7 (−2.0)	59.2 (−1.4)	56.2 (−2.1)*

Table 6.8: Results for compound interpretation with t-score weighting. */** denote results that are significantly different at the 0.05/0.01 level from the corresponding unweighted results, using paired t-tests.

effect is dramatic. For almost every feature set their performance improves by a considerable margin, and for the BNC and 5-Gram *and* features these improvements consistently attain statistical significance. Furthermore, the performance of the reweighted L_2 kernels is comparable to that of the best distributional kernels, exceeding 60% accuracy four times on the compound dataset and 70% accuracy five times on the SemEval dataset. Reweighting has an inconsistent effect on the distributional kernels, improving performance for some kernel-feature combinations and harming performance for others. The 5-Gram *and* features responded best to the t-score reweighting, yielding some increase in performance with all kernels. The best-scoring kernels with these reweighted features were the Jensen-Shannon and Hellinger linear kernels, with 72.3% accuracy, 69.7% F-score and 72.9% accuracy, 69.3% F-score respectively.

These results strongly support the hypothesis outlined in Section 6.8.1 that the distributional kernels' robustness to marginal frequency effects plays a major role in their superiority over L_2 kernels. By compensating for marginal effects, t-score reweighting confers the same robustness on the L_2 kernels, allowing them to bridge the performance gap. The fact that the same procedure does not usually have a significant effect on the distributional kernels also indicates that these kernels already possess the benefits brought by reweighting as part of their default behaviour. Nevertheless, there are a number of reasons why a user would opt to use distributional kernels instead of L_2 kernels. Firstly, a single-step process is more parsimonious and simpler than a process requiring multiple steps. More decisively, there are many circumstances where obtaining information about marginal frequencies is impractical or impossible (Section 5.3.1.2), for example

	BNC		5-Gram (*and*)		5-Gram (*all*)	
LINEAR	Accuracy	F -Score	Accuracy	F-Score	Accuracy	F-Score
L_2	**71.0** (+3.4)	**69.7** (+12.6) †	70.1 (+4.7)	66.1 (+2.8) †	**71.6** (+1.8)	**67.6** (+1.8)
L_1	68.9 (−0.1)	64.4 (+0.8)	70.1 (+2.2)	65.6 (+1.6) †	68.7 (−1.4)	63.2 (−2.4)
JSD	69.0 (−2.4)*	65.3 (−3.5) †	72.3 (+2.9)	**69.7** (+3.9) †	**71.6** (+0.7)	**67.6** (+0.8)
H	70.3 (+3.6)	68.1 (+6.6) †	**72.9** (+2.1)*	69.3 (+3.2) †	70.9 (+0.3)	66.4 (+1.5)
RBF	Accuracy	F -Score	Accuracy	F-Score	Accuracy	F-Score
L_2	**70.1** (+3.3)*	**67.1** (+6.4) *†	71.0 (+5.4)	67.0 (+4.1) ††	69.6 (+0.6)	64.4 (-0.6)
L_1	69.0 (+1.1)	65.2 (+2.2)	70.3 (+2.2)	65.9 (+1.8)	69.2 (−0.2)	64.8 (−0.2)
JSD	69.0 (+0.9)	62.8 (−3.9)	**72.3** (+1.6)	67.7 (+0.2)	**71.6** (−0.5)	**68.2** (−0.4)
H	63.4 (−2.2)	59.2 (−1.2)	70.9 (+2.4)	**68.3** (+2.3)	68.9 (−1.0)	63.7 (−1.7)

Table 6.9: Results for SemEval Task 4 with t-score weighting. */** and †/†† indicate significant differences at the 0.05 and 0.01 level compared to the corresponding unweighted results, using paired t-tests and McNemar's test respectively.

when the user does not have access to the original corpus or when deep processing is required to identify co-occurrences and the corpus is too large to process in its entirety. Finally, the best results on the two tasks described above were obtained with distributional kernels, indicating that they maintain a narrow superiority even when reweighting is feasible. On the other hand, the L_2 linear kernel might be preferred for tasks where speed is paramount and the dataset is very large, as specialised SVM implementations are available for this kernel that scale linearly in the number of data items and the number of features (Joachims, 2006).

6.9 Conclusion

In this chapter I have shown that an approach to semantic relation classification based on lexical similarity can give very good results. I have also shown that kernels on co-occurrence distributions offer a means of kernelising popular measures of lexical similarity and are very effective when used for classification with support vector machines. These distributional kernels, which have not previously been applied to semantic processing tasks, have been shown to outperform the L_2 linear and Gaussian kernels that are standardly used. On the SemEval Task 4 dataset, they achieve state-of-the-art performance, scoring higher than the best WordNet-free entry in that competition.

It appears that one of the principal factors contributing to the superiority of distributional kernels is that they are influenced to a lesser degree than the L_2 kernels by variation in

the marginal frequencies of co-occurrence types. This variation does not generally have predictive value, and can be seen as a kind of noise obscuring the underlying true co-occurrence distributions. Theoretical evidence for this analysis comes from the formulae used to compute the kernels, while empirical evidence comes from studying the effects of statistical association measures that compensate for the effects of marginal frequency.

Chapter 7

Learning with strings and sets

7.1 Introduction

In Chapter 5 I described two approaches to modelling semantic similarity between noun pairs – one based on lexical similarity, the other on relational similarity. Lexical similarity was the subject of Chapter 6. In this chapter I develop a model of relational similarity based on kernel methods that compare strings (Section 7.2) and sets of strings (Section 7.3). These methods implement what were called *token-level* and *type-level* relational similarity in Section 5.3.2, and are respectively appropriate for application to SemEval Task 4 and the compound interpretation task. While the relational models described here do not attain the same level of performance as the lexical models of the previous chapter, I demonstrate their value by showing that systems combining the two similarity types can be more effective than either model alone.

7.2 String kernels for token-level relational similarity

7.2.1 Kernels on strings

Many structured objects can be viewed as being constructed out of simpler substructures: we can decompose strings into substrings, graphs into subgraphs, trees into subtrees and so on. Haussler (1999) uses this insight to define the class of *convolution kernels* on compositional structures. The particular family of kernels that is of relevance here, called *R-convolution kernels* by Haussler, is defined in terms of a relation R that holds between an indexed set of substructures $\mathbf{x}_i = (x_{i1} \in \mathcal{X}_1, \ldots, x_{iD} \in \mathcal{X}_D)$ and a composite structure $x_i \in \mathcal{X}$ if a decomposition of x_i gives $x_{i1}, .., x_{iD}$, or equivalently, when $x_{i1}, \ldots, x_{iD}$ are the parts of x_i. The inverse function R^{-1} maps a structure $x_i \in \mathcal{X}$ onto the set $\{\mathbf{x}_i | R(\mathbf{x}_i, x_i)\}$ of all valid decompositions of x_i. Assuming that kernels $k_1, \ldots, k_D$ are defined on each of the part sets $\mathcal{X}_1, \ldots, \mathcal{X}_D$, a kernel k_R can be defined on the set $\mathcal{X}$ as follows:

$$k_R(x_i, x_j) = \sum_{\mathbf{x}_i \in R^{-1}(x_i)} \sum_{\mathbf{x}_j \in R^{-1}(x_j)} \prod_{d=1}^{D} k_d(x_{id}, x_{jd}) \tag{7.1}$$

This very general definition accommodates kernels on a wide range of objects, including RBF and ANOVA kernels on vectors (Haussler, 1999), trees (Collins and Duffy, 2001;

Zelenko et al., 2003; Moschitti, 2006) and other classes of graphs (Suzuki et al., 2003; Vishwanathan et al., 2006), sequences of images (Cao et al., 2006), as well as a variety of different kernels on strings. The last of these, the *string kernels*, will be the basis of the methods explored in this chapter.

String kernels have become popular in both natural language processing and bioinformatics, two domains where data often takes the form of symbolic sequences. The most commonly used kernels compute the similarity of two strings by counting their shared subsequences.[1] These kernels correspond to inner products in feature spaces where each dimension indexes a single subsequence and an input string is mapped onto a vector of subsequence counts. The space of subsequences used for this mapping can be restricted in various ways, by only counting subsequences of a fixed length or subsequences that are contiguous in the input string, or by limiting the number and size of permitted gaps in discontinuous subsequences. Applying such restrictions, especially those on discontinuity, allows the use of extremely fast algorithms for kernel computation; the resulting loss of richness in the embedding model is tolerable in applications where discriminative patterns are expected to be localised, e.g., in protein or DNA sequence comparisons (Vishwanathan and Smola, 2002; Leslie and Kuang, 2004; Sonnenburg et al., 2007). For natural language processing, however, the ability to capture long-distance relationships between words is important and most applications in this field have used kernels that count all contiguous and non-contiguous subsequences in a string, typically with a weighting parameter that penalises subsequences with large gaps. The initial publications on these *gap-weighted subsequence kernels* considered subsequences of characters (Lodhi et al., 2002), but subsequent work has adopted a more intuitive word-based representation (Cancedda et al., 2003). Notable applications of string kernels to semantic processing tasks include work on word sense disambiguation (Gliozzo et al., 2005), relation extraction (Bunescu and Mooney, 2005b) and semantic parsing (Kate and Mooney, 2006).

A *string* is defined as a finite sequence $\mathbf{s} = (s_1, \ldots, s_l)$ of symbols belonging to an alphabet Σ. Σ^l is the set of all strings of length l, and Σ^* is set of all strings or the *language*. A subsequence $\mathbf{u}$ of $\mathbf{s}$ is defined by a sequence of indices $\mathbf{i} = (i_1, \ldots, i_{|\mathbf{u}|})$ such that $1 \leq i_1 < \cdots < i_{|\mathbf{u}|} \leq |\mathbf{s}|$, where $|\mathbf{s}|$ is the length of $\mathbf{s}$. $len(\mathbf{i}) = i_{|\mathbf{u}|} - i_1 + 1$ is the length of the subsequence in $\mathbf{s}$. For example, if $\mathbf{s}$ is the string *cut the bread with the knife* and $\mathbf{u}$ is the subsequence (*cut, with*) indexed by $\mathbf{i}$ then $len(\mathbf{i}) = 4$. λ is a decay parameter between 0 and 1. The smaller the value of λ, the more the contribution of a gappy subsequence is reduced. The gap-weighted kernel value for subsequences of length l of strings $\mathbf{s}$ and $\mathbf{t}$ is given by

$$k_{String_l}(\mathbf{s}, \mathbf{t}) = \sum_{u \in \Sigma^k} \sum_{\mathbf{i},\mathbf{j}:\mathbf{s}[\mathbf{i}]=\mathbf{u}=\mathbf{t}[\mathbf{j}]} \lambda^{len(\mathbf{i})+len(\mathbf{j})} \tag{7.2}$$

This kernel induces a feature embedding $\phi_{String_l} : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|^l}$ that maps a string $\mathbf{s}$ onto a vector of positive "counts" that are not generally integers unless $\lambda = 0$ or $\lambda = 1$. Directly computing the function in (7.2) would be intractable, as the sum is over all $|\Sigma|^n$ possible subsequences of length n; however, Lodhi et al. (2002) present an efficient dynamic programming algorithm that can evaluate the kernel in $O(l|s||t|)$ time and does not require that the feature vector $\phi_{String_l}(\mathbf{s})$ of all subsequence counts be represented explicitly.

[1] A separate class of string kernels is based on the comparison of probabilistic sequence models such as hidden Markov models (Jaakola and Haussler, 1998; Watkins, 2000; Jebara et al., 2004). As far as I am aware, these methods have not gained significant traction in the NLP community.

The λ decay parameter is set to 0.5 throughout the experiments reported in this chapter, following the recommendation of Cancedda et al. (2003). I have observed that while optimising λ for a particular dataset can lead to small performance improvements, the optimal value is often difficult to find and $\lambda = 0.5$ consistently gives near-optimal results.

String kernels, and convolution kernels in general, should also be normalised so that larger substructures are not assigned higher similarity values simply because of their size. For example, the string $\mathbf{s}$ = *the dog runs and the dog jumps* has a higher count of length-two subsequences shared with $\mathbf{t}$ = *the dog runs* than $\mathbf{t}$ does with itself and thus $k_{String_2}(\mathbf{s},\mathbf{t}) > k_{String_2}(\mathbf{s},\mathbf{s})$; however, $\mathbf{s}$ also contains many subsequences that are *not* shared with $\mathbf{t}$. The standard method for normalising a kernel k is through the operation

$$\bar{k}(\mathbf{s},\mathbf{t}) = \frac{k(\mathbf{s},\mathbf{t})}{\sqrt{k(\mathbf{s},\mathbf{s})}\sqrt{k(\mathbf{t},\mathbf{t})}} \tag{7.3}$$

This is equivalent to normalising the substructure count vectors $\phi(\mathbf{s})$ and $\phi(\mathbf{t})$ to have unit L_2 norm. As a result, the normalised kernel $\bar{k}(\mathbf{s},\mathbf{t})$ has a maximal value of 1, which is taken when $\mathbf{s}$ and $\mathbf{t}$ are identical.

7.2.2 Distributional string kernels

In Section 6.2 I showed how distributional kernels provide alternative feature space inner products to the dot product provided by the standard L_2 kernels. Distributional kernels can also be applied to structures, by treating the feature embedding ϕ as a function that maps structures to unnormalised distributions over substructures. By normalising the feature vector $\phi(\mathbf{s})$ for a structure $\mathbf{s}$ to have unit L_1 norm, we obtain a vector $\mathbf{p}_\mathbf{s} = (P(s_1|\mathbf{s}),\ldots,P(s_d|\mathbf{s}))$ parameterising a multinomial probability distribution of dimension d. Distributional kernels can then be applied to these probability vectors in the same way as to co-occurrence probability vectors. Alternatively, L_2 normalisation can be applied if an L_2-based kernel is to be used for the string comparison.[2] This vector normalisation step is sufficient to ensure the string kernel matrix is normalised: the L_2, L_1 and Hellinger linear kernels, and all RBF kernels, will take values between 0 and 1, while the JSD linear kernel will take values between 0 and 2. The JSD linear kernel can be scaled by 0.5 to bring it into the same range as the other kernels, for example when combining kernels. In this chapter I consider distributional kernels on strings only, but in principle the approach sketched here is general.

In order to compute the kernel value for a pair of strings, their subsequence probability vectors must be represented in memory. Although these vectors typically have very high dimension, they are also very sparse and can be stored efficiently in sorted arrays or hash tables. Storing the feature vector $\phi_{String_l}(\mathbf{s})$ entails representing up to $\binom{|\mathbf{s}|}{l}$ subsequence counts for each string. This is not problematic for the small SemEval dataset but can lead to high memory loads when the dataset is very large. In Section 7.3.1 I describe how time efficiency can be traded off for space efficiency in the special context of set learning, leading to acceptable resource requirements even for hundreds of thousands of strings.

Computing the feature mapping ϕ_{String_l}, which must be performed just once for each string, takes $O(|\mathbf{s}|^2\binom{|\mathbf{s}|}{l})$ time as each subsequence must be explicitly counted; for $l \ll s$

[2] In fact, applying the L_2 linear kernel to L_2-normalised subsequence count vectors gives the standard string kernel (7.2) after normalisation (7.3).

this is close to $O(|\mathbf{s}|^{l+2})$. The exponential dependence on subsequence length l may look worrying, but in practice the values of l used will be very small; in my experiments I did not find any advantage in using values greater than $l = 3$. Once the feature mapping has been performed, the distributional kernel can then be computed for each string pair $(\mathbf{s}, \mathbf{t})$ in $O(\binom{\max(|\mathbf{s}|,|\mathbf{t}|)}{l})$ time. This is not generally more efficient in the token-level relational scenario where each string must be compared to each other string, but as I show in Section 7.3 it allows the development of very efficient kernels on sets of strings as the second kernel calculation step must only be computed once per set.

This application of distributional kernels to convolutional mappings seems to be novel, although there is some relevant prior work. Jebara et al. (2004) use the Hellinger linear kernel to compare hidden Markov models trained on gene sequence data. Rieck and Laskov (2008) have recently described a general framework for comparing representations in $\mathcal{F}$ induced by a string kernel. Their framework accommodates my method but they do not consider a probabilistic interpretation or the use of kernels on distributions, which are key to facilitating the extension to set kernels introduced in Section 7.3.2. Several authors have suggested applying distributional similarity measures to sentences and phrases for tasks such as question answering (Lin and Pantel, 2001; Weeds et al., 2005). Distributional kernels on strings and trees should provide a flexible implementation of these suggestions that is compatible with SVM classification and does not require manual feature engineering.

