COMPOSITIONAL LEXICAL SEMANTICS IN NATURAL LANGUAGE INFERENCE

Ellie Pavlick

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2017

Supervisor of Dissertation

Chris Callison-Burch, Associate Professor of Computer and Information Science

Graduate Group Chairperson

Lyle Ungar, Professor of Computer and Information Science

Dissertation Committee

Chris Callison-Burch, Associate Professor of Computer and Information Science

Ido Dagan, Professor of Computer Science

Mitch Marcus, Professor of Computer and Information Science

Florian Schwarz, Associate Professor of Linguistics

Lyle Ungar, Professor of Computer and Information Science

COMPOSITIONAL LEXICAL SEMANTICS IN NATURAL LANGUAGE INFERENCE

Dedicated to my sisters, Cassie and Ginger.

Your creativity and sense of humor is why I love language.

ACKNOWLEDGEMENT

I am incredibly lucky to have had such a supportive advisor, Chris Callison-Burch. Chris invested in me very early in my computer science career and had confidence in me well before I had any idea what I was doing. He has been my go-to mentor and my strongest advocate since. With every year, I have only become more excited about the field, and I credit that to Chris's consistent energy and encouragement. It is been a genuine privilege to work together.

A huge thank you to my dissertation committee. Mitch Marcus has been a constant source of wisdom and guidance, and his personality, in my mind, will always be synonymous with Penn NLP and the wonderful intellectual environment I have experienced during my time here. Lyle Ungar's directness and pragmatism has been invaluable, and he has provided me with clarity and perspective when trying to articulate my own thoughts. Florian Schwarz gave me my first formal education in linguistics, and I am not exaggerating to say that that has had the single largest influence on the way I approach research. Annual meetings with Ido Dagan have been the most enlightening part of ACL conferences over the past several years–his drive is inspiring and every conversation I have had with him has reaffirmed my enthusiasm for the ambitiousness of NLP.

I have been fortunate to work with many other incredible research mentors. I have learned so much from Ben Van Durme, Joel Tetreault, Peter Clark, and Niranjan Balasubramanian and owe each of them a huge thank you. Thank you especially to Marius Pasca, for engaging in every discussion with such depth and for challenging me on every opinion–working together made me a holistically better researcher, and I am extremely grateful. Of course, thank you also to my many coauthors and other collaborators, who have been a constant source of new ideas and great conversations.

Finally, I am forever thankful to my family and friends who have been an unwavering support network. Thank you first to my incredible husband Steve, for being my best friend

and for somehow managing to both keep me inspired and keep me grounded over the years. Thank you to my parents, Karin and Michael, for endless love and encouragement and for raising me to be confident and ambitious and unapologetic. Thank you to my sisters, Cassie and Ginger–growing up with you two is the best thing that ever happened to me. Thank you to Andrea, Jan, Howard, Nate, and Sherrie for all the love and support. Last, but never least, thank you to all the friends I have made at Penn and within the ACL community, and to the friends from my other lives at JHU, Peabody, and elsewhere, for all the happiness you've brought me.

ABSTRACT

COMPOSITIONAL LEXICAL SEMANTICS IN NATURAL LANGUAGE INFERENCE

Ellie Pavlick

Chris Callison-Burch

The focus of this thesis is to incorporate linguistic theories of semantics into data-driven models for automatic natural language understanding. Most current models rely on an impoverished version of semantics which can be learned automatically from large volumes of unannotated text. However, many aspects of language understanding require deeper models of semantic meaning than those which can be easily derived from word co-occurrence alone. In this thesis, we inform our models using insights from linguistics, so that we can continue to take advantage of large-scale statistical models of language without compromising on depth and interpretability. We begin with a discussion of lexical entailment. We classify pairs of words according a small set of distinct entailment relations: e.g. equivalence, entailment, exclusion, and independence. We show that imposing these relations onto a large, automatically constructed lexical entailment resource leads to measurable improvements in an end-to-end inference task. We then turn our attention to compositional entailment, in particular, to modifier-noun composition. We show that inferences involving modifier-noun phrases (e.g. *"red dress"*, *"imaginary friend"*) are much more complex than the conventional wisdom states. In a systematic evaluation of a range of existing state-of-the-art natural language inference systems, we illustrate the inability of current technology to handle the types of common sense inferences necessary for human-like processing of modifier-noun phrases. We propose a data-driven method for operationalizing a formal semantics framework which assigns interpretable semantic representations to individual modifiers. We use our method in order to find instances of fine-grained classes involving multiple modifiers (e.g. *"1950s American jazz composers"*). We demonstrate that our proposed compositional model outperforms existing non-compositional approaches.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1 : Introduction

1.1. Overview

The focus of this thesis is to incorporate linguistic theories of semantics into data-driven models for automatic natural language understanding (NLU). Current approaches to NLU have tended toward shallow models of semantics which can be learned automatically from large volumes of unannotated text. By trading depth for speed and scalability, current techniques have enabled profound advances in language technology for many applications, most notably information retrieval, machine translation, and speech recognition. However, the majority of the models in widespread use today lack precise representations of meaning and of inferences in language. As we ask computers to perform progressively more complex natural language tasks, and to engage more interactively with humans, precision and interpretability of inference become increasingly necessary. Attempting to learn deep models of semantics naively from data–i.e. basing such models on statistical co-occurances alone–quickly encounters problems stemming from data sparsity, spurious correlations, and failure to recognize implicit "common sense" context. As a result, most current NLU models rely on an impoverished version of semantics in which two natural language expressions (whether words, phrases, sentences, or documents) are either "similar" or "dissimilar", but their relationship more specifically is not made clear. This severely limits the depth of natural language tasks that such models are able to perform.

In this thesis, we explore ways in which, by injecting insights about human language from formal and experimental linguistics, we can increase the inference capabilities of automatic NLU. We focus on *lexical semantics*, i.e. the meanings of and relationships between individual words, and on *compositional semantics*, i.e. how the meanings of individual words combine to produce the meanings of larger phrases. We devote most of our attention to single words and short (two-word) phrases, but are able to show that incorporating linguistic principles even at this low level has measurable impact on downstream tasks involving

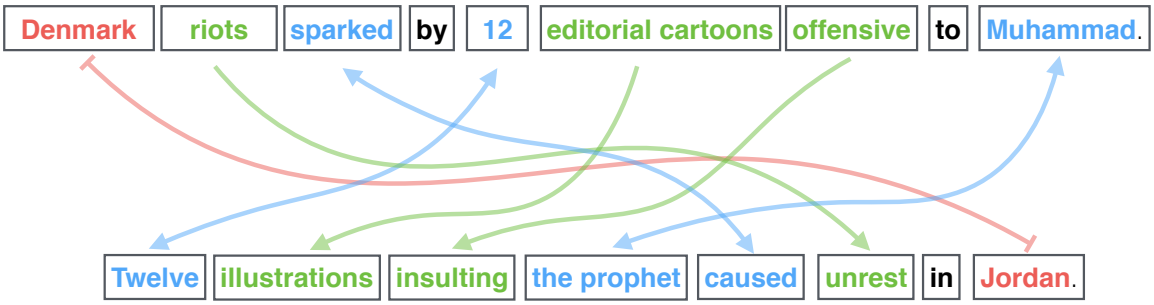


Figure 1: Example of a premise p and hypothesis h. Recognizing similarities and differences in sentence meaning requires understanding different types of relationships that can exist between words, such as equivalence (blue), entailment (green), and exclusion (red).

reasoning and inference about natural language.

Much of our work is motivated by the task of *recognizing textual entailment* (RTE), also know as *natural language inference*. In natural language processing (NLP), this task is defined simply as: given two natural language statements, a premise p and a hypothesis h, would a typical person reading p likely infer h to be true? If the answer is yes, we say that p *entails* h (Dagan et al. (2006)). This task of determining whether or not differences in wording correspond to differences in meaning is a core component of many important NLP applications such as summarization, question answering, and information extraction. Figure 1 shows an example of a premise p and a hypothesis h that illustrate the RTE task. In the example shown, p does not entail h. Recognizing this depends on knowing that *"Denmark"* and *"Jordan"* refer to mutually-exclusive locations, and that this prevents p from entailing h despite the fact that much of other information in the two sentences is the same, e.g. *"Muhammad"* is equivalent to *"the prophet"* and *"riots"* entail *"unrest"*.

The area of linguistics and NLP concerned with the relationships that hold between individual words–whether they entail (*"riots"/"unrest"*) or contradict (*"Denmark"/"Jordan"*) one another–is known as *lexical entailment*. Early computational work in this area focused on the manual construction of lexical entailment resources such as WordNet (Fellbaum (1998)), and FrameNet (Baker et al. (1998)). These resources explicitly specify the se-

mantic relationships between words and are intended to be used by automatic systems in NLU tasks. For example, WordNet organizes nouns into an ontology which clearly specifies whether words are related by synonymy (e.g. *"the prophet"*/"Mohammad"), hypernymy (*"riots"*/*"unrest"*), or antonymy (*"violence"*/*"peace"*).

Equivalent	Entailment	Exclusion	Unrelated
look at/watch	little girl/girl	close/open	girl/play
a person/someone	kuwait/country	minimal/significant	found/party
clean/cleanse	tower/building	boy/young girl	profit/year
away/out	the cia/agency	nobody/someone	man/talk
distant/remote	sneaker/footwear	blue/green	car/family
phone/telephone	heroin/drug	france/germany	holiday/series

Table 1: Examples of different types of entailment relations appearing in the Paraphrase Database (Ganitkevitch et al. (2013)).

More recently, work in lexical entailment has turned toward trying to build lexical entailment resources automatically from data. Automatically constructed resources are less expensive and less time-consuming to build than are manually-constructed ones, making them easier to adapt to new languages or domains. However, resources extracted using automatic processes lack the precise information available in hand-built ontologies. That is, rather than specifying that a *"riot"* is a type of *"unrest"*, or that *"peace"* is the opposite of *"violence"*, most automatically-built lexical entailment resources provide only a list of pairs of words which are believed, with some degree of confidence, to have "similar" or "related" meanings. The precise nature of that relation, however, is not known, and it may be different for each pair in the resource. For example, in the Paraphrase Database (Ganitkevitch et al. (2013)), words or phrases are considered to be "paraphrases" if they share a translation in some foreign language (Section 2.2.5). This method extracts pairs with equivalent meanings, but also pairs related by one-directional entailment (*"tower"*/*"building"*) and pairs that have opposite meanings (*"close"*/*"open"*) (Table 1).

In order for automatically-constructed resources like the Paraphrase Database (PPDB) to be applied in more nuanced NLU settings, like the RTE example shown Figure 1, they need a clearer definition of what it means for words to be "similar". In this thesis (Chapter

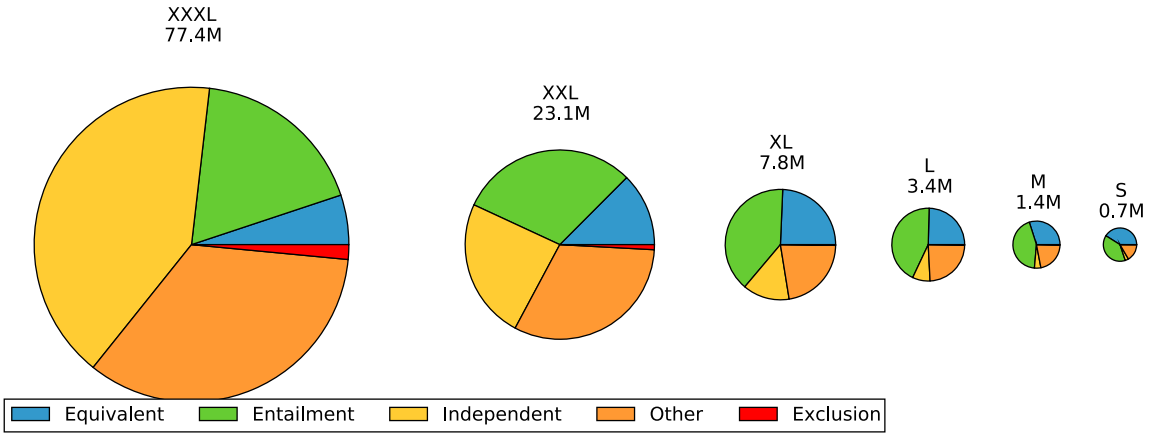


Figure 2: Estimated distribution of fine-grained entailment relations across the six different sizes of the Paraphrase Database (PPDB). Smaller sizes have been filtered to contain higher precision sets of paraphrases, yet they still contain a multitude of distinct fine-grained entailment relations.

3), we add explicit and interpretable lexical entailment relations, like those available in WordNet, to each of the paraphrase pairs in PPDB. To do this, we build a statistical classifier which, given a pair of English words or phrases, predicts which entailment relation holds. Unlike prior work on building lexical entailment resources from data, our classifier is able to combine multiple signals from varying types of corpora, enabling us to achieve high precision even when distinguishing between subtle differences in semantic relations. Using the classifier, we assign entailment relations to every paraphrase pair in PPDB, revealing that a multitude of fine-grained entailment relations exist in the database (Figure 2). We show that, using our automatically-annotated version of PPDB, a state-of-the-art RTE system achieves measurable performance gains over what is achieved using either WordNet or the original, unannotated version of PPDB .

In general, the study of lexical entailment is simplified by the fact that single words (and some short phrases) can be treated as non-compositional. That is, we can treat the words and phrases as atomic symbols that represent single units of meaning. As a result, we can reason about the meaning associated with each word or phrase by, for example, observing the contexts in which it is used, or looking at the ways in which it is

translated. While this method makes sense for single words (*"girl"*, *"cartoon"*), it makes less sense for phrases which could be clearly decomposed into smaller units of meaning (*"little girl"* $\rightarrow$ *"little"* + *"girl"*, *"editorial cartoon"* $\rightarrow$ *"editorial"* + *"cartoon"*). Modeling language compositionally is important for a number of reasons. First, non-compositional models are computationally inefficient: for a vocabulary of size N there are N^k k-word phrases, meaning that modeling the semantics of all short sentences would become intractable for even a small vocabulary. Perhaps more problematic is the fact that, in many cases, non-compositional models are incapable of learning the meaning of a phrase altogether. There are an infinite number of possible things that can be said in natural language, and the probability that any specific expression we are interested in understanding has been said before is very low. The probability that it has been said frequently enough for us to robustly estimate, for example, its "typical" usage in a distributional sense is near zero. Thus, compositional models of language are essential for advanced automatic natural language understanding.