7.2.3 Application to SemEval Task 4

7.2.3.1 Method

In Chapter 6 I presented a model for classifying the SemEval Task 4 dataset using only information about the lexical similarity of relation arguments. However, it seems intuitive that ignoring the context in which the arguments appear entails discarding valuable information. For example, the sentence *The <e1>patient</e1> had crushed a pencil with this <e2>toe</e2> about 30 years previously* is labelled as a positive instance of the *INSTRUMENT-AGENCY* relation in the dataset; this is not because toes are inherently tools, but rather because the sentence describes a toe being used in a tool-like manner. An approach to this task based on comparing context sentences – in the terminology of Section 5.3.2, a *token-level relational similarity* approach – can therefore complement the efficacy of lexical approaches. String kernels offer a means of implementing this kind of relational similarity.

As a preprocessing step, the SemEval Task 4 context sentences were tagged and lemmatised with RASP (Briscoe et al., 2006). In order to avoid overfitting on particular argument words and to focus on purely relational information, the candidate relation arguments $e1$ and $e2$ were replaced with placeholder tokens tagged as nouns. All non-alphanumeric characters were removed and punctuation tokens were discarded. As in the previous chapter, classification was performed with LIBSVM (Chang and Lin, 2001) and the SVM c parameter was optimised through leave-one-out cross-validation.

Length	Accuracy	F-Score
1	58.3	46.2
2	**63.2**	**59.0**
3	61.2	45.2
Σ_{12}	61.2	53.4
Σ_{23}	**64.1**	**59.7**
Σ_{123}	61.6	54.8

Table 7.1: Results for string kernels on SemEval Task 4

	BNC		5-Gram (*and*)		5-Gram(*all*)	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	68.7	68.3	67.6	66.6	68.9	67.5
2	70.5	**68.1**	**71.4**	**69.7**	70.9	**69.6**
3	**70.9**	66.9	71.0	67.4	**71.4**	68.7
Σ_{12}	69.4	68.5	70.3	68.1	70.3	68.7
Σ_{23}	**71.4**	**68.9**	71.0	69.1	**72.3**	**70.4**
Σ_{123}	70.3	69.2	71.0	68.2	70.9	68.6
No String	**71.4**	68.8	69.6	65.8	70.9	66.8

Table 7.2: Results for string and JSD linear co-occurrence kernel combination on SemEval Task 4

7.2.3.2 Results

Results using the gap-weighted string kernel algorithm of Lodhi et al. (2002) are presented in Table 7.1. Only subsequence lengths l up to three are considered, as longer subsequences are extremely sparse and give very poor classification results. The $l = 2$ subsequence kernel gives the best results for an individual kernel (63.2% accuracy, 59.0% F-score), while the summed combination of the kernels with $l = 2$ and $l = 3$ (Σ_{23} in the table) gives a slight improvement to 64.1% accuracy and 59.7% F-score. These figures do not compare well with the results achieved using the lexical similarity model (Table 6.6), nor do they reach the *alltrue* F-score baseline of 64.8%. It seems that the information contained in the context sentences is insufficient or overly sparse for successful classification with current state-of-the-art methods. Nevertheless, combining the relational information provided by string kernels and the lexical information provided by the kernels of Chapter 6 can lead to an improvement in performance over both individual methods, as shown in Table 7.2. This combination method seems to benefit F-score in particular, with the most beneficial string kernels (k_{String_2} and $k_{String_{\Sigma_{23}}}$) providing a boost of 0.8–4.0% over the JSD linear kernel on all co-occurrence feature sets. However, none of these increases are statistically significant.

SemEval Task 4 results for L_1, Jensen-Shannon and Hellinger distributional string kernels are presented in Table 7.3. I only consider the linear versions of these kernels to avoid the complication of optimising the α width parameter for RBF kernels. As can be seen from the table, these kernels do not perform better than the standard string kernels. One reason for this may be the artificiality of fitting a multinomial distribution to a single observation, which is essentially what I am doing here. Combinations of distributional kernels on strings and co-occurrence vectors can nevertheless be quite effective, as demonstrated in

	L_1		*JSD*		*H*	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	59.7	44.6	60.1	47.5	60.3	44.4
2	61.6	57.5	61.0	54.8	63.4	**57.7**
3	**64.1**	55.7	61.7	45.8	61.2	48.2
Σ_{12}	59.9	48.0	60.8	48.6	61.2	46.9
Σ_{23}	61.4	56.1	62.1	**58.8**	62.5	57.7
Σ_{123}	60.1	47.8	62.8	58.2	63.6	56.7

Table 7.3: Results for distributional string kernels on SemEval Task 4

	BNC		5-Gram (*and*)		5-Gram(*all*)	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	69.8	68.9	69.6	68.5	71.8	**70.2**
2	**71.2**	**69.2**	71.0	**68.8**	**71.9**	69.1
3	71.0	68.1	**71.2**	68.6	71.2	68.5
Σ_{12}	70.1	68.7	71.6	69.3	71.4	69.7
Σ_{23}	**71.6**	**69.3**	70.5	67.5	72.3	69.2
Σ_{123}	71.0	69.0	**72.7**	**70.7*** †	**72.7**	**70.6**
No String	71.4	68.8	69.6	65.8	70.9	66.8

Table 7.4: Results for JSD linear string and JSD linear co-occurrence kernel combination on SemEval Task 4. * and † indicate significance at the $p < 0.5$ level with paired t-tests and McNemar's test respectively.

Tables 7.4 and 7.5. The Jensen-Shannon string kernels perform particularly well in this context. Almost all the combinations listed in Table 7.4 outperform the corresponding co-occurrence kernel, the best-performing being the sum of the JSD linear kernel computed on the 5-Gram *and* co-occurrence features with the combined Σ_{123} JSD linear string kernel which achieves 72.7% accuracy and 70.7% F-score. This is the best result yet reported on the SemEval Task 4 dataset for a WordNet-free method, and the improvement over the performance of the co-occurrence kernel alone is found to be statistically significant by paired t-tests as well as McNemar's test. The Hellinger string kernels perform slightly less well in combination (Table 7.5), while still having a positive effect in most cases and achieving a statistically significant (with paired t-tests) improvement using the combined Σ_{123} kernel with 5-Gram *and* features.

Table 7.6 shows the effect of combining lexical and relational information on classifying individual relations. I have used the JSD linear kernel with 5-Gram *and* features and the JSD linear string kernel as representative of the lexical and relational kernels respectively; the same patterns of behaviour are also observed with other kernel combinations. Although the relational string kernel does not match the performance of the lexical kernel on any relation, the combined kernel achieves a notable improvement on the *ORIGIN-ENTITY*, *THEME-TOOL*, *PART-WHOLE* and *CONTENT-CONTAINER* relations while maintaining the performance of the lexical kernel on the others. Accuracy on *PRODUCT-PRODUCER* does drop slightly.

	BNC		5-Gram (*and*)		5-Gram(*all*)	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	67.5	66.2	67.4	65.5	70.7	69.4
2	69.9	66.7	**71.2**	**69.1**	70.1	67.5
3	**71.0**	**68.3**	71.0	67.6	**72.3**	**69.0**
Σ_{12}	69.6	68.0	70.9	68.9	71.0	69.5
Σ_{23}	**70.9**	**68.7**	71.2	68.4	70.7	67.6
Σ_{123}	69.9	68.4	**71.4**	**69.3***	**71.6**	**69.6**
No String	**71.4**	68.8	69.6	65.8	70.9	66.8

Table 7.5: Results for Hellinger linear string and JSD linear kernel combination on SemEval Task 4. * indicates significance at the $p < 0.5$ level with paired t-tests.

	Co-occurrence only		String only		Co-occurrence + String	
Relation	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
CAUSE	**72.5**	**74.4**	60.0	66.7	**72.5**	**74.4**
INSTRUMENT	66.7	66.7	60.3	64.4	**67.9**	**68.4**
PRODUCT	**72.0**	**80.6**	66.7	80.0	69.9	79.7
ORIGIN	64.2	52.5	56.8	36.4	**66.7**	**57.1**
THEME	71.8	66.7	67.6	43.9	**78.9**	**74.6**
PART	70.8	53.3	70.8	58.8	**79.2**	**66.7**
CONTENT	68.9	66.7	58.1	57.5	**75.7**	**74.3**
OVERALL	69.6	65.8	62.8	58.2	**72.7**	**70.7**

Table 7.6: Results on SemEval Task 4 with co-occurrence information (JSD linear kernel with 5-Gram *and* features), context string information (JSD linear string kernel, length Σ_{123}), and the combination of both information sources (summed kernel)

7.3 Set kernels for type-level relational similarity

7.3.1 Kernels on sets

Given a basic kernel k_0 on objects of a certain kind, we can derive a kernel on sets of those objects. Informally speaking, the kernel similarity between two sets will be a function of the basic kernel similarities between their members. Here I describe some previously proposed kernels on sets, as well as novel kernels based on a multinomial distributional model. In the next section I apply a range of set kernels to the compound interpretation task, by associating each compound constituent pair with a set of context strings extracted from a corpus. This application implements the type-level relational similarity model of Section 5.3.2.

One natural way of defining a kernel over sets is to take the average of the pairwise basic kernel values between members of the two sets A and B. Let k_0 be a kernel on a set $\mathcal{X}$, and let $A, B \subseteq \mathcal{X}$ be sets of cardinality $|A|$ and $|B|$ respectively. The *averaged kernel* is defined as

$$k_{ave}(A, B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} k_0(a, b) \tag{7.4}$$

This kernel was used by Gärtner et al. (2002) in the context of multiple instance learning.

It is relatively efficient, with the computation being dominated by the $|A|\times|B|$ basic kernel calculations. Lyu (2005b) suggests modifying the basic definition in (7.4) by raising the summed term raised to a power $k_0(a,b)^p$. However, this modification did not improve on the performance of the basic averaged kernel (7.4) in my experiments and will not be discussed further.

An alternative perspective views each set as corresponding to a probability distribution, and the members of the set as samples from that distribution. In this way a kernel on distributions can be cast as a kernel on sets. Indeed, Smola et al. (2007) show that the averaged kernel (7.4) is also a kernel on distributions, being the inner product between the means of distributions in the feature space $\mathcal{F}$, each of which is estimated as the sample mean of the feature space representations of the members of each set.

Cuturi et al. (2005) propose a kernel on measures and distributions in $\mathcal{F}$ which can be used to formulate a kernel on sets. Cuturi et al. suggest that the similarity of two measures μ and μ' corresponds to the dispersion of their sum $\frac{\mu+\mu'}{2}$ – if the measures are similar then their sum will be more concentrated than if they are dissimilar. Taking entropy as a suitable measure of dispersion, Cuturi et al. derive a kernel that is the same as the previously discussed Jensen-Shannon RBF kernel of Hein and Bousquet (2005). The second dispersion measure they consider is *inverse generalised variance (IGV)*, the determinant of a measure's covariance matrix. The IGV kernel is defined as:

$$k_{igv}(A,B) = \frac{1}{\det(\frac{1}{\eta}\tilde{K}_0\Delta + I)} \tag{7.5}$$

where $\tilde{K}_0$ is the centred $(|A|+|B|)\times(|A|+|B|)$ basic kernel matrix between all members of A and B, Δ is a diagonal matrix of the same dimension with entries $\Delta_{ii} = \frac{1}{|A|+|B|}$ and $\eta > 0$ is a regularisation parameter that smooths the eigenspectrum of $\tilde{K}_0\Delta$ and also guarantees its invertibility. The determinant of $\tilde{K}_0\Delta$ is the same as that of the covariance matrix Σ of the measure $\frac{\mu+\mu'}{2}$ as these matrices have the same eigenspectrum; this allows the calculation of the inverse generalised variance without representing the elements of $\mathcal{F}$ so long as their representation is not necessary to compute the basic kernel k_0. The time requirements of the IGV kernel are dominated by the determinant calculation, which has $O((|A|+|B|)^3)$ time complexity.[3]

Kondor and Jebara (2003) also adopt a probabilistic framework for set classification, proposing a kernel based on Bhattacharyya's affinity (Bhattacharyya, 1943):

$$k_{bhatt}(p,p') = \int \sqrt{p(x)}\sqrt{p'(x)}\,\mathrm{d}x$$

This kernel is identical to the Hellinger linear kernel of Hein and Bousquet (2005). When that kernel was used in Chapter 6, multinomial distributions were used to model co-occurrence probabilities. However, the set kernel used by Kondor and Jebara is based on a different probability model, fitting multivariate normal distributions to the feature space mappings of the sets A and B. This yields a closed-form expression for the kernel that can be computed without explicitly representing the elements of $\mathcal{F}$:

$$k_{bhatt}(A,B) = \det(\Sigma_{\mathcal{W}})^{-\frac{1}{4}}\det(\Sigma'_{\mathcal{W}})^{-\frac{1}{4}}\det(\Sigma^{\dagger}_{\mathcal{W}})^{-\frac{1}{2}}e^{-(\mu^T\Sigma^{-1}\mu)/4}e^{-(\mu'^T\Sigma'^{-1}\mu')/4}e^{(\mu^{\dagger T}\Sigma^{\dagger -1}\mu^{\dagger})/2} \tag{7.6}$$

[3]The matrix operations involved in the calculation of the IGV and Bhattacharyya kernels were carried out using the JLAPACK library of linear algebra algorithms (Doolin et al., 1999).

where $\mathcal{W}$ is the subspace spanned by the kernel mappings of the elements of A and B, μ and $\Sigma_{\mathcal{W}}$ are the (estimated) mean and regularised covariance matrix of the distribution fit to the embedding of A in that space, μ' and $\Sigma'_{\mathcal{W}}$ are the mean and covariance of the distribution fit to the embedding of B, $\mu^\dagger = \frac{1}{2}\Sigma_{\mathcal{W}}^{-1}\mu + \frac{1}{2}\Sigma_{\mathcal{W}}'^{-1}\mu'$ and $\Sigma^\dagger = (\frac{1}{2}\Sigma_{\mathcal{W}}^{-1} + \frac{1}{2}\Sigma_{\mathcal{W}}'^{-1})^{-1}$. To calculate these means and covariances we require an orthogonal basis $\mathbf{e}_1, \ldots, \mathbf{e}_{\dim(\mathcal{W})}$ for $\mathcal{W}$; this can be found through the eigendecomposition of the $(|A|+|B|) \times (|A|+|B|)$ basic kernel matrix K_0.[4] Letting $\lambda_1, \ldots, \lambda_{\dim(\mathcal{W})}$ be the non-zero eigenvalues of K_0 and $\mathbf{v}_1, \ldots, \mathbf{v}_{\dim(\mathcal{W})}$ the corresponding eigenvalues, we obtain an orthonormal basis by setting $\mathbf{e}_i = \mathbf{v}_i/\sqrt{\lambda_i}$. E is the matrix containing the basis vectors $\mathbf{e}_i$ as columns. Then:

$$\mu = E^T K_0 M, \qquad M_{i,j} = \begin{cases} \frac{1}{|A|} & \text{if } i = j \wedge x_i \in A \\ 0 & \text{otherwise} \end{cases}$$

$$\Sigma_{\mathcal{W}} = E^T K_0 S K_0 E + \eta I, \qquad S_{i,j} = \begin{cases} \frac{1}{|A|} + \frac{1}{|A|^2} & \text{if } i = j \wedge x_i \in A \\ \frac{1}{|A|^2} & \text{if } i \neq j \wedge x_i \in A \\ 0 & \text{otherwise} \end{cases}, \qquad \eta > 0$$

with equivalent definitions for μ' and $\Sigma'_{\mathcal{W}}$. I is the $(|A|+|B|) \times (|A|+|B|)$ identity matrix. η is again a regularisation parameter. While computing (7.6) has low memory requirements it can be quite slow when the sets being compared are large. Using the method described in Kondor (2005) the length of the calculation is dominated by a number of costly matrix multiplications and LU decompositions (for finding inverses and determinants), each of which has $O((|A|+|B|)^3)$ time complexity. A potentially faster method based on dimensionality reduction of the distributions in $\mathcal{F}$ is outlined by Kondor and Jebara (2003), though I have not implemented this.

7.3.2 Multinomial set kernels

The distributional model used by Kondor and Jebara (2003) has the advantage that it allows an implicit computation of the Bhattacharyya kernel in a feature space $\mathcal{F}$ using only information about inner products in that space, i.e., the combined kernel matrix for each pair of sets. One disadvantage of this approach, and of the other set kernels that have been proposed in the literature, is that $|A||B|$ basic kernel evaluations must be computed for each pair of sets A, B. When the basic kernels have a significant computational cost, as most convolution kernels do, this can make working with large sets impractical. A second potential disadvantage is that the restriction to multivariate normal distributions may not be appropriate to the data. Normal distributions assign non-zero probabilities to negative feature values, but convolution kernel embeddings only map structures onto non-negative measures.

These concerns suggest investigating alternative probability models, such as multinomial distributions. In Section 7.2 I described how strings can be mapped onto vectors of subsequence counts. A multinomial distribution over a set of strings can then be estimated by taking the sum of the count vectors of the set members, i.e.:

$$\phi_{Set_l}(A) = \sum_{\mathbf{s} \in A} \phi_{String_l}(\mathbf{s}) \tag{7.7}$$

[4] K_0 may not have full rank, for example when A or B contain duplicate members or when $A \cap B \neq \emptyset$. In this case the eigenvalues of K_0 will not all be non-zero. This is not a serious problem, but it can be avoided altogether by adding a small amount of mass (e.g., 0.0001) to the diagonal of K_0.

where l is the subsequence length associated with the string embedding ϕ_{String_l}. The embedded vector $\phi_{Set_l}(A)$ should then be normalised to have unit L_1 or L_2 norm, as appropriate. Any suitable inner product can be applied to these vectors, e.g., L_2 linear or RBF kernels or the distributional kernels of Section 6.2. In fact, when the L_2 linear kernel is used, the resulting set kernel is equivalent to the averaged set kernel (7.4) without the averaging term $\frac{1}{|A||B|}$. Instead of requiring $|A||B|$ basic kernel evaluations for each pair of sets, multinomial set kernels only require the embedding $\phi_{Set_l}(A)$ once for each set and then a single vector inner product for each pair of sets. This is generally far more efficient than previously proposed set kernels. The significant drawback is that representing the feature vector for each set demands a large amount of memory; each vector potentially contains up to $|A|\binom{|\mathbf{s}_{max}|}{l}$ entries, where $\mathbf{s}_{max}$ is the longest string in A. In practice, however, the vector length will be lower due to subsequences occurring more than once and many strings being shorter than $\mathbf{s}_{max}$.

One way to reduce the memory load is to reduce the lengths of the strings used, either by retaining just the part of each string expected to be informative or by discarding all strings longer than an acceptable maximum. Bunescu and Mooney (2005b) use three separate kernels to compare preceding, middle and subsequent contexts and use a linear combination of the individual kernels to compute string similarity. Another method, which does not reduce the representative power of the model, is to trade off time efficiency for space efficiency by computing the set kernel matrix in a blockwise fashion. To do this, the input data is divided into blocks of roughly equal size – the size that is relevant here is the sum of the cardinalities of the sets in a given block. For each set, all members should be in the same block. In order to compute the set kernel matrix, one block at a time is selected as the active block. The feature mapping ϕ_{Set_l} is computed for each set in that block and the kernel values for pairs of sets in the block are calculated. Then, each other block which has not yet been paired with the active block is selected in turn; the sets in this block are embedded with ϕ_{Set_l} and compared to the sets in the active block. If m is the sum of the cardinalities of all sets in the data and b is the block size, then the number of blocks is $o = \lceil \frac{m}{b} \rceil$. The total number of string embeddings ϕ_{String_l} that must be calculated in the course of the set kernel matrix calculation is approximately $\frac{bo}{2}(o+1)$, due to the symmetry of the kernel matrix.[5] Larger block sizes b therefore allow faster computation, but they require more memory. An acceptable balance can usually be found, as shown in Section 7.3.4.

7.3.3 Related work

The above discussion is not an exhaustive account of all set kernels proposed in the machine learning literature. However, many of the kernels I have not discussed are unsuitable for use in the experiments that follow in Section 7.3.4, either because they can only be applied to vectorial data or because their computational costs scale poorly with increasing set sizes. Wolf and Shashua (2003) describe a kernel based on the principal angles between the subspaces spanned by the feature space embeddings of two sets. This kernel is only guaranteed to be positive semi-definite when the sets compared have equal cardinality; furthermore, Cuturi et al. (2005) observe that this approach is only suitable for sets of small cardinalities, as the kernel matrices produced become highly diagonal otherwise. The set kernel of Shashua and Hazan (2004) is restricted to comparing sets of

[5]The term "approximately" appears because in practice the blocks will vary in size.

vectors, and is not easily extended to work in the feature space of a convolution kernel. The same is true of Grauman and Darrell's (2007) *pyramid match kernel*. Cuturi (2007) shows that the permanent of the basic kernel matrix is also a useful positive semi-definite kernel on sets, but its computation is prohibitively expensive for sets of medium to large cardinality. Lyu (2005a) describes a kernel between mixtures of Gaussian distributions. These mixtures can be located in a kernel feature space $\mathcal{F}$, in which case a kernelised version of the EM algorithm must be applied before the kernel is computed.

Most applications of set kernels have been in visual classification experiments; for example, representing images as unordered sets of local feature vectors provides a greater robustness to transformations than using a single global feature vector. In NLP there is relatively little use of set representations, and hence little need for set classification methods. Some notable exceptions were described in Section 5.3.2.3's overview of type-level relational methods. Bunescu and Mooney (2007) tackle a relation extraction task by considering the set of contexts in which the members of a candidate relation argument pair co-occur. While this gives a set representation for each pair, Bunescu and Mooney do not directly compare sets. Rather, they apply a string kernel SVM classifier to each context string individually and classify a candidate pair as positive if any of the context strings are labelled positive. This may be reasonable under the assumptions of the relation extraction task they study, but it is not appropriate for compound interpretation or any non-binary classification task. Rosario and Hearst (2005) take a similar approach to classifying protein-protein interactions, representing each document-protein-protein triple in their dataset as a set of strings and classifying each sentence individually. The label assigned to each triple is decided by a majority vote of the sentence labels. This approach could be applied to the compound noun task, though it would not be more efficient than the averaged set kernel, still effectively requiring $|A||B|$ basic kernel calculations for each pair of compounds.

As far as I am aware, none of the set kernels described above have previously been used for natural language tasks. However, there is a close connection between the multinomial probability model I have proposed and the pervasive *bag of words* (or *bag of n-grams*) representation. It is common in NLP to represent documents, co-occurrence contexts or any other collection of strings as an unordered bag of unigram or n-gram observations. This implicitly estimates an unnormalised multinomial measure for the collection of strings, and indeed some authors have used this insight to apply distributional kernels for document classification (Jebara et al., 2004; Hein and Bousquet, 2005; Lafferty and Lebanon, 2005). Distributional kernels based on a gap-weighted feature embedding extend these models by using bags of discontinuous n-grams and downweighting gappy subsequences; when only subsequences of length 1 are used, this is equivalent to a standard bag of words model.

Turney's (2008) PairClass algorithm is also related to the multinomial model and can in fact be viewed as a special case where a more restrictive embedding function is used.[6] Further differences are that PairClass uses the Gaussian kernel to compare feature vectors (though in principle any vector kernel could be used), and that in PairClass patterns falling

[6]PairClass considers only length-n contexts of the form [$0-1$ word] N_1/N_2 [$0-3$ words] N_2/N_1 [$0-1$ word] and performs a feature embedding by mapping each such context onto 2^n-1 length-n "patterns" obtained by substituting up to $n-1$ context words with wildcards. These patterns are not discontinuous in the way that those produced by ϕ_{String_l} are; the PairClass pattern ** knife cuts * cheese* obtained from the context *the knife cuts the cheese* would not match the context *the knife cuts the blue cheese* as each wildcard can match only one word.

below a certain frequency threshold in the dataset are discarded, whereas my relational methods consider all subsequences regardless of their frequency. In future work I intend to investigate in detail the effects of these various differences.

7.3.4 Application to compound noun interpretation

7.3.4.1 Method

In Section 5.3.2 I described the concept of type-level relational similarity between noun pairs: two pairs are assumed to be similar if the contexts in which the members of one pair co-occur are similar to the contexts in which the members of the other pair co-occur. Relational similarity can be useful for identifying the semantic relations in compound nouns, as shown by Turney (2006) for example. If we assume that the contexts where the constituents of a compound appear together provide evidence for the compound's relational semantics, we can compare compounds by comparing the corresponding context sets. One method of implementing this comparison is to use kernels on sets of strings.[7]

I performed classification experiments with set kernels on the same dataset of 1,443 noun-noun compounds that was used in Chapter 6. Context strings for each compound in the dataset were extracted from two corpora: the written component of the British National Corpus (Burnard, 1995) and the English Gigaword Corpus, 2nd Edition (Graff et al., 2005). As in previous chapters, the BNC was tagged and lemmatised with RASP (Briscoe et al., 2006). However, the Gigaword Corpus, containing approximately 2.3 billion words of newswire text and taking up 5.3 Gigabytes in compressed form, is impractical for preprocessing. To generate a more tractable corpus, all Gigaword paragraphs containing both constituents of at least one compound in the dataset were extracted. Extraction was performed at the paragraph level as the corpus is not annotated for sentence boundaries. A dictionary of plural forms and American English variants was used to expand the coverage of the corpus trawl; this dictionary was created manually, but it could also have been created automatically as in Section 6.4.2. This extraction procedure yielded a much-reduced subcorpus of 187 million words. Using RASP the subcorpus was split into sentences, tagged and lemmatised.

Combining the BNC and the Gigaword subcorpus resulted in a corpus of 277 million words. For each compound in the dataset, the set of sentences in the combined corpus containing both constituents of the compound was identified. As the Gigaword Corpus contains many duplicate and near-duplicate articles, duplicate sentences were discarded; this was observed to improve set classification performance, presumably by preventing frequent context strings from dominating the similarity estimates. The compound modifier and head were replaced with placeholder tokens `M:n` and `H:n` in each sentence to ensure that the classifier would learn from relational information only and not from lexical information about the constituents. Punctuation and tokens containing non-alphanumeric characters were removed. Finally, all tokens more than five words to the left of the leftmost constituent or more than five words to the right of the rightmost constituent were

[7]Given that the compound dataset was annotated in context (Chapter 4) and that context is known to affect compound meaning, it might seem useful to implement a token-level relational model using information about the BNC sentences in which the data items were found. However, experiments with the method I applied to SemEval Task 4 in Section 7.2.3 were unsuccessful, failing even to attain chance-level performance.

discarded; this has the effect of speeding up the set kernel computations and should also focus the classifier on the most informative parts of the context sentences. Examples of the context strings extracted for the modifier-head pair (*history,book*) are

```
the:a 1957:m pulitizer:n prize-winning:j H:n describe:v event:n
in:i american:j M:n when:c elect:v official:n take:v principle:v

this:d H:n will:v appeal:v to:i lover:n of:i military:j M:n
but:c its:a characterisation:n often:r seem:v

you:p will:v enter:v the:a H:n of:i M:n as:c patriot:n
museveni:n say:v

in:i the:a past:n many:d H:n have:v be:v publish:v on:i the:a
M:n of:i mongolia:n but:c the:a new:j

subject:n no:a surprise:n be:v a:a M:n of:i the:a american:j
comic:j H:n something:p about:i which:d he:p be:v

he:p read:v constantly:r usually:r H:n about:i american:j M:n
or:c biography:n
```

There was significant variation in the number of context strings extracted for each compound: 49 compounds were associated with 10,000 or more sentences, while 161 were associated with 10 or fewer and no sentences were found for 33 constituent pairs. The largest context sets were predominantly associated with political or economic topics (e.g., *government official*, *oil price*, *government policy*), reflecting the journalistic sources of the Gigaword sentences. The total number of context strings was 2,266,943.

I applied three previously proposed set kernels – Gärtner et al.'s (2002) averaged kernel (k_{ave}), Kondor and Jebara's (2003) Bhattacharyya kernel (k_{bhatt}) and Cuturi et al.'s (2005) IGV kernel (k_{igv}) – and three multinomial set kernels based on the L_2 (k_{L_2}), Jensen-Shannon (k_{jsd}) and Hellinger (k_{hell}) linear inner products to the compound noun dataset.[8] For each kernel I tested values in the range $\{1, 2, 3\}$ for the subsequence length parameter l, as well as summed kernels for all combinations of values in this range. Subsequence lengths greater than 3 were not observed to contribute to the overall performance.

As in Section 7.2.3, the λ parameter for the gap-weighted substring embedding was set to 0.5 throughout. For the IGV and Bhattacharyya kernels the covariance smoothing parameter η must also be specified; Cuturi et al. use a value of 0.01, while Kondor and Jebara use both 0.1 and 0.01 in different experiments. The optimal parameter value depends on the nature of the feature space induced by the basic kernel and also on how a particular dataset is mapped to this space. While Cuturi et al. and Kondor and Jebara use a Gaussian basic kernel, I am using string kernels here. The computational cost of the IGV and Bhattacharyya kernels precludes optimising η by cross-validation; I have found that $\eta = 0.5$ gives reasonable results and use this value in all experiments. For those kernels requiring computation of the basic kernel matrix for each set pair (k_{ave},

[8] I also performed experiments with L_1-based multinomial kernels, which gave very similar results to those obtained with k_{L_2}, k_{jsd} and k_{hell}. I omit these results to streamline my presentation.

$q = 50$	$l = 1$	$l = 2$	$l = 3$
k_{ave}	06h 40m 03s	12h 33m 32s	16h 46m 25s
k_{igv}	09h 49m 13s	15h 19m 34s	19h 44m 00s
k_{bhatt}	14h 06m 05s	1d 00h 42m 58s	1d 04h 27m 38s
k_{jsd}	01m 57s	41m 24s	07h 27m 12s
$q = 250$	$l = 1$	$l = 2$	$l = 3$
k_{ave}	4d 03h 50m 23s	7d 03h 13m 57s	9d 20h 58m 30s
k_{igv}	16d 14h 15m 48s	18d 05h 32m 15s	23d 01h 06m 44s
k_{bhatt}	35d 06h 11m 13s	44d 13h 54m 41s	45d 20h 30m 02s
k_{jsd}	08m 36s	03h 06m 50s	1d 23h 25m 40s
$q = 1,000$	$l = 1$	$l = 2$	$l = 3$
k_{ave}	29d 11h 43m 26s	52d 00h 22m 15s	71d 07h 58m 4s
k_{jsd}	27m 05s	10h 23m 39s	*8Gb* 10d 01h 57m 09s
			20Gb 03d 23h 13m 16s

Table 7.7: Execution times for set kernel computations

k_{bhatt}, k_{igv}), a two-step normalisation process was applied: each basic kernel evaluation was normalised using the formula in (7.3), then each set kernel evaluation was normalised in the same way. As in Section 7.2, the multinomial kernels require only a single L_1 or L_2 normalisation of the feature vector for each set. To investigate the trade-off between performance and efficiency I ran experiments with context sets of maximal cardinality 50, 250 and 1,000. These sets were randomly sampled for each compound; for compounds associated with fewer strings than the maximal cardinality, all associated strings were used. It was not possible to apply the IGV and Bhattacharyya kernels to the 1,000-string sets for computational reasons (illustrated by the timing data in Table 7.7).

7.3.4.2 Comparison of time requirements

All experiments were run on near-identical machines with 2.4 Ghz 64-bit processors. The set kernel matrix computation is trivial to parallelise, as each cell is independent. Spreading the computational load across multiple processors is a simple way to reduce the real time cost of the procedure. While the kernels k_{ave}, k_{igv} and k_{bhatt} have no significant memory requirements, the k_{L_2}, k_{jsd} and k_{hell} multinomial kernels are computed faster when more memory is available as larger block sizes b can be used. The multinomial kernels were computed on machines with 8 Gigabytes of memory; to illustrate the speed-ups that can be obtained by increasing the available memory even further, I also ran the most resource-intensive calculation (k_{jsd} with $q = 1,000$, $l = 3$) on a machine with 20 Gigabytes. Execution times for k_{ave}, k_{igv}, k_{bhatt} and k_{jsd} are shown in Table 7.7. I omit timing results for k_{L_2} and k_{hell} as they have the same scaling properties as k_{jsd}.