In this thesis, we focus our study of composition on one particular type: the combination of modifiers (e.g. *"red"*, *"little"*, *"editorial"*) with common nouns (e.g. *"dress"*, *"girl"*, *"cartoon"*). Despite its apparent simplicity, modifier-noun composition proves to be highly complex. Traditionally, in formal semantics, modifiers are classified based on the effect the modifier has on the *denotation* of the noun it modifies (Figure 3). For example, *subsective modifiers* like *"red"* lead to inferences of one-directional entailment: every *"red dress"* is a *"dress"* but not every *"dress"* is a *"red dress"*. In contrast, *privative modifiers* like *"fake"* lead to inferences of contradiction: a *"fake gun"* is not in fact a *"gun"*. *Plain non-subsective* modifiers like *"alleged"* do not permit inferences of entailment or of contradiction: not every *"alleged thief"* is a *"thief"*, but some are. In previous work on RTE and NLU, these classes have largely governed the way systems reason about modifiers. For example, when the hypothesis contains a modifier that is not in the premise, systems usually infer that p does not entail h: e.g. *"She is wearing a dress"* does not entail *"She is wearing a red dress"*.

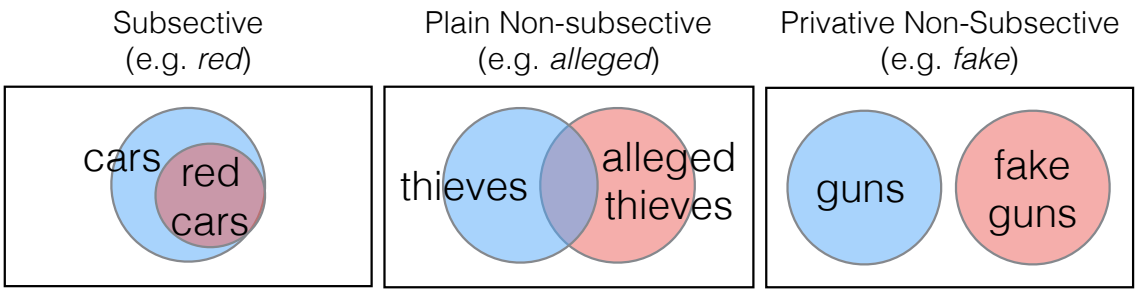


Figure 3: Classes of modifiers based on the entailment relationship between the denotation of the noun and that of the modified noun, as traditionally defined in formal semantics.

We show, however, that the entailment behavior of modifier-noun composition is much more complex than these classes suggest and than NLU systems frequently assume (Chapter 4). Table 2 provides examples of the types of inferences that humans make in practice. As shown, humans often accept the addition of modifiers that are not explicitly stated but nonetheless can be inferred from context, and reject the addition of modifiers that are not explicitly precluded but nonetheless deemed unlikely given the context. For example, even though not every *"crowd"* is an *"enthusiastic crowd"*, when given the sentence *"The crowd roared"*, humans easily infer that *"crowd"* entails *"enthusiastic crowd"*.

Sentence Pair	Human Consensus
p: The **crowd** roared. h: The **enthusiastic crowd** roared.	p entails h
p: Some went for the history and **political culture**. h: Some went for the history and **culture**.	p doesn't necessarily entail h
p: Bush spoke in Michigan on the **economy**. h: Bush spoke in Michigan on the **Japanese economy**.	p contradicts h

Table 2: Examples of contexts in which human inferences do not match what is expected given the classes of modifiers defined by formal semantics.

These types of human inferences appear to be governed not by rigorous logical inference, but rather by pragmatic inferences about what is most likely the case based on common knowledge of the events that tend to happen and the way people tend to speak about those events. We illustrate that this type of "common sense" reasoning is especially challenging for automatic systems. We construct a simplified RTE task in which p and h differ only

by a single modifier, like the examples given in Table 2, and evaluate the performance of a variety of state-of-the-art RTE systems. We show that none of the systems tested achieve significantly better than chance performance on this task. Our results demonstrate that even for a linguistic phenomenon as seemingly simple as modifier-noun composition, current statistical models fail to mimic human inferences.

One limitation of current treatments of modifier-noun composition in NLP is that they concentrate exclusively on the *semantic containment* aspect of meaning. That is, they focus exclusively on the fact that the set of *"enthusiastic crowds"* is a proper subset of the set of all *"crowds"*. This inference, while useful, neglects the fact that *"enthusiastic crowds"* are not just a random subset of *"crowds"*, but in fact have a specific set of properties that qualify them as *"enthusiastic"*. In other words, most current approaches neglect to model the *intrinsic meaning* of the modifier itself. As a result, even though a crowd that roars is by definition *"enthusiastic"*, current approaches can infer only that not *every "crowd"* is an *"enthusiastic crowd"* and so fail to infer that *"crowd"* entails *"enthusiastic crowd"* even when the context unambiguously permits the inference.

Thus, the taxonomic approach to modeling modifier-noun phrases–i.e. focusing only on the entailment relationship between the noun and the modified noun rather than on the meaning of the modifier itself–limits the capability of NLU systems. This limitation is especially visible in the task of *class-instance identification*: recognizing whether an entity (e.g. *"Charles Mingus"*) belongs to a particular class (e.g. *"jazz composers"*). Because current state-of-the-art approaches do not model the meaning of the modifiers themselves, they often require that the entity be explicitly named as member of the class in order to recognize that the instance-of relation holds. For example, in order to recognize that *"Charles Mingus"* is a *"jazz composer"*, many current methods require that a phrase like *"Charles Mingus is a jazz composer"* appears somewhere in a text corpus (Hearst (1992); Snow et al. (2004); Shwartz et al. (2016)). This technique works for coarse-grained classes like *"jazz composer"*, but it fails when trying to handle more fine-grained classes like *"1950s*

American jazz composer", for which the entire class label is unlikely to appear in even a very large text corpus. For such fine-grained classes, class-instance relations are often still inferable from text, but making the inferences requires compositional NLU models capable of reasoning individually about each of the modifiers and about the properties that they entail (Figure 4).

1950s American jazz composers
...seminal **composers such as** Charles Mingus and George Russell...
...virtuoso bassist and composer, Mingus **irrevocably changed the face of jazz**...
...Mingus **truly was a product of America** in all its historic complexities...
...Mingus **dominated the scene back in the 1950s** and 1960s...

Figure 4: Illustration of how the information required to infer Instance-Of relations for fine-grained classes is often distributed across multiple sentences, and thus requires compositional NLU models capable of reasoning individually about each of the modifiers.

As part of this thesis (Chapter 5), we operationalize a formal semantics model of modifier-noun composition in which the intrinsic meaning of a modifier (e.g. *"1950s"*) corresponds to the properties which differentiate instances of the modified-noun (*"1950s composers"*) from instances of the noun (*"composers"*) more generally. We model the meaning of a modifier as a list of properties, stated in natural language, which characterize the individual entities for which that modifier holds. These properties are acquired by paraphrasing noun phrases so that the usually-implicit relationship between the modifier and the head is made explicit (*"1950s composers"*$\rightarrow$ *"composers active in the 1950s"*). We evaluate our method on its ability to find all instances of a class given only the class name and a corpus of raw web text. Our compositional model is able to achieve significant improvements over the state-of-the-art non-compositional approaches on this task.

1.2. Outline of this Document

Chapter 2 gives an overview of the background and prior work relevant to this thesis. This includes a discussion of how "entailment" has been formalized in both linguistics and in NLP (Section 2.1), an introduction to how linguists and computer scientists have attempted to characterize the relationships between individual words (Section 2.2), and an overview

of work on composing word meanings to derive phrase meanings, specifically in the case of combining modifiers with nouns (Section 2.3). Chapter 2 also includes definitions of the 5 basic entailment relations to which we refer throughout this thesis (Section 2.4).

Chapter 3 presents our work on assigning fine-grained lexical entailment relations to pairs of natural language strings from the Paraphrase Database (PPDB). We describe our work on building a gold-standard dataset of phrase pairs labeled according to our basic entailment relations (Section 3.1). We use these labeled pairs to train a statistical classifier (Section 3.2), with which we transform PPDB into a fine-grained lexical entailment resource. We evaluate the quality of our classifier intrinsically (Section 3.3) and demonstrate that this automatically-annotated PPDB enables better performance for an end-to-end RTE system (Section 3.4). This chapter covers work previously published in Pavlick et al. (2015a).

Chapter 4 begins our discussion of composition by looking at how modifier-noun composition affects inferences about semantic containment: how we decide whether a modifier-noun phrase (e.g. *"red dress"*) entails or is entailed by the noun alone (*"dress"*). The chapter outlines our methodology for annotating and discussing modifier-noun composition in context (Section 4.1) and describes the human-labeled dataset we build to analyze modifier-noun composition in practice (Section 4.2). We analyze human inferences involving modifier-noun composition (Section 4.3 and 4.4) and show that a wide range of state-of-the-art RTE systems fail to learn human-like inferences in this seemingly simple setting (Section 4.5). This chapter covers work previously published in Pavlick and Callison-Burch (2016a) and Pavlick and Callison-Burch (2016b).

Chapter 5 focuses on how we can assign semantics to modifiers themselves in a meaningful and applicable way. We build our model directly on the formalization of modifier-noun composition given by formal semantics (Section 5.1). We assign meanings to modifiers using techniques borrowed from noun-phrase paraphrasing (Sections 5.2), which we apply to the task of class-instance identification, i.e. recognizing whether a given entity (e.g. *"Charles Mingus"*) is an instance of a given class (*"jazz composer"*) (Section 5.3). We show

that our compositional approach to class-instance identification outperforms state-of-the-art non-compositional baselines (Section 5.4). This chapter covers work previously published in Pavlick and Pasca (2017).

Chapter 6 concludes with a summary of Chapters 3 through 5 and a general discussion of directions for future work.

CHAPTER 2 : Background and Related Work

2.1. Definition of "Entailment" in Natural Language

Natural language inference, broadly construed, is the process by which we draw conclusions about what is true or false based on information expressed in natural language. Enabling computers to perform natural language inference is a lofty goal, which depends on advances not only in natural language processing, but in many areas of artificial intelligence, including knowledge representation, reasoning, and learning.

The study of "entailment" provides a means by which to isolate the language component of natural language inference, so that one can reason about how natural language expressions relate to one another–e.g. whether statements reaffirm or contradict each other–even without knowledge of whether those statements are actually true or false. According to Richard Montague, characterizing entailment is the "basic aim" of semantics (Janssen (2012)). This sentiment has been echoed in NLP, albeit is weaker terms, through the observation that "many natural language processing applications, such as Question Answering (QA), Information Extraction (IE), (multi-document) summarization, and machine translation (MT) evaluation, need ... to recognize that a particular target meaning can be inferred from different text variants" (Dagan et al. (2006)). As a result, defining and recognizing entailment has received substantial attention, at varying levels of rigor, from linguists, logicians, and computer scientists. In this section, I will give an overview of the most relevant prior work, which influences the ideas and models presented throughout this thesis.

2.1.1. Entailment in Formal Linguistics

Formal semantics aims to understand natural language using tools traditionally applied to formal languages, like mathematics or logic. Central to this analysis is *model theoretic semantics*. Model theory was originally designed for formal languages, not for human language, with the primary goal of characterizing the notion of true and false sentences. Model

theory creates a distinction between the actual world and the abstract symbols we use to talk about the world (that is, language). The "meaning" of a sentence in a language is defined as the conditions in the world that would have to hold in order for the sentence to be true. Under this definition, the meaning of a sentence is then independent of whether or not those conditions hold in actuality.

For example, in mathematics, we might have a sentence S which is: "$x > 17$". Whether S is true depends on our interpretations of the symbols in the sentence, specifically, our interpretations of "x" and "$>$". We refer to $\mathcal{I}$, an *interpretation* or *model of the world*, which fills in concrete values for the symbols in our language. For example, we can have one interpretation $\mathcal{I}$ in which "x" is equal to 29 and the relation "$>$" is the usual "greater than" relation. We could also define a different interpretation, $\mathcal{I}'$, in which "x" is equal to 14 and "$>$" is the same as in $\mathcal{I}$. In the first interpretation, the sentence S would be true, and so we can say $\mathcal{I}$ *models* S, or $\mathcal{I} \models S$. In $\mathcal{I}'$, however, S is false, so $\mathcal{I}' \not\models S$. The "meaning" of S, however, is unaffected by the choice of $\mathcal{I}$.

Symbols in natural language (i.e. words) can be treated similarly. For example, we can define the meaning of the word *"cat"* as a function that takes as input an entity and returns true if that entity is a cat and false otherwise, as shown below. (We go into greater depth on the formalization of words as functions in Section 2.2.1.)

$$[\![\text{“}cat''\!]\!](x) = \begin{bmatrix} 1 & \text{if } x \text{ is a cat} \\ 0 & \text{otherwise} \end{bmatrix}$$

It is possible to have an interpretation of the world in which $[\![\text{“}cat''\!]\!](Eddy) = 1$ and $[\![\text{“}cat''\!]\!](Lulu) = 0$ and another in which $[\![\text{“}cat''\!]\!](Eddy) = 0$ and $[\![\text{“}cat''\!]\!](Lulu) = 1$. In either case, the meaning of *"cat"* is still a function that takes as input an entity and returns a boolean corresponding to whether or not that entity is a cat.

Under the above formalization, determining whether a given sentence (e.g. *"Eddy is a cat"*) is true or false requires more than just knowledge of symbols and grammar of natural

language; it requires specific interpretation of the world. In contrast, reasoning about *entailment* means reasoning about truth conditions of natural language expressions and sentences, independently of any specific interpretation of the world. In formal semantics and logic, entailment can be defined as follows. Let p and h be sentences and $\mathcal{I}$ be an interpretation of the world; then

$$p \text{ entails } h \Leftrightarrow \forall \mathcal{I}((\mathcal{I} \models p) \Rightarrow (\mathcal{I} \models h))$$

That is, in any interpretation of the world in which p is true, h is also true (Janssen (2012)). For example, we can say that *"Eddy is a cat"* entails *"Eddy is an animal"* because in any world where $[\![\text{“}cat''\!]\!](Eddy) = 1$, it must be the case that $[\![\text{“}animal''\!]\!](Eddy) = 1$ as well. This is dependent on the relationship between the definitions of $[\![\text{“}cat''\!]\!]$ and $[\![\text{“}animal''\!]\!]$, not on the particular state of the world.

2.1.2. Entailment in Natural Language Processing

In the field of natural language processing (NLP), the concept of entailment has also received considerable research attention. However, NLP researchers have adopted a much less rigorous definition for what it means for a sentence p to entail a sentence h. The task of recognizing textual entailment (RTE) was originally defined by Dagan et al. (2006) as follows:

> ... [O]ur applied notion of textual entailment is defined as a directional relationship between pairs of text expressions, denoted by p- the entailing Text, and h- the entailed Hypothesis. We say that p entails h if, typically, a human reading p would infer that h is most likely true. This somewhat informal definition is based on (and assumes) common human understanding of language as well as common background knowledge.[1]

[1]The original quote uses T and H for "text" and "hypothesis". I have replaced these with the symbols p and h for consistency with the notation used throughout the rest of this dissertation.

That is, an automatic RTE system receives as input a pair of sentences, p and h, and returns ENTAILMENT if a typical human would infer h from p and NON-ENTAILMENT otherwise. Often, rather than treating the problem as a binary classification, systems treat it as a three-way classification, in which a system should return ENTAILMENT if p entails h, CONTRADICTION if p directly contradicts h, and UNKNOWN if p and h are compatible but p does not necessarily entail h.