When comparing the observed execution times, it is useful to keep in mind the theoretical complexities of the relevant algorithms. Assuming a slight idealisation of the dataset in which all strings have length s and all sets have cardinality q,[9] and letting n be the number of sets in the data, l be the subsequence length of the basic kernel embedding and b be the block size for k_{jsd}, the time requirements are as follows:

[9]This idealisation clarifies the exposition but does not materially affect the analysis.

$$\begin{array}{ll} k_{ave} & O(n^2q^2ls^2) \\ k_{igv} & O(q^3) \\ k_{bhatt} & O(q^3) \\ k_{jsd} & O\left(\left(\frac{q^2s^2}{b}+q\right)n^2\frac{s^l}{l!}\right) \end{array}$$

n is constant in Table 7.7 as there are 1,443 context sets in the data. The maximum cardinality q varies from 50 to 1,000, while the subsequence length l varies from 1 to 3. The results show that the time taken by k_{ave} scales close to linearly as l increases, and superlinearly as q increases; the theoretical quadratic dependence on q is not observed because for many constituent pairs there are not enough context strings available to keep adding as q grows large. The computation of k_{igv} and k_{bhatt} is dominated by the $O(q^3)$ matrix operations, which render those kernels unusable for sets of large cardinality. The observed scaling of k_{igv} suggests that applying it to the $q = 1,000$ dataset would take a number of years (k_{bhatt} would take even longer). k_{jsd} is by far the fastest kernel on all parameter settings considered here, often requiring just minutes or hours to achieve what takes the other kernels days or weeks. When the block size b is greater than q the time taken by k_{jsd} scales linearly with q; this condition is in fact necessarily satisfied, as the blockwise computation algorithm does not permit any set to occupy more than one block. Most of my experiments used $b = 6,000$. In practice, keeping b as large as possible has a clear effect on efficiency. The execution time scales dramatically as l increases, principally because of the essentially s^l cost of each feature embedding but also because larger subsequence lengths entail larger feature spaces and hence smaller block sizes. For the $q = 1,000$ dataset with $l = 3$, b was reduced to 3,000 on an 8Gb machine; when 20Gb was available, a block size of 15,000 could be used.

On the basis of this analysis, it seems clear that the JSD and other multinomial set kernels have a definite advantage over the averaged set kernel when the basic kernel embedding ϕ_{String_l} uses a small subsequence length l, tackling even very large datasets with relative ease (the $q = 1,000$ dataset contains 598,342 strings). For large l and small set cardinality q the averaged kernel may be preferable, but in practice large values of l tend not to be used with string kernels.[10] It is unlikely that any of these kernels would be usable for higher values of both l and q.

7.3.4.3 Results

Cross-validation performance figures are given for k_{ave}, k_{igv} and k_{bhatt} in Table 7.8, and for k_{L_2}, k_{jsd} and k_{hell} in Table 7.9. Results with maximal set cardinality $q = 50$ are averaged across five random samples to reduce sampling variation. None of the set kernels approach the performance attained by the lexical similarity methods of Chapter 6. The best-performing set kernels are the Jensen-Shannon multinomial kernel with maximal cardinality $q = 1,000$ and subsequence length $l = 2$ (52.7% accuracy, 50.3% F-score) and the combination of the averaged set kernels with $q = 1,000$ and $l = 2$ and $l = 3$ (52.6% accuracy, 51.1% F-score). I was unable to obtain competitive results with the IGV kernel, which performs very poorly throughout. The other five kernels all attain roughly equal performance, with no kernel outperforming the rest on all subsequence lengths and set

[10] For example, Cancedda et al. (2003) obtain their best results with $l = 2$ and Bunescu and Mooney (2005b) use values up to $l = 4$.

$q = 50$	k_{ave}		k_{igv}		k_{bhatt}	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	41.4	38.7	26.9	23.7	42.2	34.7
2	45.8	43.9	30.6	26.0	46.0	43.7
3	44.9	43.0	28.9	24.2	43.6	41.1
Σ_{12}	46.4	44.7	27.9	25.1	45.8	43.7
Σ_{23}	45.7	43.7	30.2	25.5	45.2	42.8
Σ_{123}	**46.9**	**45.1**	28.2	25.0	46.3	44.1
$q = 250$	k_{ave}		k_{igv}		k_{bhatt}	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	44.5	40.8	24.7	20.1	46.4	43.5
2	48.9	47.3	27.0	20.2	49.1	46.8
3	48.6	46.8	26.8	22.5	49.0	46.4
Σ_{12}	48.8	47.0	26.5	21.2	48.6	46.8
Σ_{23}	**51.1**	**49.5**	26.7	20.1	50.0	47.7
Σ_{123}	50.4	48.4	26.2	20.4	49.4	47.3
$q = 1,000$	k_{ave}					
Length	Accuracy	F-Score				
1	45.0	42.2				
2	50.0	48.3				
3	50.9	49.5				
Σ_{12}	50.6	48.8				
Σ_{23}	**52.6**	**51.1**				
Σ_{123}	52.5	50.9				

Table 7.8: Results for set kernels on the compound interpretation task

sizes. As might be expected, more data helps: as the maximal cardinality q increases, so do the performance figures.

Combining token-level relational similarity and lexical similarity was observed to improve performance on SemEval Task 4 in Section 7.2.3. It seems intuitive that combining type-level relational similarity and lexical similarity could also help compound noun interpretation. Tables 7.10–7.12 show classification results for combinations of set kernels and Jensen-Shannon linear kernels trained on constituent co-occurrence features (Section 6.6). The combined kernels almost always outperform the corresponding individual kernels, except for combinations containing string kernels with subsequence length $l = 1$ which have varying effects. The greatest and most consistent improvements are achieved with the multinomial kernels. The best overall result is attained by the combination of the JSD linear kernel computed on BNC co-occurrence features with the summed JSD set kernel with length $l = \Sigma_{123}$: 62.7% accuracy, 61.2% F-score. This result is found to be a statistically significant improvement over all co-occurrence-only kernels (cf., Table 6.3) with the exception of the JSD RBF kernel with 5-Gram *and* features, in which case the improvement is close to significance ($p = 0.077$).

Table 7.13 examines the effect of kernel combination on individual compound relations, taking the JSD linear kernel with BNC conjunction features and the JSD multinomial string kernel as an example pair. Kernel combination improves classification on five of six relations, and the only decrease (on the *ACTOR*) relation is relatively small.

$q = 50$	k_{L_2}		k_{jsd}		k_{hell}	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	41.5	38.8	42.6	39.6	42.1	39.0
2	45.6	43.9	47.3	44.5	47.1	44.4
3	43.8	42.0	44.8	41.7	45.0	41.9
Σ_{12}	45.8	44.1	46.4	43.9	46.1	43.8
Σ_{23}	45.5	43.7	47.2	44.2	47.0	44.1
Σ_{123}	46.3	44.6	**47.6**	**45.0**	47.0	44.2
$q = 250$	k_{L_2}		k_{jsd}		k_{hell}	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	45.2	41.9	46.9	44.3	47.4	45.3
2	49.3	47.7	50.9	48.2	49.1	46.2
3	48.8	47.1	49.3	46.1	48.2	44.6
Σ_{12}	50.1	48.4	**51.6**	49.1	50.3	47.9
Σ_{23}	49.3	47.3	50.8	47.8	50.0	46.8
Σ_{123}	51.5	**49.8**	50.9	48.4	49.4	46.9
$q = 1,000$	k_{L_2}		k_{jsd}		k_{hell}	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	44.8	41.5	49.3	47.1	49.3	47.0
2	49.2	47.4	**52.7**	**50.3**	51.4	48.8
3	50.2	48.8	50.5	47.5	48.2	45.1
Σ_{12}	49.8	48.1	52.4	50.2	52.0	49.9
Σ_{23}	50.7	49.1	52.1	49.6	51.1	48.5
Σ_{123}	51.6	50.2	52.1	49.9	51.7	49.3

Table 7.9: Results for multinomial distributional set kernels on the compound interpretation task

	BNC		5-Gram (*and*)		5-Gram (*all*)	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	57.8	55.6	58.1	56.0	59.0	56.5
2	60.2	58.6	60.5	58.7	61.2	59.2
3	61.7	60.1*	**61.7**	**60.1**	**61.8***	59.7*
Σ_{12}	59.7	58.1	59.4	57.4	60.4	58.0
Σ_{23}	61.8	60.2	60.8	59.1	**61.8**	**59.9**
Σ_{123}	**62.1***	**60.4***	60.1	58.3	61.4	59.3
No String	59.9	57.8	60.2	58.1	59.9	57.8

Table 7.10: Results for averaged set kernel and JSD linear co-occurrence kernel combination on the compound interpretation task. * indicates significant improvement at the 0.05 level over the co-occurrence kernel alone, estimated by paired t-tests.

	BNC		5-Gram (*and*)		5-Gram (*all*)	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	59.9	58.1	59.9	57.6	61.3	59.3
2	62.2*	60.7**	61.6	59.9	61.7*	59.8**
3	62.2**	60.3**	61.9	60.0	61.5	59.3
Σ_{12}	62.1*	60.6**	61.1	59.4	61.1*	59.0
Σ_{23}	62.4*	60.8*	**62.3**	**60.5**	**62.4****	**60.5****
Σ_{123}	**62.7****	**61.2****	61.8	60.1	61.7**	59.7**
No String	59.9	57.8	60.2	58.1	59.9	57.8

Table 7.11: Results for JSD linear set kernel and JSD linear co-occurrence kernel combination on the compound interpretation task. */** indicate significant improvement at the 0.05/0.01 level over the co-occurrence kernel alone, estimated by paired t-tests.

	BNC		5-Gram (*and*)		5-Gram (*all*)	
Length	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
1	59.9	58.3	60.4	58.2	60.6	58.4
2	62.0*	60.5**	61.6	59.9	61.7	59.4
3	62.0**	60.2**	61.5	59.6	61.1	58.7
Σ_{12}	61.7*	60.2**	61.5	59.8	60.5	57.9
Σ_{23}	62.3*	60.7**	**62.1**	**60.3***	**62.2***	**60.3****
Σ_{123}	**62.6***	**61.1****	62.0	60.2	61.0	58.7
No String	59.9	57.8	60.2	58.1	59.9	57.8

Table 7.12: Results for Hellinger linear set kernel and JSD linear co-occurrence kernel combination on the compound interpretation task. */** indicate significant improvement at the 0.05/0.01 level over the co-occurrence kernel alone, estimated by paired t-tests.

	Co-occurrence only		String only		Co-occurrence + String	
Relation	Accuracy	F-Score	Accuracy	F-Score	Accuracy	F-Score
BE	45.0	49.4	31.9	35.0	**48.7**	**53.0**
HAVE	31.6	38.0	30.2	37.4	**43.2****	**49.4****
IN	69.5	66.3	68.5	59.9	**72.4***	**69.5***
ACTOR	**73.7**	**68.9**	66.9	62.9	72.0	67.9
INST	63.2	61.8	49.6	50.5	**63.9**	**63.3**
ABOUT	65.8	62.6	53.5	53.5	**67.1**	**64.3**
OVERALL	59.9	57.8	52.1	49.9	**62.7****	**61.2****

Table 7.13: Results for compound interpretation with co-occurrence information (JSD linear kernel with BNC features), context string information (JSD linear string kernel, length Σ_{123}), and the combination of both information sources (summed kernel). */** indicate significant improvement at the 0.05/0.01 level over the co-occurrence-only results, estimated by paired t-tests.

7.4 Conclusion

In this Chapter I have demonstrated how string kernels can be used to classify noun pairs on the basis of their relational similarity, i.e., the similarity between the contexts in which the two constituents of each pair occur together. For the SemEval Task 4 dataset the implementation of relational similarity is a token-level one, in that only the context sentences the candidate relation arguments appear in are used for the similarity computation. For the compound noun dataset each constituent pair is associated with a set of sentences extracted from their co-occurrences in a large corpus.

I have described previously proposed kernel methods for comparing strings and sets, and applied them to these two tasks. I have also described novel kernels based on fitting multinomial distributions to the embeddings of string sets in the feature space associated with a convolution kernel. These multinomial kernels do not consistently outperform the previously proposed kernels, but in the context of set classification they afford impressive gains in time efficiency.

The performance achieved with these relational similarity methods does not match that achieved with the lexical similarity methods described in Chapter 6. However, combining relational and lexical similarity through kernel combination brings improvements over either method alone, attaining state-of-the-art results on both compound interpretation and SemEval Task 4.

Chapter 8

Conclusions and future work

8.1 Contributions of the thesis

In this thesis I have dealt with the problem of automatically classifying semantic relations between nouns, with a particular focus on the semantics of compound nouns. This has involved novel approaches to the linguistic annotation of compound noun data, and to statistical learning methods for relation classification. The main contributions of the thesis are summarised in this section.

A new relational annotation scheme for compounds: In Chapter 2, I surveyed a variety of frameworks that have been proposed by theoretical and computational linguistics for representing compound noun semantics. Numerous relation inventories have previously been used in computational research on compound interpretation, but in general measures of inter-annotator agreement, when reported at all, have been quite low and annotation guidelines have not been made publicly available. In view of these factors, I decided to develop a new scheme for annotating the relational semantics of compound nouns. In Chapter 3 I presented a number of general desiderata for semantic annotation schemes, which can be used to guide and evaluate design decisions. I described the process of developing the new scheme and presented it in some detail. The annotation guidelines that accompany the scheme are publicly available, and are included here as Appendix B.

A new annotated compound dataset: I used the annotation scheme introduced in Chapter 3 to annotate a sample of 2,000 noun sequences extracted from the British National Corpus. A sample of 500 items was annotated by a second annotator to estimate the inter-annotator agreement achieved with the new scheme and guidelines. All annotation took account of the compounds' sentential context, which is an important aspect of their semantics but had not been investigated in previous annotation studies. The Kappa measure of agreement was 0.62, which compares very favourably to previously reported results and attests to the importance of rigorously developed guidelines for reproducible annotation. These results are presented in Chapter 4, alongside a detailed analysis of observed patterns of agreement and disagreement.

Lexical and relational similarity paradigms: In Chapter 5 I discussed two approaches to comparing pairs of nouns. The lexical similarity approach is based on comparing each constituent of a pair to the corresponding constituent of another pair; many techniques for estimating the similarity of single words have been applied in the NLP literature. The relational similarity approach is based on comparing the set of contexts

associated with each word pair. This context set can simply contain the sentence in which the word pair was found (what I have called *token-level* similarity), or it can contain a sample of all sentences in a large corpus containing both constituents of the pair (*type-level* similarity). The distinction between lexical and relational similarity has been recognised by other researchers, but the two models have not previously been applied to compound interpretation in an integrated manner. In this thesis I have demonstrated how kernel methods provide a flexible framework for implementing and combining different kinds of similarity models.

Distributional kernels for semantic classification: Standard distributional measures of lexical similarity implicitly use a probabilistic representation of a word's co-occurrence behaviour. In Chapter 6 I described a family of kernels on probability measures and showed that they are closely connected to popular and proven methods for distributional similarity. These distributional kernels, which had not previously been applied to semantic classification, performed very well on datasets for compound noun interpretation and the SemEval 2007 task on identifying semantic relations between nominals. Furthermore, the distributional kernels consistently outperformed the Gaussian and linear kernels standardly used for classification with support vector machines on such tasks. I proposed an analysis of the superiority of distributional kernels in terms of robustness to the marginal frequencies of co-occurrence types, and provided theoretical and empirical evidence supporting this analysis.

Old and new methods for relational similarity: In Chapter 7 I took convolutional string kernels, which compare strings by implicitly mapping strings to vectors of subsequence counts, as a starting point for implementing relational similarity. I showed how distributional inner products can be applied to feature space mappings of strings, implicitly fitting multinomial distributions to these mappings. There was little difference between the resulting multinomial string kernels and standard string kernels on the SemEval Task 4 dataset, but when implementing a type-level relational approach to compound interpretation the multinomial model facilitated the development of very efficient kernels on sets of strings. These multinomial set kernels can be computed many times faster than other set kernels described in the literature, while achieving equal or better classification performance.

Combining lexical and relational similarity: The classification results obtained with the relational similarity methods of Chapter 7 were not as good as the lexical similarity results of Chapter 6. However, integrating the two models through kernel combination led to better performance than either method achieved alone, yielding state-of-the-art results both for compound noun interpretation and SemEval Task 4. Performance on the SemEval dataset was the best yet reported for any system not making use of WordNet or other manually constructed resources (Accuracy = 72.7%, F-score = 70.7%).

8.2 Future work

Other sets of semantic relations: The classification methods I have introduced in this thesis are not restricted to the sets of semantic relations assumed by the compound noun and SemEval datasets. They can also be applied to similar tasks that use different relation inventories, for example domain-specific sets of relations in biomedical texts (Rosario and Hearst, 2001; Rosario and Hearst, 2004).

Further applications of distributional kernels: The results presented in this thesis show that distributional kernels are highly effective tools for capturing lexical semantic information in a classification framework. It seems likely that this effectiveness will transfer to other semantic classification tasks. Support vector machines have been widely adopted in computational semantics for tasks ranging from word sense disambiguation (Gliozzo et al., 2005) to semantic role labelling (Pradhan et al., 2004), and distributional kernels could be applied to many of these. In Ó Séaghdha and Copestake (2008) I show that distributional kernels attain state-of-the-art performance on a task of classifying verbs into semantic categories using subcategorisation frame information. However, the standard feature sets for semantic role labelling and many other tasks are collections of heterogeneous features that do not correspond to probability distributions. So long as the features are restricted to positive values, distributional kernels can still be used; it will be interesting (and informative) to see whether they prove as successful in this setting.

Extension of the relational model to analogical reasoning tasks: Turney (2006), having developed a model of analogical reasoning based on type-level relational similarity, then showed that it could be used to solve both standard analogical tests and compound noun interpretation problems. Conversely, the type-relational model I have developed for compound interpretation could provide a promising method for SAT-style analogy tasks. Given the connection between the two tasks that has been observed by Turney, the combined lexical-relational method I have shown to perform successfully on compound data may well improve further on standard relational approaches to analogy problems.

Coupling lexical and relational models: The method I have used to combine lexical and relational methods is simple and has been shown to work quite well. However, it is limited in that it treats lexical and relational similarity as independent and combines the two models only after the kernel computation stage. A method for coupling the lexical and relational similarity models, allowing each to inform the estimation of the other, could potentially yield more powerful combined models. For example, such an approach might be effective at tackling problems caused by polysemy. Lexical co-occurrence distributions for polysemous words conflate information about various senses, introducing noise into the resulting similarity models. On the other hand, relational similarity models may be more robust to this phenomenon due to a "one sense per collocation" effect: in contexts where one constituent of a word pair co-occurs with the other constituent, that word is likely to be used in its appropriate sense. Furthermore, other words occurring in these contexts are also likely to be particularly informative. The knowledge about word senses learned by the relational model might thus be able to guide the co-occurrence distributions learned by the lexical model. It is not obvious how dynamics of this sort can best be captured, however. One promising direction is suggested by recent work on multitask learning in which different classification models sharing a common pool of knowledge are trained in parallel. Ando and Zhang (2005) and Collobert and Weston (2008) show that this approach can work very well on natural language tasks.

References

ACE, 2008. *Automatic Content Extraction 2008 Evaluation Plan.* Available online at `http://www.nist.gov/speech/tests/ace/ace08/doc/ace08-evalplan.v1.2d.pdf`.

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (DL-00)*, San Antonio, TX.

M. Aizerman, E. Braverman, and L. Rozonoer. 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837.

Rie Kubota Ando and Tong Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI.

R. Harald Baayen. 2001. *Word Frequency Distributions.* Kluwer, Dordrecht.

Anna Babarczy, John Carroll, and Geoffrey Sampson. 2006. Definitional, personal, and mechanical constraints on part of speech annotation performance. *Journal of Natural Language Engineering*, 12(1):77–90.

Ross Baldick. 2006. *Applied Optimization: Formulation and Algorithms for Engineering Systems.* Cambridge University Press, Cambridge.

Timothy Baldwin and Takaaki Tanaka. 2004. Translation by machine of complex nominals: Getting it right. In *Proceedings of the ACL-04 Workshop on Multiword Expressions: Integrating Processing*, Barcelona, Spain.

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02)*, Mexico City, Mexico.

Colin Bannard, Timothy Baldwin, and Alex Lascarides. 2003. A statistical approach to the semantics of verb-particles. In *Proceedings of the ACL-03 SIGLEX Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan.

Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2006. A semantic kernel to classify texts with very few training examples. *Informatica*, 30(2):163–172.

Laurie Bauer. 1979. On the need for pragmatics in the study of nominal compounding. *Journal of Pragmatics*, 3(1):45–50.

Laurie Bauer. 1998. When is a sequence of noun + noun a compound in English? *English Language and Linguistics*, 2(1):65–86.

Laurie Bauer. 2001. Compounding. In Martin Haspelmath, editor, *Language Typology and Language Universals.* Mouton de Gruyter, The Hague.

Brandon Beamer, Suma Bhat, Brant Chee, Andrew Fister, Alla Rozovskaya, and Roxana Girju. 2007. UIUC: A knowledge-rich approach to identifying semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-07)*, Prague, Czech Republic.

Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. 2003. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208.

Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. 1984. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Springer, Berlin.

Adam L. Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

A. Bhattacharyya. 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–110.

Douglas Biber and Victoria Clark. 2002. Historical shifts in modification patterns with complex noun phrase structures. In Teresa Fanego, Javier Pérez-Guerra, and María José López-Couso, editors, *English Historical Syntax and Morphology*. John Benjamins, Amsterdam.

Daniel M. Bikel. 2004. A distributional analysis of a lexicalized statistical parsing model. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain.

Nicole J.-M. Blackman and John J. Koval. 2000. Interval estimation for Cohen's kappa as a measure of agreement. *Statistics in Medicine*, 19(5):723–741.

B. K. Boguraev and K. Spärck Jones. 1983. How to drive a database front end using general semantic information. In *Proceedings of the 1st Conference on Applied Natural Language Processing (ANLP-83)*, Santa Monica, CA.

Lera Boroditsky. 2000. Metaphoric structuring: Understanding time through spatial metaphors. *Cognition*, 75(1):1–28.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (COLT-92)*, Pittsburgh, PA.

Léon Bottou and Chih-Jen Lin. 2007. Support vector machine solvers. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large-Scale Kernel Machines*. MIT Press, Cambridge, MA.

Thorsten Brants and Alex Franz, 2006. *Web 1T 5-gram Corpus Version 1.1*. Linguistic Data Consortium.

Sergei Brin. 1998. Extracting patterns and relations from the World Wide Web. In *Proceedings of the International Workshop on the Web and Databases (WebDB-98)*, Valencia, Spain.

Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-02)*, Las Palmas, Spain.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the ACL-06 Interactive Presentation Sessions*, Sydney, Australia.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Razvan Bunescu and Raymond J. Mooney. 2005a. A shortest path dependency kernel for relation extraction. In *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP-05)*, Vancouver, Canada.

Razvan Bunescu and Raymond J. Mooney. 2005b. Subsequence kernels for relation extraction. In *Proceedings of the 19th Conference on Neural Information Processing Systems (NIPS-05)*, Vancouver, Canada.

Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the Web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Christopher J. C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Curt Burgess and Kevin Lund. 1998. Modeling cerebral asymmetries in high-dimensional semantic space. In Mark Beeman and Christine Chiarello, editors, *Right Hemisphere Language Comprehension: Perspectives from Cognitive Neuroscience*. Routledge, London.

Lou Burnard, 1995. *Users' Guide for the British National Corpus*. British National Corpus Consortium, Oxford University Computing Service, Oxford, UK.

Ted Byrt, Janet Bishop, and John B. Carlin. 1993. Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46(5):423–429.

Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean-Michel Renders. 2003. Word-sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Dongwei Cao, Osama T. Masoud, and Daniel Boley. 2006. Human motion recognition with a convolution kernel. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA-06)*, Orlando, FL.

Sharon Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, College Park, MD.

George Cardona. 1976. *Pāṇini: A Survey of Research*. Mouton, The Hague.

Bob Carpenter. 2008. Comment on "continuing bad ideas". Blog comment published as `http://nlpers.blogspot.com/2008/05/continuing-bad-ideas.html?showComment=1211477580000#c7793782576140289269`, May.

Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: A library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Olivier Chapelle, Patrick Haffner, and Vladimir N. Vapnik. 1999. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006a. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, Sydney, Australia.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006b. Unsupervised relation disambiguation with order identification capabilities. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP-06)*, Sydney, Australia.

Kenneth W. Church and Patrick Hanks. 1989. Word association norms, mutual information and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL-89)*, Vancouver, Canada.

Philipp Cimiano and Johanna Wenderoth. 2007. Automatic acquisition of ranked qualia structures from the Web. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Stephen Clark and James R. Curran. 2007. Formalism-independent parser evaluation with CCG and DepBank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Herbert H. Clark. 1973. Space, time, semantics, and the child. In Timothy E. Moore, editor, *Cognitive Development and the Acquisition of Language*. Academic Press, New York.

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of the 15th Conference on Neural Information Processing Systems (NIPS-01)*, Vancouver, Canada.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML-08)*, Helsinki, Finland.

Ann Copestake and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-00)*, Athens, Greece.

Ann Copestake and Alex Lascarides. 1997. Integrating symbolic and statistical representations: The lexicon-pragmatics interface. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, Madrid, Spain.

Corinna Cortes and Vladimir Vapnik. 1995. Support vector networks. *Machine Learning*, 20(3):273–297.

Fintan Costello and Mark T. Keane. 2000. Efficient creativity: Constraints on conceptual combination. *Cognitive Science*, 24(2):299–349.

Seana Coulson. 2001. *Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction*. Cambridge University Press, Cambridge.

Koby Crammer and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

Nello Cristianini, Jaz Kandola, Andre Elisseeff, and John Shawe-Taylor. 2001. On kernel target alignment. Technical Report NC-TR-01-087, NeuroCOLT.

D. A. Cruse. 1986. *Lexical Semantics*. Cambridge University Press, Cambridge.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain.

James Curran. 2003. *From Distributional to Semantic Similarity*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Marco Cuturi, Kenji Fukumizu, and Jean-Philippe Vert. 2005. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198.

Marco Cuturi. 2007. Permanents, transportation polytopes and positive definite kernels on histograms. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India.

Ido Dagan, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, Madrid, Spain.

Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1–4):43–69.

Dmitry Davidov and Ari Rappoport. 2008. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, Columbus, OH.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Barry Devereux and Fintan Costello. 2005. Investigating the relations used in conceptual combination. *Artificial Intelligence Review*, 24(3-4):489–515.

Barry Devereux and Fintan Costello. 2007. Learning to interpret novel noun-noun compounds: Evidence from a category learning experiment. In *Proceedings of the ACL-07 Workshop on Cognitive Aspects of Computational Language Acquisition*, Prague, Czech Republic.

Barbara Di Eugenio. 2000. On the usage of Kappa to evaluate agreement on coding tasks. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-00)*, Athens, Greece.

Lee R. Dice. 1945. Measures of the amount of ecologic association between species. *Journal of Ecology*, 26(3):297–302.

Thomas G. Dietterich, Richard H. Lathrop, and Tomas Lozano-Perez. 1997. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

David M. Doolin, Jack Dongarra, and Keith Seymour. 1999. JLAPACK – Compiling LAPACK Fortran to Java. *Scientific Programming*, 7(2):111–138. Software available at `http://icl.cs.utk.edu/f2j/software/index.html`.

Pamela Downing. 1977. On the creation and use of English compound nouns. *Language*, 53(4):810–842.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Dominik M. Endres and Johannes E. Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860.

Ute Essen and Volker Steinbiss. 1992. Co-occurrence smoothing for stochastic language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-92)*, San Francisco, CA.

Zachary Estes and Lara L. Jones. 2006. Priming via relational similarity: A *copper horse* is faster when seen through a *glass eye*. *Journal of Memory and Language*, 55(1):89–101.

Zachary Estes. 2003. Attributive and relational processes in nominal combination. *Journal of Memory and Language*, 48(2):304–319.

Stefan Evert. 2004. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Afsaneh Fazly and Suzanne Stevenson. 2006. Automatically constructing a lexicon of verb phrase idiomatic combinations. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, Trento, Italy.

Ronen Feldman and Benjamin Rosenfeld. 2006. Boosting unsupervised relation extraction by using NER. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP-06)*, Sydney, Australia.

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

Tim Finin. 1980. The semantic interpretation of nominal compounds. In *Proceedings of the 1st National Conference on Artificial Intelligence (AAAI-80)*, Stanford, CA.

J. R. Firth. 1957. A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford.

Joseph L. Fleiss, Bruce A. Levin, and Myunghee Cho Paik. 2003. *Statistical Methods for Rates and Proportions*. Wiley, Hoboken, NJ, 3rd edition.

Bent Fuglede. 2005. Spirals in Hilbert space: With an application in information theory. *Expositiones Mathematicae*, 23(1):23–45.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of The 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India.

Christina L. Gagné and Edward J. Shoben. 1997. Influence of thematic relations on the comprehension of modifier-noun combinations. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 23(1):71–87.

Christina L. Gagné and Edward J. Shoben. 2002. Priming relations in ambiguous noun-noun compounds. *Memory and Cognition*, 30(4):637–646.

Christina L. Gagné, Thomas L. Spalding, and Melissa C. Gorrie. 2005a. Sentential context and the interpretation of familiar open-compounds and novel modifier-noun phrases. *Language and Speech*, 48(2):203–221.

Christina L. Gagné, Thomas L. Spalding, and Hongbo Ji. 2005b. Re-examining evidence for the use of independent relational representations during conceptual combination. *Journal of Memory and Language*, 53(3):445–455.

Christina L. Gagné. 2002. Lexical and relational influences on the processing of novel compounds. *Brain and Language*, 81(1–3):723–735.

Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. 2002. Multi-instance kernels. In *Proceedings of the 19th International Conference on Machine Learning (ICML-02)*, Sydney, Australia.

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. *Computer Speech and Language*, 19(4):479–496.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 Task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-07)*, Prague, Czech Republic.

Roxana Girju. 2006. Out-of-context noun phrase semantic interpretation with cross-linguistic evidence. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM-06)*, Arlington, VA.

Roxana Girju. 2007a. Experiments with an annotation scheme for a knowledge-rich noun phrase interpretation system. In *Proceedings of the ACL-07 Linguistic Annotation Workshop*, Prague, Czech Republic.

Roxana Girju. 2007b. Improving the interpretation of noun phrases with cross-linguistic information. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Claudio Giuliano, Alberto Lavelli, Daniele Pighin, and Lorenza Romano. 2007. FBK-IRST: Kernel methods for semantic relation extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-07)*, Prague, Czech Republic.

Lila R. Gleitman and Henry Gleitman. 1970. *Phrase and Paraphrase: Some Innovative Uses of Language*. Norton, New York.

Alfio Gliozzo, Claudio Giuliano, and Carlo Strapparava. 2005. Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI.

Mark Goadrich, Louis Oliphant, and Jude Shavlik. 2004. Learning ensembles of first-order clauses for recall-precision curves: A case study in biomedical information extraction. In *Proceedings of the 14th International Conference on Inductive Logic Programming (ILP-04)*, Porto, Portugal.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda, 2005. *English Gigaword Corpus, 2nd Edition*. Linguistic Data Consortium, Philadelphia, PA.

Kristen Grauman and Trevor Darrell. 2007. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760.

Mark A. Greenwood and Mark Stevenson. 2006. Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the ACL-06 Workshop on Information Extraction Beyond the Document*, Sydney, Australia.

Gregory Grefenstette. 1993. Evaluation techniques for automatic semantic extraction: Comparing syntactic and window based approaches. In *Proceedings of the ACL-93 Workshop on Acquisition of Lexical Knowledge from Text*, Columbus, OH.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer, Dordrecht.

Jakob Grimm. 1826. *Deutsche Grammatik, Theil 2*. Dieterich, Göttingen.

Ralph Grishman and John Sterling. 1994. Generalizing automatically generated selectional patterns. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan.

Bernard Haasdonk. 2005. Feature space interpretation of SVMs with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492.

Shelby J. Haberman. 1973. The analysis of residuals in cross-classified tables. *Biometrics*, 29(1):205–220.

Ben Hachey. 2006. Comparison of similarity models for the relation discovery task. In *Proceedings of the ACL-06 Workshop on Linguistic Distances*, Sydney, Australia.

Patrick Hanks. 2000. Do word meanings exist? *Computers and the Humanities*, 34(1–2):205–215.

Kenneth E. Harper. 1965. Measurement of similarity between nouns. In *Proceedings of the 1965 International Conference on Computational Linguistics (COLING-65)*, New York, NY.

Brian Harrington and Stephen Clark. 2007. ASKNet: Automated semantic knowledge network. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, Vancouver, Canada.

Zellig S. Harris. 1968. *Mathematical Structures of Language*. Interscience, New York.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer, Berlin.

Anna G. Hatcher. 1960. An introduction to the analysis of English noun compounds. *Word*, 16:356–373.

David Haussler. 1999. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Computer Science Department, University of California at Santa Cruz.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France.

Matthias Hein and Olivier Bousquet. 2005. Hilbertian metrics and positive definite kernels on probability measures. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS-05)*, Barbados.

Matthias Hein, Thomas Navin Lal, and Olivier Bousquet. 2004. Hilbertian metrics on probability measures and their application in SVM's. In *Proceedings of the 26th DAGM Pattern Recognition Symposium*, Tübingen, Germany.

Matthias Hein, Olivier Bousquet, and Bernhard Schölkopf. 2005. Maximal margin classification for metric spaces. *Journal of Computer and System Sciences*, 71(3):333–354.

Ralf Herbrich, Thore Graepel, and Colin Campbell. 2001. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279.

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL-90)*, Pittsburgh, PA.