This working definition differs substantially from that used in formal semantics. Specifically, in the definition used in NLP, entailment is defined probabilistically: h is entailed by p if, given p, a "typical" human would infer that h is "most likely" true. This is in sharp contrast to the definition given in Section 2.1.1, in which h is entailed by p if there is *no possible world* in which p is true and h is not. As a consequence, while the formal semantics notion of entailment is invariant to any particular state of the world–i.e. if p entails h in one world, p entails h in any world–the NLP notion does not make this guarantee. In practice, in nearly all NLP applications and datasets, judgments of whether p entails h are dependent on one particular interpretation of the world: namely, the "real" world. For example, under the NLP definition, the premise *"My dad is the richest man in the world"* would entail the hypothesis *"My dad is Bill Gates"*, while this entailment would not hold under the formal linguistics definition.

2.1.3. Types of Knowledge Tested by RTE

Popular Datasets

Because the NLP notion of entailment is intentionally informal and underspecified, it leaves ample room for variation across datasets as what constitutes "common human understanding of language" and "common background knowledge." Below, I outline some of the most significant and influential datasets for RTE, and describe the assumptions each makes about the task. The lack of standardization makes it difficult, or impossible, to compare the performance of different RTE systems which have been designed for different datasets, as dif-

ferent technological contributions often advance performance on one dataset at the expense of performance on others. Table 3 shows examples from the three datasets described.

FraCas	p	The people who were at the meeting voted for a new chairman.
	h	Everyone at the meeting voted for a new chairman.
RTE2	p	Actual statistics about the deterrent value of capital punishment are not available because it is impossible to know who may have been deterred from committing a crime.
	h	Capital punishment is a deterrent to crime.
SICK	p	A talk about an adult and a boy is given in the amphitheater.
	h	An adult is in the amphitheater and is talking to a boy.
SNLI	p	A woman in costume is marching with a large drum.
	h	She plays in a band.

Table 3: Example sentence pairs from four standard RTE datasets.

The FraCas Test Suite. One of the first RTE data sets was the FraCas test suite (Cooper et al. (1996)), which consists of only a few hundred problems, each consisting of a short sentence or group of sentences, followed by a hypothesis. The problems are designed to test specific linguistic phenomena, such as quantification, negation, and ellipsis. The dataset was especially popular for logic-oriented RTE systems, prior to the advent of statistical NLP methods, although it has been used more recently for systems which use a hybrid of logical and statistcal approaches (Lewis and Steedman (2013); Mineshima et al. (2015)). The FraCas suite was the primary dataset used in the evaluation of MacCartney and Manning (2008)'s NatLog system, which we discuss in detail in Section 2.2.3 and reference frequently thereafter.

The PASCAL Challenges. Dagan et al. (2006)'s initial definition of the RTE task, as quoted above, was accompanied by a series standardized RTE shared tasks, known as the PASCAL RTE Challenges. These shared tasks and the accompanying datasets have collectively been one of the biggest drivers of RTE system development. The datasets, numbered RTE1 through RTE8, are mostly derived from news articles and headlines. As a result, the premise/hypothesis pairs contain few linguistic "tricks" of the type tested by

FraCas. Instead, most inferences in the RTE datasets hinge on information from the text itself (e.g. words, dependencies, etc.) or on encyclopedic knowledge (e.g. named entities, locations, etc). The premise is usually substantially longer than the hypothesis, and often contains much more information than what is necessary to infer the hypothesis.

Datasets Aimed at Distributional Semantics Models. At the time of writing, arguably the most popular RTE dataset is the SICK dataset (Marelli et al. (2014a)), which stands for "Sentences Involving Compositional Knowledge". SICK was developed for the SemEval 2014 shared task, and aimed at evaluating the increasingly-popular distributional approaches to the RTE task. The sentences in the SICK data are mostly derived from crowdsourced descriptions of images, and thus cover a small vocabulary consisting of fairly concrete language. The inferences in SICK are intentionally designed not to require encyclopedic knowledge, temporal knowledge, or knowledge of multiword expressions, as is often the case for inferences in the older RTE data sets. Bowman et al. (2015) has recently released a much larger dataset, designed similarly through crowdsourcing and based on image captions. This dataset, called the Stanford Natural Language Inference (SNLI) corpus, consists of 570,000 sentence pairs, and is intended for training RTE systems based on deep neural networks (Section 2.1.4).

Common Sense Inferences in RTE Datasets

While the RTE task has always been framed as a discrete classification task–systems must label each p/h pair as one of ENTAILMENT, CONTRADICTION, or UNKNOWN–the definition of the task itself does not require that these judgements be objective. While earlier datasets (i.e. FraCas) defined entailment in the formal semantics sense, newer datasets, especially those built through crowdsourcing, have appealed more often to "common sense" knowledge and subjective inferences. As a result, more and more of the inferences tested in the RTE task depend on a much looser notion of entailment governed by context and assumptions about what is "probably" the case. For example, the SICK dataset requires systems to

infer that if *"two dogs are running along a beach"*, then the dogs must be *"playing"*, an inference which is certainly sensible, but is in no way logically entailed. Several examples of "common sense" inferences occurring in RTE datasets are shown in Table 4.

SICK	ENTAIL.	p	A couple of white dogs are running along a beach.
		h	Two dogs are playing on the beach.
RTE2	ENTAIL.	p	About one million years ago, people began to leave Africa.
		h	Humans existed 10,000 years ago.
SNLI	ENTAIL.	p	People listening to a choir in a Catholic church.
		h	Choir singing in mass.
SNLI	CONTRA.	p	High fashion ladies wait outside a tram.
		h	The women do not care what clothes they wear.

Table 4: Examples of sentence pairs occurring in RTE datasets which rely on common sense inference and world knowledge rather than logical inference.

2.1.4. RTE Systems and Approaches

There have been a wide range of approaches to solving the RTE task, which demonstrate different strengths and weaknesses, and for which the relative performance varies substantially based on the dataset. I give a high-level review of several different approaches to RTE below. It is important to note that each of the categories discussed represents a broad set of methods used in building RTE systems and they are not mutually-exclusive. Many modern RTE systems use two or more approaches simultaneously.

Logical Deduction

The earliest approaches to automatic natural language understanding sought to treat natural language like one would a formal language like logic or mathematics. In order to solve RTE, such systems attempt to parse both the premise and the hypothesis into an unambiguous logical representation, such as first order logic or λ-calculus, which is then fed into a logical deduction system (Harabagiu et al. (2000); Akhmatova (2005); Bayer et al. (2005); Fowler et al. (2005); Raina et al. (2005); Bos and Markert (2006); Mineshima et al. (2015)). The deduction system, which is sometimes strict and sometimes probabilistic, then searches for a proof or logical contradiction of h starting from p and any provided "back-

ground knowledge" or "axioms". While the purely proof-based approaches are less common for current RTE systems, the approach has not been abandoned altogether. Modern systems which used proof-based approaches often blend logical inference with shallower but more robust statistical methods (Lewis and Steedman (2013); Bjerva et al. (2014)).

Transformation-Based Approaches

A more robust approach seeks to find an interpretable "proof" of an entailing or contradictory relationship between p and h, but not in the form of a logical proof. RTE systems taking this approach can be thought of as "transformation-based" or "alignment-based." Transformation-based systems operate on the assumption that, when p entails h, everything stated in the hypothesis should correspond to some part of the premise (Haghighi et al. (2005); Kouylekov and Magnini (2005); Chambers et al. (2007); MacCartney and Manning (2008); Das and Smith (2009); Wang and Manning (2010); Heilman and Smith (2010); Stern and Dagan (2012)). Performing an alignment between p and h lends some interpretability to the system's prediction, for example by allowing a system to point to specific span in the hypothesis that could not be aligned, without using a fully logical approach. Most systems combine alignment with some amount of machine learning in order to learn the importance of each alignment (or lack of alignment) from data.