Graeme Hirst and Alexander Budanitsky. 2005. Correcting real-world spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1):87–111.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Jerry R. Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

Eduard Hoenkamp and Rob de Groot. 2000. Finding relevant passages using noun-noun compounds: Coherence vs. proximity. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-00)*, pages 385–387, Athens, Greece.

Eduard Hovy. 2005. Methodologies for the reliable construction of ontological knowledge. In *Proceedings of the 13th Annual Conference on Conceptual Structures (ICCS-05)*, Kassel, Germany.

Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2008. A practical guide to support vector classification. Technical report, Dept. of Computer Science, National Taiwan University. Available online at `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

Falk Huettig, Philip T. Quinlan, Scott A. McDonald, and Gerry T. M. Altmann. 2006. Models of high-dimensional semantic space predict language-mediated eye movements in the visual world. *Acta Psychologica*, 121(1):65–80.

Richard D. Hull and Fernando Gomez. 1996. Semantic interpretation of nominalizations. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR.

Pierre Isabelle. 1984. Another look at nominal compounds. In *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84)*, Stanford, CA.

Tommi S. Jaakola and David Haussler. 1998. Exploiting generative models in discriminative classifiers. In *Proceedings of the 12th Conference on Neural Information Processing Systems (NIPS-98)*, Denver, CO.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579.

Ray Jackendoff. 2002. *Foundations of Language*. Oxford University Press, Oxford.

Tony Jebara, Risi Kondor, and Andrew Howard. 2004. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844.

Otto Jespersen. 1942. *A Modern English Grammar on Historical Principles, Part VI: Morphology*. Ejnar Munksgaard, Copenhagen.

Jing Jiang and Chengxiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Proceedings of the 2007 Human Language Technology Conference and Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-07)*, Rochester, NY.

Thorsten Joachims, Nello Cristianini, and John Shawe-Taylor. 2001. Composite kernels for hypertext categorisation. In *Proceedings of the 18th International Conference on Machine Learning (ICML-01)*, Williamstown, MA.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM Conference on Knowledge Discovery and Data Mining (KDD-06)*, Philadelphia, PA.

Michael Johnston and Frederica Busa. 1996. Qualia structure and the compositional interpretation of compounds. In *Proceedings of the ACL-96 SIGLEX Workshop on Breadth and Depth of Semantic Lexicons*, Santa Cruz, CA.

S. D. Joshi. 1968. *Patañjali's Vyākaraṇa-Mahābhaṣya: Samarthāhnika (P 2.1.1). Edited with Translation and Explanatory Notes.* University of Poona Press, Poona.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL-04 Poster Session*, Barcelona, Spain.

Jussi Karlgren. 2005. Compound terms and their constituent elements in information retrieval. In *15th Nordic Conference of Computational Linguistics (NODALIDA-05)*, Joensuu, Finland.

William Karush. 1939. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Dept. of Mathematics, University of Chicago.

Rohit J. Kate and Raymond J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, Sydney, Australia.

Boaz Keysar and Bridget Bly. 1995. Intuitions of the transparency of idioms: Can one keep a secret by spilling the beans? *Journal of Memory and Language*, 34(1):89–109.

Adam Kilgarriff and Colin Yallop. 2000. What's in a thesaurus? In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-00)*, Athens, Greece.

Adam Kilgarriff. 2004. How dominant is the commonest sense of a word? In *Proceedings of the 7th International Conference on Text, Speech and Dialogue (TSD-04)*, Brno, Czech Republic.

Su Nam Kim and Timothy Baldwin. 2005. Automatic interpretation of noun compounds using WordNet similarity. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, Jeju Island, Korea.

Su Nam Kim and Timothy Baldwin. 2006. Interpreting semantic relations in noun compounds via verb semantics. In *Proceedings of the ACL-06 Main Conference Poster Session*, Sydney, Australia.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, Budapest, Hungary.

Risi Kondor and Tony Jebara. 2003. A kernel between sets of vectors. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, Washington, DC.

Risi Kondor. 2005. Computing the Bhattacharyya kernel. Unpublished note, available online at `http://www.gatsby.ucl.ac.uk/~risi/papers/computeBhatta.ps`.

Anna Korhonen, Yuval Krymolowski, and Zvika Marx. 2003. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan.

Helena Chmura Kraemer, Vyjeyanthi S. Periyakoil, and Art Noda. 2004. Kappa coefficients in medical research. In R. B. D'Agostino, editor, *Tutorials in Biostatistics, Volume 1: Statistical Methods in Clinical Studies*. Wiley, Chichester.

R. M. Krauss and S. Weinheimer. 1964. Changes in the length of reference phrases as a function of social interaction : A preliminary study. *Psychonomic Science*, 1:113–114.

Brigitte Krenn and Stefan Evert. 2004. Determining intercoder agreement for a collocation identification task. In *Tagungsband der 7. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS-04)*, Vienna, Austria.

Andrea Krott, Robert Schreuder, and R. Harald Baayen. 2002. Linking elements in Dutch noun-noun compounds: Constituent families as analogical predictors for response latencies. *Brain and Language*, 81(1–3):708–722.

Andrea Krott, Robert Schreuder, R. Harald Baayen, and Wolfgang U. Dressler. 2007. Analogical effects on linking elements in German compounds. *Language and Cognitive Processes*, 22(1):25–57.

Harold W. Kuhn and Albert W. Tucker. 1951. Nonlinear programming. In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, Berkeley, CA.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

John Lafferty and Guy Lebanon. 2005. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163.

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

Ronald W. Langacker. 1999. *Grammar and Conceptualization*. Mouton de Gruyter, Berlin.

Mirella Lapata and Frank Keller. 2004. The Web as a baseline: Evaluating the performance of unsupervised Web-based models for a range of NLP tasks. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-04)*, Boston, MA.

Mirella Lapata and Alex Lascarides. 2003. Detecting novel compounds: The role of distributional evidence. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, Budapest, Hungary.

Maria Lapata. 2002. The disambiguation of nominalisations. *Computational Linguistics*, 28(3):357–388.

Sabine Lappe and Ingo Plag. 2007. The variability of compound stress in English: Towards an exemplar-based alternative to the compound stress rule. In *Proceedings of the ESSLLI-07 Workshop on Exemplar-Based Models of Language Acquisition and Use*, Dublin, Ireland.

Mark Lauer. 1995. *Designing Statistical Language Learners: Experiments on Compound Nouns*. Ph.D. thesis, Macquarie University.

Michael Lebowitz. 1988. The use of memory in text processing. *Communications of the ACM*, 31(12):1483–1502.

J. Jack Lee and Z. Nora Tu. 1994. A better confidence interval for kappa (κ) on measuring agreement between two raters with binary outcomes. *Journal of Computational and Graphical Statistics*, 3(3):301–321.

Yoonkyung Lee, Yi Lin, and Grace Wahba. 2004. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, College Park, MD.

Robert B. Lees. 1970. Problems in the grammatical analysis of English nominal compounds. In Manfred Bierwisch and Karl Erich Heidolph, editors, *Progress in Linguistics*. Mouton de Gruyter, The Hague.

Wendy Lehnert. 1988. The analysis of nominal compounds. In Umberto Eco, Marco Santambrogio, and Patrizia Violi, editors, *Meaning and Mental Representations*. Indiana University Press, Bloomington, IN.

Rosemary Leonard. 1984. *The Interpretation of English Noun Sequences on the Computer*. North-Holland, Amsterdam.

Edda Leopold and Jörg Kindermann. 2002. Text categorization with support vector machines. How to represent texts in input space? *Machine Learning*, 46(1–3):423–444.

Michael Lesk. 1985. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC-85)*, Ithaca, NY.

Christina Leslie and Rui Kuang. 2004. Fast string kernels using inexact matching for protein sequences. *Journal of Machine Learning Research*, 5:1435–1455.

Judith N. Levi. 1978. *The Syntax and Semantics of Complex Nominals*. Academic Press, New York.

Roger Levy and T. Florian Jaeger. 2006. Speakers optimize information density through syntactic reduction. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS-06)*, Vancouver, Canada.

Ping Li and Kenneth W. Church. 2007. A sketch algorithm for estimating two-way and multi-way associations. *Computational Linguistics*, 33(3):305–354.

Charles N. Li. 1971. *Semantics and the structure of compounds in Chinese*. Ph.D. thesis, University of California, Berkeley.

Gary Libben. 2006. Why study compound processing? An overview of the issues. In Gary Libben and Gonia Jarema, editors, *The Representation and Processing of Compound Words*. Cambridge University Press, Cambridge.

Mark Liberman and Richard Sproat. 1992. The stress and structure of modified noun phrases in English. In Ivan A. Sag and Anna Szabolcsi, editors, *Lexical Matters*. CSLI Publications, Stanford.

Rochelle Lieber. 2004. *Morphology and Lexical Semantics*. Cambridge University Press, Cambridge.

Friedrich Liese and Igor Vajda. 2006. On divergences and informations in statistical information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Jinhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL-98)*, Montreal, Canada.

Dekang Lin. 1998b. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, Madison, WI.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, College Park, MD.

Ken Litkowski and Orin Hargraves. 2007. SemEval-2007 Task 06: Word-sense disambiguation of prepositions. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-07)*, Prague, Czech Republic.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Christopher J. C. H. Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. 2003. Investigating semantic similarity measures across the Gene Ontology: The relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283.

Will Lowe and Scott McDonald. 2000. The direct route: Mediated priming in semantic space. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society (CogSci-00)*, Philadelphia, PA.

Siwei Lyu. 2005a. Kernels for unordered sets: The Gaussian mixture approach. In *Proceedings of the 16th European Conference on Machine Learning (ECML-05)*, Porto, Portugal.

Siwei Lyu. 2005b. Mercer kernels for object recognition with local features. In *Proceedings of the IEEE Computer Society International Conference on Computer Vision and Pattern Recognition (CVPR-05)*, San Diego, CA.

Peter Master. 1993. On-line nominal compound formation in an experimental pidgin. *Journal of Pragmatics*, 20(4):359–375.

Margaret Masterman. 1956. The potentialities of a mechanical thesaurus. *MT: Mechanical Translation*, 11(3):369–390.

Diana McCarthy, Bill Keller, and John Carroll. 2003. Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL-03 SIGLEX Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan.

David McDonald and Frederick Hayes-Roth. 1978. Inferential searches of knowledge networks as an approach to extensible language-understanding systems. In D. A. Waterman and Frederick Hayes-Roth, editors, *Pattern-Directed Inference Systems*. Academic Press, New York.

Scott A. McDonald and Richard C. Shillcock. 2001. Rethinking the word frequency effect: The neglected role of distributional information in lexical processing. *Language and Speech*, 44(3):295–323.

J. Mercer. 1909. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London, Series A*, 209:415–446.

Rada Mihalcea. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP-05)*, Vancouver, Canada.

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-00)*, Seattle, WA.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of the 1st International Natural Language Generation Conference (INLG-00)*, Mitzpe Ramon, Israel.

Tom M. Mitchell, Svetlana V. Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L. Malave, Robert A. Mason, and Marcel Adam Just. 2008. Predicting human brain activity associated with the meanings of nouns. *Science*, 320(5880):1191–1195.

Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. 2004. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37(6):396–410.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. *Proceedings of the 17th European Conference on Machine Learning (ECML-06)*.

Gregory L. Murphy. 1990. Noun phrase interpretation and conceptual combination. *Journal of Memory and Language*, 29(3):259–288.

Preslav Nakov and Marti Hearst. 2005. Search engine statistics beyond the n-gram: Application to noun compound bracketing. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL-05)*, Ann Arbor, MI.

Preslav Nakov and Marti Hearst. 2006. Using verbs to characterise noun-noun relations. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems and Applications (AIMSA-06)*, Varna, Bulgaria.

Preslav Nakov and Marti Hearst. 2007a. UCB system description for the WMT 2007 shared task. In *Proceedings of the 2nd Workshop on Statistical Machine Translation (WMT-07)*, Prague, Czech Republic.

Preslav I. Nakov and Marti A. Hearst. 2007b. UCB: System description for SemEval task #4. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-07)*, Prague, Czech Republic.

Preslav I. Nakov. 2007. *Using the Web as an Implicit Training Set: Application to Noun Compound Syntax and Semantics*. Ph.D. thesis, EECS Department, University of California at Berkeley.

Preslav Nakov. 2008. Noun compound interpretation using paraphrasing verbs: Feasibility study. In *Proceedings of the 13th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA-08)*, Varna, Bulgaria.

Vivi Nastase and Stan Szpakowicz. 2001. Unifying semantic relations across syntactic levels. In *Proceedings of the EuroConference on Recent Advances in NLP (RANLP-01)*, Tzigov Chark, Bulgaria.

Vivi Nastase and Stan Szpakowicz. 2003. Exploring noun-modifier semantic relations. In *Proceedings of the 5th International Workshop on Computational Semantics (IWCS-03)*, Tilburg, The Netherlands.

Vivi Nastase, Jelber Sayyad Shirabad, Marina Sokolova, and Stan Szpakowicz. 2006. Learning noun-modifier semantic relations with corpus-based and WordNet-based features. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, Boston, MA.

M. E. J. Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):323–351.

Jeremy Nicholson and Timothy Baldwin. 2006. Interpretation of compound nominalisations using corpus and Web statistics. In *Proceedings of the ACL-06 Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, Sydney, Australia.

Zheng-Yu Niu, Dong-Hong Ji, and Chew Lim Tan. 2005. Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI.

Adolf Noreen. 1904. *Vårt Språk*, volume 5. C. W. K. Gleerups Förlag, Lund.

Paul Nulty. 2007a. Semantic classification of noun phrases using web counts and learning algorithms. In *Proceedings of the ACL-07 Student Research Workshop*, Prague, Czech Republic.

Paul Nulty. 2007b. UCD-PN: Classification of semantic relations between nominals using WordNet and web counts. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-07)*, Prague, Czech Republic.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.

Diarmuid Ó Séaghdha and Ann Copestake. 2007. Co-occurrence contexts for noun compound interpretation. In *Proceedings of the ACL-07 Workshop on A Broader Perspective on Multiword Expressions*, Prague, Czech Republic.

Diarmuid Ó Séaghdha and Ann Copestake. 2008. Semantic classification with distributional kernels. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, Manchester, UK.

Diarmuid Ó Séaghdha. 2007a. Annotating and learning compound noun semantics. In *Proceedings of the ACL-07 Student Research Workshop*, Prague, Czech Republic.

Diarmuid Ó Séaghdha. 2007b. Designing and evaluating a semantic annotation scheme for compound nouns. In *Proceedings of the 4th Corpus Linguistics Conference (CL-07)*, Birmingham, UK.

Sebastian Padó and Mirella Lapata. 2003. Constructing semantic space models from parsed corpora. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan.

Ted Pedersen, Serguei V. S. Pakhomov, Siddarth Patwardhan, and Christopher G. Chute. 2007. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288–299.

Bolette Sanford Pedersen. 2007. Using shallow linguistic analysis to improve search on Danish compounds. *Natural Language Engineering*, 13(1):75–90.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-93)*, Columbus, OH.

Ingo Plag. 2006. The variability of compound stress in English: structural, semantic, and analogical factors. *English Language and Linguistics*, 10(1):143–172.

John Platt. 1999. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA.

Massimo Poesio, Uwe Reyle, and Rosemary Stevenson. 2006. Justified sloppiness in anaphoric reference. In Harry Bunt and Reinhard Muskens, editors, *Computing Meaning 3*. Springer, Berlin.

Massimo Poesio. 1996. Semantic ambiguity and perceived ambiguity. In Kees van Deemter and Stanley Peters, editors, *Ambiguity and Underspecification*. CSLI Publications, Stanford, CA.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2004. Support vector learning for semantic argument classification. *Machine Learning*, 60(1):11–39.

J. Pustejovsky, J. Castaño, J. Zhang, M. Kotecki, and B. Cochran. 2002. Robust relational parsing over biomedical literature: Extracting *inhibit* relations. In *Proceedings of the 7th Pacific Symposium on Biocomputing (PSB-02)*, Lihue, Hawaii.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press, Cambridge, MA.

Ulrike Rackow, Ido Dagan, and Ulrike Schwall. 1992. Automatic translation of noun compounds. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France.

Claudine N. Raffray, Martin J. Pickering, and Holly P. Branigan. 2007. Priming the interpretation of noun-noun compounds. *Journal of Memory and Language*, 57(3):380–395.

C. R. Rao. 1982. Diversity and dissimilarity coefficients: A unified approach. *Theoretical Population Biology*, 21(1):24–43.

C. R. Rao. 1987. Differential metrics in probability spaces. In Shun-Ichi Amari, O. E. Barndorff-Nielsen, Robert E. Kass, Steffen L. Lauritzen, and C. R. Rao, editors, *Differential Geometry in Statistical Inference*. Institute of Mathematical Statistics, Hayward, CA.

Soumya Ray and Mark Craven. 2001. Representing sentence structure in hidden Markov models for information extraction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, Seattle, WA.

Konrad Rieck and Pavel Laskov. 2008. Linear-time computation of similarity measures for sequential data. *Journal of Machine Learning Research*, 9:23–48.

Ryan Rifkin and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141.

Brian Roark and Eugene Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL-98)*, Montreal, Canada.

Barbara Rosario and Marti Hearst. 2001. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-01)*, Pittsburgh, PA.

Barbara Rosario and Marti A. Hearst. 2004. Classifying semantic relations in bioscience texts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain.

Barbara Rosario and Marti A. Hearst. 2005. Multi-way relation classification: Application to protein-protein interactions. In *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP-05)*, Vancouver, Canada.

Barbara Rosario, Marti A. Hearst, and Charles Fillmore. 2002. The descent of hierarchy, and selection in relational semantics. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA.

Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of similarity. *Communications of the ACM*, 8(10):627–633.

Sylvia Weber Russell. 1972. Semantic categories of nominals for conceptual dependency analysis of natural language. Computer Science Department Report CS-299, Stanford University.

Mary Ellen Ryder. 1994. *Ordered Chaos: The Interpretation of English Noun-Noun Compounds*. University of California Press, Berkeley, CA.

Jerrold M. Sadock. 1998. On the autonomy of compounding morphology. In Steven G. Lapointe, Diane K. Brentari, and Patrick M. Farrell, editors, *Morphology and its Relation to Phonology and Syntax*. CSLI Publications, Stanford, CA.

Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain the neck for NLP. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02)*, Mexico City, Mexico.

Geoffrey Sampson and Anna Babarczy. 2006. Definitional and human constraints on structural annotation of English. In *Proceedings of the 2nd Conference on Quantitative Investigations in Theoretical Linguistics (QITL-06)*, Osnabrück, Germany.

Michael F. Schober and Herbert H. Clark. 1989. Understanding by addressees and overhearers. *Cognitive Psychology*, 21:211–232.

I. J. Schoenberg. 1938. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536.

Bernhard Schölkopf. 2000. The kernel trick for distances. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems (NIPS-00)*, Denver, CO.

Sabine Schulte im Walde. 2008. Human associations and the choice of features for semantic verb classification. *Research on Language and Communication*, 6(1):79–111.

Issai Schur. 1911. Bemerkungen zur Theorie der beschränkten Bilinearformen mit unendlich vielen Veränderlichen. *Journal fur reine und angewandte Mathematik*, 140:1–28.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings of Supercomputing '92*.

Elisabeth Selkirk. 1982. *The Syntax of Words*. MIT Press, Cambridge, MA.

Amnon Shashua and Tamir Hazan. 2004. Algebraic set kernels with application to inference over local image representations. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS-04)*, Vancouver, Canada.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge.

Libin Shen, Giorgio Satta, and Aravind K. Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Georges Siolas and Florence d'Alche-Buc. 2000. Support vector machines based on a semantic kernel for text categorization. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy.

Noam Slonim and Naftali Tishby. 2000. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-00)*, Athens, Greece.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. 2007. A Hilbert space embedding for distributions. In *Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT-07)*, Sendai, Japan.

Anders Søgaard. 2005. Compounding theories and linguistic diversity. In Zygmunt Frajzyngier, Adam Hodges, and David S. Rood, editors, *Linguistic Diversity and Language Theories*. John Benjamins, Amsterdam.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. 2006. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 6:1531–1565.

Sören Sonnenburg, Gunnar Rätsch, and Konrad Rieck. 2007. Large scale learning with string kernels. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large Scale Kernel Machines*. MIT Press, Cambridge, MA.

Karen Spärck Jones. 1964. *Synonymy and Semantic Classification*. Ph.D. thesis, University of Cambridge.

Mark Stevenson and Mark A. Greenwood. 2005. A semantic approach to IE pattern induction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI.

Michael Strube and Simone Paolo Ponzetto. 2006. WikiRelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, Boston, MA.

Stanley Y. W. Su. 1969. A semantic theory based upon interactive meaning. Computer Sciences Technical Report #68, University of Wisconsin.

Jun Suzuki, Tsutomu Hirao, Yutaka Sasaki, and Eisaku Maeda. 2003. Hierarchical directed acyclic graph kernel: Methods for structured natural language data. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan.

Georgios Tagalakis and Mark T. Keane. 2005. How understanding novel compounds is facilitated by priming from similar, known compounds. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society (CogSci-05)*, Stresa, Italy.

Leonard Talmy. 2000. The semantics of causation. In *Toward a Cognitive Semantics, Volume 1: Concept Structuring Systems*. MIT Press, Cambridge, MA.

John R. Taylor. 1996. *Possessives in English: An Exploration in Cognitive Grammar*. Oxford University Press, Oxford.

Junji Tomita, Stephen Soderland, and Oren Etzioni. 2006. Expanding the recall of relation extraction by bootstrapping. In *Proceedings of the EACL-06 Workshop on Adaptive Text Extraction and Mining*, Trento, Italy.

Flemming Topsøe. 2000. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, 46(4):1602–1609.

Flemming Topsøe. 2003. Jensen-Shannon divergence and norm-based measures of discrimination and variation. Unpublished manuscript available at `http://www.math.ku.dk/~topsoe/sh.ps`.

Peter D. Turney and Michael L. Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1–3):251–278.

Peter D. Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the 2003 International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, Borovets, Bulgaria.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Peter D. Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, Manchester, UK.

Amos Tversky. 1977. Features of similarity. *Psychological Review*, 84(4):327–352.

C. J. van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London, 2nd edition. Available online at `www.dcs.gla.ac.uk/Keith/Preface.html`.

Lucy Vanderwende. 1994. Algorithm for automatic interpretation of noun sequences. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Wiley, New York.

S. V. N. Vishwanathan and Alexander J. Smola. 2002. Fast kernels for string and tree matching. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems (NIPS-02)*, Vancouver, Canada.

S. V. N. Vishwanathan, Karsten M. Borgwardt, and Nicol Schraudolph. 2006. Fast computation of graph kernels. In *Proceedings of the 20th Conference on Neural Information Processing Systems (NIPS-06)*, Vancouver, Canada.

Beatrice Warren. 1978. *Semantic Patterns of Noun-Noun Compounds*. Acta Universitatis Gothoburgensis, Göteborg.

Chris Watkins. 2000. Dynamic alignment kernels. In Peter J. Bartlett, Bernhard Schölkopf, Dale Schuurmans, and Alex J. Smola, editors, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA.

Julie Weeds and David Weir. 2005. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–476.

Julie Weeds, David Weir, and Bill Keller. 2005. The distributional similarity of subparses. In *Proceedings of the ACL-05 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor, MI.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)*, Taipei, Taiwan.

Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the 2003 Human Language Technology Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-03)*, Edmonton, Canada.

Yorick A. Wilks, Dan C. Fass, Cheng-Ming Guo, James E. McDonald, Tony Plate, and Brian M. Slator. 1989. A tractable machine dictionary as a resource for computational semantics. In Bran Boguraev and Ted Briscoe, editors, *Computational Lexicography for Natural Language Processing*. Longman, London.

Andrew Wilson and Jenny Thomas. 1997. Semantic annotation. In Roger Garside, Geoffrey Leech, and Anthony McEnery, editors, *Corpus Annotation*. Longman, London.

Edward J. Wisniewski. 1997. When concepts combine. *Psychonomic Bulletin and Review*, 4(2):167–183.

Lior Wolf and Amnon Shashua. 2003. Learning over sets using kernel principal angles. *Journal of Machine Learning Research*, 4:913–931.

S. K. M. Wong, Wojciech Ziarko, and Patrick C.N. Wong. 1985. Generalized vector space model in information retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-85)*, Montreal, Canada.

Huilin Xiong, M. N. S. Swamy, and M. Omair Ahmad. 2005. Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks*, 16(2):460–474.

Ichiro Yamada, Timothy Baldwin, Hideki Sumiyoshi, Masahiro Shibata, and Nobuyuki Yagi. 2007. Automatic acquisition of qualia structure from corpus data. *IEICE Transactions on Information and Systems*, E90-D(10):1534–1541.

Annie Zaenen, Jean Carletta, Gregory Garretson, Joan Bresnan, Andrew Koontz-Garboden, Tatiana Nikitina, M. Catherine O'Connor, and Tom Wasow. 2004. Animacy encoding in English: Why and how. In *Proceedings of the ACL-04 Workshop on Discourse Annotation*, Barcelona, Spain.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, Sydney, Australia.

Min Zhang, Wanxiang Che, Aiti Aw, Chew Lim Tan, Guodong Zhou, Ting Liu, and Sheng Li. 2007. A grammar-driven convolution tree kernel for semantic role classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI.

Guodong Zhou, Min Zhang, Donghong Ji, and Qiaoming Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-07*, Prague, Czech Republic.

Karl E. Zimmer. 1971. Some general observations about nominal compounds. *Stanford Working Papers on Linguistic Universals*, 5:C1–C21.

Appendix A

Notational conventions

I have attempted to be as consistent as possible in my use of mathematical notation while also retaining consistency with standard conventions in the literature. The table below covers the notation most frequently used in this thesis. I hope that any ambiguous cases in the text are made clear by their contexts.

$\mathbf{x},\ \mathbf{x}_i, \ldots$	Vectors or ordered sets
$\mathbf{s},\ \mathbf{t}$	Strings
$x_i,\ x_{il},\ s_i$	Elements of the vectors/ordered sets/strings $\mathbf{x}$, $\mathbf{x}_i$ and $\mathbf{s}$ respectively
$a,\ b,\ c, \ldots$	Scalars
$A,\ B,\ C, \ldots$	Matrices or sets
K	Kernel matrix
$\mathcal{X}$	Data input space
$\mathcal{F}$	Feature space associated with a kernel
$k(\cdot, \cdot)$	Kernel function
$f(\cdot),\ g(\cdot), \ldots$	Other functions
$\phi(\cdot)$	Feature space embedding associated with a kernel
$p(\cdot)$	Probability density function
$P(\cdot)$	Probability of an event

Appendix B

Annotation guidelines for compound nouns

1 General Guidelines

The task is to annotate each compound noun N1 N2 with regard to the semantic relation that holds between the constituent nouns N1 and N2. It is assumed that compounds are copulative or semantically right-headed.

Rule 1.1 *The general annotation format is* <RELATION,DIRECTION,RULE>.

RELATION is one of the 10 relation labels defined in section 2. DIRECTION specifies the order of the constituent nouns in the chosen relation's argument structure – in particular, DIRECTION will have the value 1 if the first noun in the compound (N1) fits in the first noun slot mentioned in the rule licensing the chosen relation, and will have value 2 if the second noun in the compound (N2) fits in the rule's first noun slot. RULE is the number of the rule licensing the RELATION. For example:

```
429759: water fern
IN,2,2.1.3.1
This aquatic water fern is a rosette plant which has dense ,
fibrous roots

147862: enemy provocation
ACTOR,1,2.1.4.1
The army said at the weekend that troops had reacted to enemy
provocations and intervened to protect local citizens
```

In the case of *water fern* the IN relation is licensed by Rule 2.1.3.1 *N1/N2 is an object spatially located in or near N2/N1.* Mapping the compound's constituent nouns onto the rule definition, we see that the first slot (*N1/N2 is...*) is filled by N2 *fern* and hence the DIRECTION is 2. For the categories BE, REL, LEX, MISTAG and NONCOMPOUND there is no salient sense of directionality, so it need not be annotated:

```
120214: cedar tree
BE,2.1.1.1
On rising ground at the western end of the churchyard
of St Mary 's at Morpeth in Northumberland stands ,
sheltered by cedar trees , a funerary monument
```

Rule 1.2 *Each compound is presented with its sentential context and should be interpreted within that context. Knowledge of other instances of the compound type are irrelevant.*

A given compound type can have different meanings in different contexts. A *school book* is frequently a *book read IN school*, but it could also be a *book ABOUT school*. A *wood table* might be a *table that IS wood*, but it might also be a *table for chopping wood on* (IN). The intended meaning of a compound token is often clarified by the sentence it appears in.

Rule 1.3 *Where a compound is ambiguous and is not clarified by the sentential context, the most typical meaning of the compound is favoured.*

Compound interpretation must sometimes rely on world knowledge. The compound *school book* is not clarified by a sentence such as *This is a school book*. In this case, *book read IN school* is the most typical interpretation. If the compound's ambiguity arises from the polysemy of a constituent, the same consideration applies. *University* can refer to an institution or its physical location, but in the case of *university degree* the institutional meaning must be correct as locations cannot award degrees, and the compound is labelled ACTOR.

Rule 1.4 *The referent of the compound is of interest only insofar as it elucidates the relation between the constituent nouns. Whether the compound is used metaphorically or not is irrelevant.*

For example: the compound *bird brain* is often used to refer to someone stupid, not to an actual brain, but in both cases the relation (HAVE1) between the constituents is the same; the phrase *in the dog house* contains a metaphorical use of a standard locative compound (IN).

Rule 1.5 *Where a compound consisting of two common nouns is used as a proper noun, and its meaning only differs from its use as a common noun insofar as it denotes a definite entity, it may be annotated as if it were used as a common noun.*

For example: *the Telecommunications Act* (ABOUT), *The Old Tea Shop* (IN), *Castle Hill* (IN). Many names, while constructed from two common nouns, do not seem to encode the same kind of semantics as non-name compounds, e.g. *Penguin Books*, *Sky Television*, *Dolphin Close*, *Coronation Street*. These names encode only a sense of non-specific association between the constituents, and should be classified as REL.

Rule 1.6 *The semantic relation in many compounds involves a characteristic situation or event. Whether such a situation exists for a given compound, and the roles played by its constituents in the situation, will determine which relation labels are available.*

For example, the meaning of *cheese knife* seems to involve an event of cutting, in which *cheese* and *knife* take object and instrument roles respectively. Similarly, *taxi driver*

evokes an event of driving and *night watchman* evokes an event of watching or guarding. The INST and ACTOR relations apply only where such a situation or event is present and where the compound identifies its participant(s). The application of HAVE assumes that the most salient aspect of the underlying situation is possession. It is not strictly necessary to identify the precise nature of the situation or event, only to identify the general roles played by the participants (see the discussion under Rule 2.1.5.1).

Rule 1.7 *Where there is a characteristic situation or event, it is necessary to identify which constituents of the compound are participants and which roles they play.* Participants are entities that can be described as Agent, Instrument, Object or Result:

Agent The instigator of the event, the primary source of energy

Instrument An intermediate entity that is used/acted on by the Agent and in turn exerts force on or changes the Object; more generally, an item which is used to facilitate the event but which is not the Object

Object The entity on which a force is applied or which is changed by the event and which does not exert force on any participant other than the Result. Recipients (e.g. of money or gifts, but not outcomes) also count as Objects.

Result An entity which was not present before and comes into being through the event

For example: $cheese_O\ knife_I$, $taxi_O\ driver_A$, $sneezing_R\ powder_I$. It follows from the above that locations and topics do not count as participants – compounds encoding such roles receive IN and ABOUT labels instead of the ACTOR and INST labels reserved for participants.

The participant role types are listed in order of descending agentivity. We thus have an agentivity hierarchy Agent>Instrument>Object>Result. This ordering plays an important role in distinguishing ACTOR compounds from INST compounds (see Rules 2.1.4 and 2.1.5). It is not necessary to annotate this information, and it is not always necessary to identify the exact participant role of a constituent, so long as the hierarchical order of the constituents can be identified. Identifying participants is only needed to distinguish between relations (ACTOR vs INST) and directionalities.

2 Semantic Relations

Main Relations	
BE	X is N1 and X is N2
HAVE	N1/N2 has N2/N1
IN	N1/N2 is located in N2/N1
ACTOR	N1/N2 is a sentient participant in the event N2/N1
	N1/N2 is sentient and is the more agentive participant of N1 and N2 in an associated event
INST	N1/N2 is a non-sentient participant in the event N2/N1
	N1/N2 is non-sentient and is the more agentive participant of N1 and N2 in an associated event
ABOUT	N1/N2 is about N2/N1
REL	The relation is not described by any of the specific relations but seems productive
LEX	The relation is idiosyncratic and not productive
UNKNOWN	The compound's meaning is unclear
Noncompounds	
MISTAG	N1 and/or N2 have been mistagged and are not common nouns
NONCOMPOUND	The sequence N1 N2 is not a 2-noun compound

2.1 Main Relations

2.1.1 BE

Rule 2.1.1.1 *X is N1 and X is N2.*

For example: *woman driver*, *elm tree*, *distillation process*, *human being*. This rule does not admit sequences such as *deputy chairman*, *fellow man* or *chief executive*, where it is not correct to state that *an [N1 N2] is an N1* (a *chief executive* is not a chief). Such sequences are not to be considered compounds, and their modifiers are to be considered (mistagged) adjectives – see Rule 2.2.1.1.