Natural Logic

Natural logic is a proof framework originally described by Lakoff (1972) and further developed in Sánchez Valencia (1991). The primary focus of the theory of natural logic has been on providing a system of deduction for natural language which operates on human-readable natural language strings themselves, as opposed to using formal logical representations. Such an approach is appealing from the point of view of computational NLU, as it requires very little processing in order to make inferences about natural language. As a result, several attempts have been made to incorporate the theory of natural logic into computational systems (Harabagiu and Hickl (2006); Bar-Haim et al. (2007); MacCartney and Manning

(2008); Schubert et al. (2010); Angeli and Manning (2014)). The best known RTE system based on natural logic is NatLog (MacCartney and Manning (2008)). At a high-level, NatLog is a transformation-based system which finds a sequence of "atomic edits"–shallow string operations including insertions, deletions, and substitutions–that can be used to transform the premise p into the hypothesis h. NatLog then works through each edit in the sequence, incrementally processing how each edit affects the entailment relationship between p and h and ultimately producing an overall sentence-level entailment prediction. We revisit several core components of natural logic later in the thesis: specifically, I describe basic entailment relations in Section 2.2.3 and atomic edits in Section 2.3.2.

Feature-Based Classification

Over the course of the RTE shared tasks, there was a shift away from logical and deduction-based approaches and toward machine learning systems which operate on featurized representations of the premise and hypothesis. Bentivogli et al. (2010) observe that the sixth RTE challenge marked the first year when none of the submitted systems took a logic-oriented approach to the task. Heavily feature-engineered RTE systems typically combine a mix of features capturing syntactic and semantic overlap between p and h, usually incorporating external resources such as WordNet (Fellbaum (1998)) and VerbOcean (Chklovski and Pantel (2004)) and linguistic analysis tools such as POS taggers and named-entity recognizers; Bentivogli et al. (2010) and Bentivogli et al. (2011) provide an analysis of features used by systems submitted to the RTE-6 and RTE-7 challenges, respectively. In more recent years, systems have incorporated more distributional features, including topic models and vector-space similarity of words and sentences; Marelli et al. (2014a) provides an analysis of features used in systems submitted to the SemEval 2014 task, in which the dataset (SICK, described above) was intended to evaluate distributional semantics models.

Deep Learning

With the advent of deep neural architectures, most current systems take an end-to-end approach to reasoning about language. In these models, both the premise and the hypothesis are represented as vectors, and their entailment relationship is a function of the relative position of the two sentences in vector space (Bowman et al. (2015); Wang and Jiang (2016)). Attention-based models, which are intended to capture the intuition of alignment between p and h, have been especially popular (Rocktäschel et al. (2016); Parikh et al. (2016)). More complex architectures have also been proposed which incorporate recursive structure of the sentences (Socher et al. (2012); Mou et al. (2016)). It is worth noting that while these architectures have demonstrated strong performance, evaluation has been carried out almost exclusively on the SICK and SNLI datasets, and there has been little evidence to suggest they capture the type of compositional or world knowledge tested by other datasets like the FraCas test suite or the PASCAL challenge sets.

2.2. Lexical Entailment

The previous section introduced the notion of entailment between two natural language sentences: what does it mean for a sentence p to entail or contradict a sentence h? In this section, I discuss the notion of entailment at the word level: what does it mean for a word to entail or contradict another word? I describe several existing NLP resources and methods designed to model lexical entailment relations.

2.2.1. Word Denotations and Semantic Types

Section 2.1.1 defined *denotations* and *interpretations*, as commonly used in linguistic theory. This formal semantics framework assumes two primitive types: truth values (type t) and entities (type e). A truth value $t \in \{0, 1\}$ can be either true (1) or false (0). An entity $e \in \mathcal{U}$ can refer to any thing in the universe $\mathcal{U}$ that might be referenced or named (e.g. *"John"*, *"your mom"*, *"that coffee mug"*). An arbitrary natural language expression can then be

modeled as a function that operates on these primitive types.

Throughout this section, $[\![\text{“}w\text{”}]\!]$ designates the meaning of the word w. $[\![\cdot]\!]$ represents the *interpretation function* which returns the denotation of a linguistic expression with respect some model of the world. For example, as in Section 2.1.1, the meaning of a common noun (e.g. *“cat”*) is a function that takes as input an entity and returns true if that entity is a cat and false otherwise:

$$[\![\text{“}cat\text{”}]\!](x) = \begin{bmatrix} 1 & \text{if } x \text{ is a cat} \\ 0 & \text{otherwise} \end{bmatrix}$$

I.e. $[\![\text{“}cat\text{”}]\!]$ is the characteristic function of a set of entities in $\mathcal{U}$. Notationally, we say nouns are functions of type $< e, t >$ meaning they take inputs of type e and return values of type t. For simplicity throughout this thesis, rather than writing out characteristic functions in this form, I will often use set notation as below. Note that these two representations are equivalent for our purposes, and may be used interchangeably, as long as it is clear from the context.

$$[\![\text{“}cat\text{”}]\!] = \{x \in \mathcal{U} \mid x \text{ is a cat}\}$$

Not all words define sets, or can be written as functions of type $< e, t >$. For example, simple transitive verbs can be formalized as functions of type $< e, < e, t >>$: e.g. *“loves”* takes as input a entity x, and returns the characteristic function of the set of all entities in $\mathcal{U}$ who x loves:

$$[\![\text{“}loves\text{”}]\!](x) = \{y \in \mathcal{U} \mid x \text{ loves } y\}$$

Subsective modifiers (e.g. *“red”*, *“good”*, *“tiny”*) are of type $<< e, t >, < e, t >>$, meaning they take as input a set and return a new set. E.g. *“tiny”* can take as input the set of

"cats" and return the set of *"tiny cats"*.

$$[\![\text{“}tiny''\text{”}]\!](\text{“}cat''\text{”}) = \{x \in \mathcal{U} \mid [\![\text{“}cat''\text{”}]\!](x) = 1 \wedge x \text{ is tiny}\}$$

We will discuss the above formalization of modifiers as functions in more depth in Section 2.3 and in Chapters 4 and Chapter 5.

2.2.2. Definition of Semantic Containment

Given a formalization of individual words as functions, it is necessary to revisit the meaning of entailment. Recall that the definitions of entailment given in Section 2.1.1 can only be applied to declarative sentences that can be assigned truth values (i.e. are of type t). Words, which may be of arbitrary semantic types, cannot be assigned truth values: it does not make sense to say that *"cat"* is either true or false. Sánchez Valencia (1991) formalizes what it means for a single word to entail another word through the notion of *semantic containment*. The definition of semantic containment serves as the basis for the lexical entailment relations defined by MacCartney (2009), which in turn serve as the basis for the relations we define in Section 2.4 and use throughout this dissertation.

Sánchez Valencia (1991) defines *semantic containment* between two words x and y conditioned on the fact that x and y are of the same semantic type. Let $x \Rightarrow_\tau y$ denote semantic containment between two words x and y, both of type τ. $x \Rightarrow_\tau y$ may be read informally as "x entails y" or "x implies y". Let D_τ denote the set of all words of type τ defined over the universe $\mathcal{U}$. I.e. D_e is the set of all entities in $\mathcal{U}$ and D_t is simply $\{0, 1\}$. Semantic containment is then defined recursively as follows:

- If $x, y \in D_t$, then $x \Rightarrow_t y$ iff $x = 0$ or $y = 1$. That is, for truth values, entailment follows the rules of boolean logic, in which true is entailed by everything, and false entails everything.