Rule 2.1.1.2 *N2 is a form/shape taken by the substance N1.*

For example: *stone obelisk*, *chalk circle*, *plastic box*, *steel knife*.

Rule 2.1.1.3 *N2 is ascribed significant properties of N1 without the ascription of identity. The compound roughly denotes "an N2 like N1".*

For example: *father figure*, *angler fish*, *chain reaction*, *pie chart*.

2.1.2 HAVE

Rule 2.1.2.1 *N1/N2 owns N2/N1 or has exclusive rights or the exclusive ability to access or to use N2/N1 or has a one-to-one possessive association with N2/N1.*

For example: *army base*, *customer account*, *government power*. The term *one-to-one possessive association* is intended to cover cases where it seems strange to speak of ownership, for example in the case of inanimate objects (*street name*, *planet atmosphere*).

Rule 2.1.2.2 *N1/N2 is a physical condition, a mental state or a mentally salient entity experienced by N2/N1.*

For example: *polio sufferer*, *cat instinct*, *student problem* (*problem which students have*), *union concern*.

Rule 2.1.2.3 *N1/N2 has the property denoted by N2/N1.*

For example: *water volume*, *human kindness*. A "property" is something that is not an entity or a substance but which an entity/substance can be described as having. *Redness*, *temperature*, *dignity*, *legibility* are all examples of properties. Property nouns are often derived from adjectives but this need not be the case.

Rule 2.1.2.4 *N1/N2 has N2/N1 as a part or constituent.*

For example: *car door*, *motor boat*, *cat fur*, *chicken curry*, *pie ingredient*, *tree sap*. The test for the presence of a part-whole relation is whether it seems natural and accurate in the context to say *The N1/N2 has/have N2/N1* and *The N1/N2 is/are part of N2/N1*. Furthermore, substances which play a functional role in a biological organism are classed as parts: *human blood*, *tree sap*, *whale blubber*. This is the case even when the substance has been extracted, as in *olive oil*.

A part is often located in its whole, but in these cases the part-whole relation is to be considered as prior to the co-location, and HAVE is preferred to IN. Complications arise with cases such as *sea chemical*, where both HAVE and IN seem acceptable. One principle that can be used tests whether the candidate part is readily separated (perceptually or physically) from the candidate whole. Chemicals in *sea water* (HAVE) are not typically separable in this way and can be viewed as parts of a whole. On the other hand, a *sea stone* or a *sea (oil) slick* are perceptually distinct and physically separable from the sea and are therefore IN.

Rule 2.1.2.5 *N1/N2 is a group/society/set/collection of entities N2/N1*

For example: *stamp collection*, *character set*, *lecture series*, *series lecture*, *committee member*, *infantry soldier*.

2.1.3 IN

In the following rules, an opposition is drawn between events/activities and objects. The class of events includes temporal entities such as times and durations. Objects are perceived as non-temporal and may be participants in an event (the term *participant* is used

as defined under Rule 1.7). To assign the correct rule, the annotator must decide whether the located thing is an event or an object, and whether the location is temporal or spatial. Events may also sometimes be participants (in the sense of Rule 1.7 and in these cases the rules dealing with objects and participants will apply – a *nursing college* is a college where nursing is taught as a subject, but not necessarily one where the activity of nursing takes place, so Rule 2.1.3.1 applies. In contrast a *nursing home*, being a home where the event of nursing takes place, would come under Rule 2.1.3.2, analogous to *dining room*. Some nouns are polysemous and can refer to both objects (*play* as a written work, *harvest* as harvested crops) and events (*play* as performance, *harvest* as activity). The annotator must decide whether the temporal or physical aspect is primary in a given context.

Rule 2.1.3.1 *N1/N2 is an object spatially located in or near N2/N1.*

For example: *forest hut*, *shoe box*, *side street*, *top player*, *crossword page*, *hospital doctor*, *sweet shop*. Where the location is due to part-whole constituency or possession, HAVE is preferred (as in *car door*, *sea salt*). Source-denoting compounds such as *country boy* and *spring water* are classed as IN as the underlying relation is one of location at a (past) point in time.

Rule 2.1.3.2 *N1/N2 is an event or activity spatially located in N2/N1.*

For example: *dining room*, *hospital visit*, *sea farming*, *football stadium*.

Rule 2.1.3.3 *N1/N2 is an object temporally located in or near N2/N1, or is a participant in an event/activity located there.*

For example: *night watchman*, *coffee morning*.

Rule 2.1.3.4 *N1/N2 is an event/activity temporally located in or near N2/N1.*

For example: *ballroom dancing*, *future event*, *midnight mass*.

2.1.4 ACTOR

The distinction between ACTOR and INST is based on sentience. Only certain classes of entities may be actors:

1. Sentient animate lifeforms: membership of the animal kingdom (*regnum animalia*) is a sufficient condition. Bacteria and viruses are not sentient enough (*flu virus* is annotated INST).

2. Organisations or groups of people: for example *finance committee*, *consultancy firm*, *manufacturing company*, *council employee*. Some words referring to institutions are polysemous in that they can denote its physical aspect or its social/organisational aspect – *university* often denotes a physical location, but in the compounds *university degree* and *university decision* it is functioning as an organisation and count as agents (granting a degree and making a decision are actions only humans or organisations can carry out). On the other hand, in *research university* it is not clear whether we have a *university that does research* (agentive) or a *university in which*

research is done (non-agentive). In such cases, the physical denotation should be considered the primary meaning of the word, and the organisational denotation is derived through metonymy – the non-agentive interpretation of these compounds is favoured unless the underlying event requires the institution to act as an agent. Such events often involve the institution acting as a legal entity. Hence *university degree* (*degree awarded by a university*, *school decision* (*decision made by a school*), *shop employee* (*employee employed by a shop*) are ACTOR; *research university*, *community school*, *school homework* and *sweet shop* are IN.

A compound can be labelled ACTOR only if the underlying semantic relation involves a characteristic situation or event. In the following definitions, the term *participant* is used in the sense of Rule 1.7.

Rule 2.1.4.1 *N1/N2 is a sentient participant in the event N2/N1.*

For example: *student demonstration*, *government interference*, *infantry assault*. That N2/N1 denote an event is not sufficient for this rule – it must be the characteristic event associated with the compound. Hence this rule would not apply to a *singing teacher*, as the characteristic event is teaching, not singing. Instead, Rule 2.1.4.2 would apply. As only one participant is mentioned in the current rule 2.1.4.1, there is no need to establish its degree of agentivity.

Rule 2.1.4.2 *N1/N2 is a sentient participant in an event in which N2/N1 is also a participant, and N1/N2 is more agentive than N2/N1.*

For example: *honey bee*, *bee honey*, *company president*, *history professor*, *taxi driver*, *student nominee* (*nominee nominated by students*), *expressionist poem*. Relative agentivity is determined by the hierarchy given under Rule 1.7. The underlying event cannot be one of possession (*car owner* = HAVE) or location (*city inhabitant* = IN). Profession-denoting compounds often have a modifier which is a location – *street cleaner*, *school principal*, *restaurant waitress*, *school teacher*. A distinction can be drawn between those where the profession involves managing or changing the state of the location, i.e. the location is an object (*school principal*, *street cleaner* = ACTOR), and those where the profession simply involves work located there (*school teacher*, *restaurant waitress* = IN by Rule 2.1.3.1). Note that modifiers in *-ist* such as *expressionist*, *modernist*, *socialist*, *atheist* are treated as nouns, so that an *expressionist poem* is analysed as a *poem such as an expressionist would characteristically write*.

2.1.5 INST

The name INST(rument) is used to distinguish this category from ACTOR, though the scope of the category is far broader than traditional definitions of instrumentality. Again, the term *participant* is used in the sense of Rule 1.7.

Rule 2.1.5.1 *N1/N2 is a participant in an activity or event N2/N1, and N1/N2 is not an ACTOR.*

For example: *skimming stone*, *gun attack*, *gas explosion*, *combustion engine*, *drug trafficking*, *rugby tactics*, *machine translation*. Compounds identifying the location of an

event (such as *street demonstration*) should be labelled IN by Rule 2.1.3.2 or 2.1.3.4, and compounds identifying the focus of or general motivation for a human activity or mental process (such as *crime investigation*), but not its direct cause, should be labelled ABOUT by Rule 2.1.6.3.

As only one participant is mentioned, there is no need to establish its degree of agentivity.

Rule 2.1.5.2 *The compound is associated with a characteristic event in which N1/N2 and N2/N1 are participants, N1/N2 is more agentive than N2/N1, and N1/N2 is not an ACTOR.*

For example: *rice cooker* (*cooker that cooks rice*), *tear gas* (*gas that causes tears*), *blaze victim* (*a blaze injures/kills a victim*). The directionality of the relation is determined by the more agentive participant in the hierarchy given in Rule 1.7: $cheese_O$ $knife_I$ (INST2), $wine_O$ $vinegar_R$ (INST1), $wind_A$ $damage_R$ (INST1), $human_O$ $virus_A$ (INST1). Sometimes it may be difficult to distinguish Agents from Instruments (*gun wound*) or Objects from Results (*blaze victim*) – this is not important so long as it is possible to identify which participant is more agentive.

In some cases, it may not be clear what the exact underlying event is, but the more agentive participant may still be identified – a *transport system* is a system that in some way provides or manages transport, but it is nonetheless clear that the appropriate label is INST2. In other cases, where both participants affect each other, it may be less clear which is more agentive – *motor oil* can be construed as *oil that lubricates/enables the function of the engine* or as *oil the engine uses*. Likewise *petrol motor*, *computer software*, *electron microscope*. At least where the relation is between a system or machine and some entity it uses to perform its function, the former should be chosen as more agentive. Hence *motor oil* is INST1, *petrol motor* is INST2, and so on.

As in Rule 2.1.5.1, where one of the constituents is the location of the associated event, then IN is the appropriate label by Rule 2.1.3.1 or 2.1.3.3. If the more agentive participant meets the criteria for ACTOR status (2.1.4), then that label should be applied instead. If the interaction between the constituents is due to one being a part of the other (as in *car engine*), HAVE is the appropriate label by Rule 2.1.2.4. A border with ABOUT must be drawn in the case of psychological states and human activities whose cause or focus is N1. As described further under Rules 2.1.6.3, the criterion adopted is based on whether there is a direct causal link between N1 and N2 in the underlying event – a bomb can by itself cause *bomb terror* (INST1), but a *spider phobia* is not a reaction to any particular spider and is classed as ABOUT.

2.1.6 ABOUT

Rule 2.1.6.1 *N1/N2's descriptive, significative or propositional content relates to N2/N1.*

For example: *fairy tale*, *flower picture*, *tax law*, *exclamation mark*, *film character*, *life principles*. Most speech acts belong to this category. Properties and attributes that seem to have a descriptive or subjective nature are still to be labelled HAVE by Rule 2.1.2.3 – *street name* and *music loudness* are HAVE1.

Rule 2.1.6.2 *N1/N2 is a collection of items whose descriptive, significative or propositional content relates to N2/N1 or an event that describes or conveys information about N2/N1.*

For example: *history exhibition*, *war archive*, *science lesson*.

Rule 2.1.6.3 *N1/N2 is a mental process or mental activity focused on N2/N1, or an activity resulting from such.*

For example: *crime investigation*, *science research*, *research topic*, *exercise obsession*, *election campaign*, *football violence*, *holiday plan*. In the case of activities, N1/N2 cannot belong to any of the participant categories given under Rule 1.7; rather it is the topic of or motivation for N2/N1. The sense of causation in, for example, *oil dispute* is not direct enough to admit an INST classification – the state of the oil supply will not lead to an *oil dispute* without the involved parties taking salient enabling action. In the case of emotions, there is also a risk of overlapping with INST; *bomb terror* is INST and *bomb dislike* is classed as ABOUT, but examples such as *bomb fear* are less clearcut. A line can be drawn whereby immediate emotional reactions to a stimulus are annotated INST, but more permanent dispositions are ABOUT. In the case of *bomb fear*, the relation must be identified from context. Problems (*debt problem*) and crises (*oil crisis*) also belong to this category, as they are created by mental processes.

Rule 2.1.6.4 *N1/N2 is an amount of money or some other commodity given in exchange for N2/N1 or to satisfy a debt arising from N2/N1.*

For example: *share price*, *printing charge*, *income tax*. N2/N1 is not the giver or recipient of N1/N2 – an *agency fee* would be INST under the interpretation fee_I *paid to an* $agency_O$, but the thing exchanged or the reason for the transaction.

2.1.7 REL

Rule 2.1.7.1 *The relation between N1 and N2 is not described by any of the above relations but seems to be produced by a productive pattern.*

A compound can be associated with a productive pattern if it displays substitutability. If both of the constituents can be replaced by an open or large set of other words to produce a compound encoding the same semantic relation, then a REL annotation is admissible. For example, the compound *reading skill* (in the sense of degree of skill at reading) is not covered by any of the foregoing categories, but the semantic relation of the compound (something like ABILITY-AT) is the same as that in *football skill*, *reading ability* and *learning capacity*. This contrasts with an idiosyncratic lexicalised compound such as *home secretary* (= LEX), where the only opportunities for substitution come from a restricted class and most substitutions with similar words will not yield the same semantic relation. Another class of compounds that should be labelled REL are names of chemical compounds such as *carbon dioxide* and *sodium carbonate*, as they are formed according to productive patterns. Proper names composed of two common nouns with no semantic connection also belong to this class (e.g. *Penguin Books*, see Rule 1.5).

2.1.8 LEX

Rule 2.1.8.1 *The meaning of the compound is not described by any of the above relations and it does not seem to be produced by a productive pattern.*

For example: *turf accountant*, *monkey business*. These are noncompositional in the sense that their meanings must be learned on a case-by-case basis and cannot be identified through knowledge of other compounds. This is because they do not have the property of substitutability - the hypothetical compounds *horse business* or *monkey activity* are unlikely to have a similar meaning to *monkey business*. LEX also applies where a single constituent has been idiosyncratically lexicalised as a modifier or head such as *X secretary* meaning *minister responsible for X*.

2.1.9 UNKNOWN

Rule 2.1.9.1 *The meaning of the compound is too unclear to classify.*

Some compounds are simply uninterpretable, even in context. This label should be avoided as much as possible but is sometimes unavoidable.

2.2 Noncompounds

2.2.1 MISTAG

Rule 2.2.1.1 *One or both of N1 and N2 have been mistagged and should not be counted as (a) common noun(s).*

For example: *fruity bouquet* (N1 is an adjective), *London town* (N1 is a proper noun). In the case of *blazing fire*, N1 is a verb, so this is also a case of mistagging; in superficially similar cases such as *dancing teacher* or *swimming pool*, however, the *-ing* form can and should be treated as a noun. The annotator must decide which analysis is correct in each case – a *dancing teacher* might be a *teacher who is dancing* (MISTAG) in one context, but a *teacher who teaches dancing* (ACTOR) in another context. Certain modifiers might be argued to be nouns but for the purposes of annotation are stipulated to be adjectives. Where one of *assistant*, *key*, *favourite*, *deputy*, *head*, *chief* or *fellow* appears as the modifier of a compound in the data, it is to be considered mistagged. This only applies when these modifiers are used in adjective-like senses – *key chain* or *head louse* are clearly valid compounds and should be annotated as such.

2.2.2 NONCOMPOUND

Rule 2.2.2.1 *The extracted sequence, while correctly tagged, is not a 2-noun compound.*

There are various reasons why two adjacent nouns may not constitute a compound:

1. An adjacent word should have been tagged as a noun, but was not.

2. The modifier is itself modified by an adjacent word, corresponding to a bracketing [[X N1] N2]. For example: [[*real tennis*] *club*], [[*Liberal Democrat*] *candidate*], [[*five dollar*] *bill*]. However compounds with conjoined modifiers such as *land and sea warfare* and *fruit and vegetable seller* can be treated as valid compounds so long as the conjunction is elliptical (*land and sea warfare* has the same meaning as *land warfare and sea warfare*). Not all conjoined modifiers satisfy this condition – a *salt and pepper beard* does not mean *a beard which is a salt beard and a pepper beard*, and the sequence *pepper beard* is a NONCOMPOUND.

3. The two words are adjacent for other reasons. For example: *the question politicians need to answer*, structureless lists of words.

4. The modifier is not found as a noun on its own. For example: *multiparty election*, *smalltown atmosphere*.