- If $x, y \in D_e$, then $x \Rightarrow_e y$ iff $x = y$. That is, entities can only entail one another if

they are the same exact entity.

- If $x, y \in D_{<a,b>}$, then $x \Rightarrow_{<a,b>} y$ iff $\forall a \in D_a, x(a) \Rightarrow_b y(a)$. E.g. both *"cat"* and *"animal"* are functions from entities (D_e) to truth values (D_t). Therefore, we can say that *"cat"* $\Rightarrow_{<e,t>}$ *"animal"* if for every x such that $[\![\text{"}cat''\!]\!](x) = 1$, we also have $[\![\text{"}animal''\!]\!](x) = 1$.

2.2.3. Basic Entailment Relations in Natural Logic

MacCartney (2009) generalizes Sánchez Valencia (1991)'s notion of semantic containment to include not just containment but also non-entailing relations such as semantic exclusion and semantic independence, as described below.

Set-Theoretic Definitions

MacCartney (2009) defines basic lexical entailment relations in terms of sets, as opposed to the functional definition used by Sánchez Valencia (1991). This leads to a slight shift of notation, in which words are used as though they denote sets, even when they in fact denote functions between sets or other higher-order operations. (E.g. while it is easy to conceptualize the set denoted by a common noun like *"cat"*, it is less intuitive to imagine the set denoted by a transitive verb like *"loves"*.) In effect, MacCartney (2009)'s definitions assume that lexical expressions of type $< a, b >$ are recursively instantiated using the definition of $x \Rightarrow_{<a,b>} y$ above to eventually arrive at truth values, thus making every lexical expression interpretable as the characteristic function of a set.[2]

We will follow MacCartney (2009) and be somewhat informal in our definitions from here forward, often leaving the precise types of words unspecified under the assumption that we only talk about entailment between words that are of the same semantic type. Below and throughout, lower case letters (x and y) refer to natural language strings of the same semantic type (τ), upper case letters (X and Y) refer to the sets denoted respectively by

[2]See Section 5.4 of MacCartney (2009) for more information.

those strings, and U refers to the space of all possible denotations for strings of type τ (i.e. the space of all denotations of words of the same type as x and y). MacCartney (2009) defines the following seven mutually-exclusive basic entailment relations that may hold between sets.

- Equivalence ($x \equiv y$): $X = Y$
- Forward Entailment ($x \sqsubset y$): $X \subset Y$
- Reverse Entailment ($x \sqsupset y$): $Y \subset X$
- Negation ($x \hat{} y$): $X \cap Y = \emptyset \wedge X \cup Y = U$
- Alternation ($x \mid y$): $X \cap Y = \emptyset \wedge X \cup Y \neq U$
- Cover ($x \smile y$): $X \cap Y \neq \emptyset \wedge X \cup Y = U$
- Independent ($x \# y$): all other cases

Inference Rules

The above relations imply inference rules for the associated lexical expressions. These inference rules along with examples are shown in Table 5. In this thesis, since our focus is on building systems for automatic natural language understanding rather than on modeling language for its own sake, we place greater emphasis on the inference rules than on the set theoretic definitions. The distinction will become especially relevant for our discussion of pragmatic inferences in Chapter 4.

Relation	Symbol	Inference Rule(s)	Example (x/y)
Equivalence	$x \equiv y$	$x \Leftrightarrow y$	couch/sofa
Forward Entailment	$x \sqsubset y$	$x \Rightarrow y$	couch/furniture
Reverse Entailment	$x \sqsupset y$	$y \Rightarrow x$	furniture/couch
Negation	$x \hat{} y$	$x \Leftrightarrow \neg y$	living/non-living
Alternation	$x \mid y$	$(x \Rightarrow \neg y) \wedge (y \Rightarrow \neg x)$	couch/table
Cover	$x \smile y$	None	living/non-human
Independent	$x \# y$	None	couch/antique

Table 5: Inference rules associated with the basic entailment relations defined in MacCartney (2009).

Atomic Edits

In practice, MacCartney (2009) operationalizes these inference rules in terms of *atomic edits* applied to natural language sentences, where an atomic edit is defined simply as the deletion ($DEL(\cdot)$), insertion ($INS(\cdot)$), or substitution ($SUB(\cdot,\cdot)$) of a subexpression.[3] For a linguistic expression s, $e(s)$ is the result of applying an atomic edit e to s. For example, if $s =$ *"She wore a red dress"* and $e = DEL($*"red"*$)$ then $e(s) =$ *"She wore a dress"*. The entailment relation (one of the seven basic relations defined above) that holds between an expression s and the edited expression $e(s)$ is said to be the relation "generated by" the edit e. This generated relation is denoted $\beta(e)$. For example, given s as above, $\beta(DEL($*"red"*$))$ should be Forward Entailment ($\sqsubset$), since $e(s) \Rightarrow s$ but $s \not\Rightarrow e(s)$ (Table 5).

2.2.4. Lexical Entailment Resources in NLP

Many natural language understanding tasks, including RTE, require knowledge of lexical entailment. As a result, there has been a great deal of effort invested in developing pre-constructed lexical semantics resources, which provide information about the relationships between words and phrases, and can be incorporated into a range of downstream NLU tasks. Below, I outline several approaches to creating such resources which are particularly relevant to the work presented in this thesis.

Manual Curation

Early work in natural language processing spurred the development of several hand-built linguistic resources. These resources were built by trained linguists and were intended to capture import semantic information that NLP systems could apply to tasks like information retrieval. The most widely known of the resources is WordNet (Fellbaum (1998)), a lexical ontology which covers nouns, verbs, adjectives, and adverbs and provides easy access to both

[3]Technically, atomic edits should specify not just the substring to be inserted, deleted, or substituted, but also the precise location at which the edit occurs (e.g. indices of token or character offsets). However, since in this thesis it will always be obvious from context where the edit is intended to be applied, we will omit indices in our notation.

taxonomic relations, like synonymy and hypernymy, as well as other semantic relations not naturally captured by the hierarchical structure, like meronymy (for nouns) and causal relations (for verbs). Other widely used lexical entailment resources include FrameNet (Baker et al. (1998)), PropBank (Kingsbury and Palmer (2002)), and VerbNet (Schuler (2005)). These resources are centered around verbs and their argument structure, and contain rich semantic annotation beyond simply lexical entailment information.

Distributional Similarity

Most automatically-constructed paraphrasing resources use a word's *distributional context*– i.e. its typical usage, computed over a large corpus–as the primary signal for determining its relationship with other words. One early and well-known resource is the DIRT database (Lin and Pantel (2001)). DIRT considers phrases to be synonymous if they have similar dependency contexts, following the intuition that synonymous verbs should tend to take the same arguments and synonymous nouns should tend to have the same modifiers. While the original DIRT resource considered all extracted pairs to be synonymous, later work attempted to infer directionality for the extracted paraphrase pairs (Bhagat et al. (2007); Kotlerman et al. (2010)). Other work has attempted to improve upon standard distributional methods by organizing the extracted paraphrases into a graph and using graph algorithms to further increase coverage and scalability (Szpektor et al. (2004); Berant et al. (2011); Brockett et al. (2013)). Recently, many NLP systems have shifted away from using "word list" style resources, in which paraphrases are listed explicitly as natural language strings, in favor of using *word embeddings*: dense low-dimensional vectors that capture a word's distributional usage such that words which appear in similar contexts tend to be near one another in vector space (Mikolov et al. (2013); Pennington et al. (2014)).

Lexico-Syntactic Patterns

While the distributional similarity approaches have been mostly concerned with finding synonymous word and phrases, a related line of work has focused specifically on mining

hypernym relations. The vast majority of such efforts are based on the concept of "lexico-syntactic patterns", or specific lexical templates like *"x is a y"* or *"y such as x"*, which are high-precision indicators that y is a hypernym of x. Hearst (1992) originally proposed a small number of hand-written patterns in order to mine taxonomic relations from text. Snow et al. (2006) built on this idea using dependency parses to automatically learn such patterns, and used the learned hypernyms to augment the WordNet noun hierarchy. Chklovski and Pantel (2004) and Hashimoto et al. (2009) use similar signals but focus on learning fine-grained relationships between verbs, such as *enablement* and *happens-before*. Most recently Shwartz et al. (2016) showed substantially improved hypernym detection by integrating these lexico-syntactic patterns with distributional word representations.

Bilingual Pivoting

Bannard and Callison-Burch (2005) proposed the method of *bilingual pivoting* for extracting paraphrases from the bilingual parallel corpora that are standardly used for machine translation. The method follows the intuition that two words or phrases are likely to have equivalent meaning if they can be translated to the same word or phrase in a foreign language. This technique was used to build the Paraphrase Database (Ganitkevitch et al. (2013)), currently the largest paraphrase resource in NLP, containing over 100 million paraphrase pairs. We describe the Paraphrase Database in detail in Section 2.2.5.

Limitations of Existing Resources

Manually-constructed resources like WordNet and FrameNet require large time commitments by expert annotators, making them slow and expensive to build and difficult to adapt to new languages or domains. Thus, the preference has been for automatically (or semi-automatically) constructed lexical entailment resources. This focus on scalability has resulted in large resources with high coverage but uninterpretable semantics. Whereas word pairs in WordNet are associated with well-defined semantic relations (e.g. *synonym*, *hypernym*, *antonym*), the relation that holds for word pairs extracted by automatic methods is

not clear. For example, if x and y are close in vector space according to a distributional semantics model, one can conclude only that x *occurs in similar contexts to* y; if x and y are associated via bilingual pivoting, one can conclude only that x *shares at least one translation with* y. Neither of these relations is meaningful or useful for the purposes of more complex natural language understanding tasks. Our work towards overcoming this limitation is presented in Chapter 3.

2.2.5. The Paraphrase Database

The Paraphrase Database (PPDB) serves as the basis for the work presented in Chapter 3. PPDB is a collection of paraphrases released by Ganitkevitch et al. (2013) and extracted from bilingual parallel corpora using the bilingual pivoting technique proposed by Bannard and Callison-Burch (2005). A "paraphrase rule" in PPDB consists of three components: a phrase (e_1), a paraphrase (e_2), a syntactic category. Paraphrase rules in PPDB fall into three categories: lexical, in which both e_1 and e_2 are single words; phrasal, in which at least one of e_1 or e_2 contains multiple words; and syntactic templates, or patterns containing non-terminal symbols that capture larger-scale paraphrastic rewrites like *"the* NP_1 *of* NP_2*"* $\rightarrow$ *"*NP_2*'s* NP_1*"*. In this thesis, we work only with lexical and phrasal paraphrases.

The intuition behind bilingual pivoting is that, for example, *"incarcerated"* is likely a good paraphrase of *"put in prison"* since they are both aligned to *"festgenommen"* in different sentence pairs in an English-German parallel corpus. Since *"incarcerated"* aligns to many foreign words (in many languages) the list of potential paraphrases is long. Paraphrases vary in quality since the alignments are automatically produced and noisy. In order to rank the paraphrases, Bannard and Callison-Burch (2005) define a paraphrase probability in terms of the translation model probabilities $p(f|e)$ and $p(e|f)$:

$$p(e_2|e_1) \approx \sum_f p(e_2|f)p(f|e_1) \tag{2.1}$$

Instead of ranking the paraphrases with a single score, paraphrases in PPDB are ranked

using a heuristic linear combination of 33 scores of paraphrase quality. These scores include the paraphrase probabilities computed according to Equation 2.1, the probability of the syntactic category given e_1 and e_2, and the number of times that e_1 has been aligned to e_2 in the bilingual corpus (the full list of scores is given in Appendix A.3.5). The combined score was used to divide PPDB into six increasingly large sizes: S, M, L, XL, XXL, and XXXL. PPDB-XXXL contains all of the paraphrase rules and has the highest recall, but the lowest average precision. The smaller sizes contain better average scores but offer lower coverage.

Our work in Chapter 3 builds on the English PPDB released by Ganitkevitch et al. (2013). Later extensions to the database include a multilingual PPDB covering 23 different languages (Ganitkevitch and Callison-Burch (2014)), and a re-ranked PPDB in which the heuristic scoring model described above was replaced by a supervised logistic regression model (Pavlick et al. (2015b)).

2.3. Compositional Entailment in Modifier-Noun Phrases

In the previous section, our focus was on *lexical entailment*: given a pair of words, how do we determine the semantic relationship between them? Current data-driven NLP methods attempt to infer lexical entailment relations by looking at the ways words are used in large corpora, for example by computing their distributional context or observing the words into which they are translated. Often, in NLP, these lexical entailment methods are extended to phrases as well: PPDB uses bilingual pivoting to learn paraphrases for *"little boy"*; the DIRT database uses distributional similarity to learn paraphrases for *"tried to solve"*. Ideally, however, these types of natural language expressions should be modeled compositionally. We should be able to combine the meaning of *"little"* with the meaning of *"boy"* in order to arrive at the meaning of *"little boy"*, and to combine the meaning of *"try"* with the meaning of *"solve"* in order to understand *"tried to solve"*.

This section overviews relevant background and prior work on *compositional entailment*:

given an expression containing more than one word, how do we derive the meaning of the whole from the meanings of the parts? In this thesis, we focus specifically in the case of composing modifiers (e.g. *"little"*, *"red"*, *"good"*) with noun phrases (e.g. *"boy"*, *"dress"*, *"chocolate chip cookie recipe"*). We do not address other types of composition (e.g. verb-verb compositions like *"tried to solve"*). Throughout this thesis, I will use MH to refer to a phrase that consists of a modifier M followed by a head noun H. H is assumed to be a common noun, and M is a modifier which might be an adjective (as in the MH *"red cup"*) or a noun (as in the MH *"coffee cup"*).

2.3.1. Classes of Modifiers in Formal Semantics

Recall from Section 2.2.1 that, in formal semantics, modification is modeled as function application, and a common noun is modeled as the characteristic function of a set of entities in the universe $\mathcal{U}$ (Heim and Kratzer (1998)). Naturally, in a compositional model of semantics, we would like such an interpretation to hold uniformly for all noun phrases. Therefore, the interpretation of a modified noun MH should similarly be a set of entities in $\mathcal{U}$.

$$[\![MH]\!] = \{e \in \mathcal{U} \mid e \text{ is a } MH\} \tag{2.2}$$

Traditionally, modifiers (specifically, adjectives) have been classified taxonomically according to the set-theoretic relationship between the denotation of the head noun H and that of the modified noun MH. In this way, Kamp and Partee (1995) categorizes modifiers as either *subsective* or *non-subsective*. Beyond this top-level distinction, some subsective modifiers can be defined more specifically as *intersective*, and all non-subsective modifiers can defined more specifically as either *privative* or *plain non-subsective*. These standard classes of modifiers are described below and summarized in Figure 5.

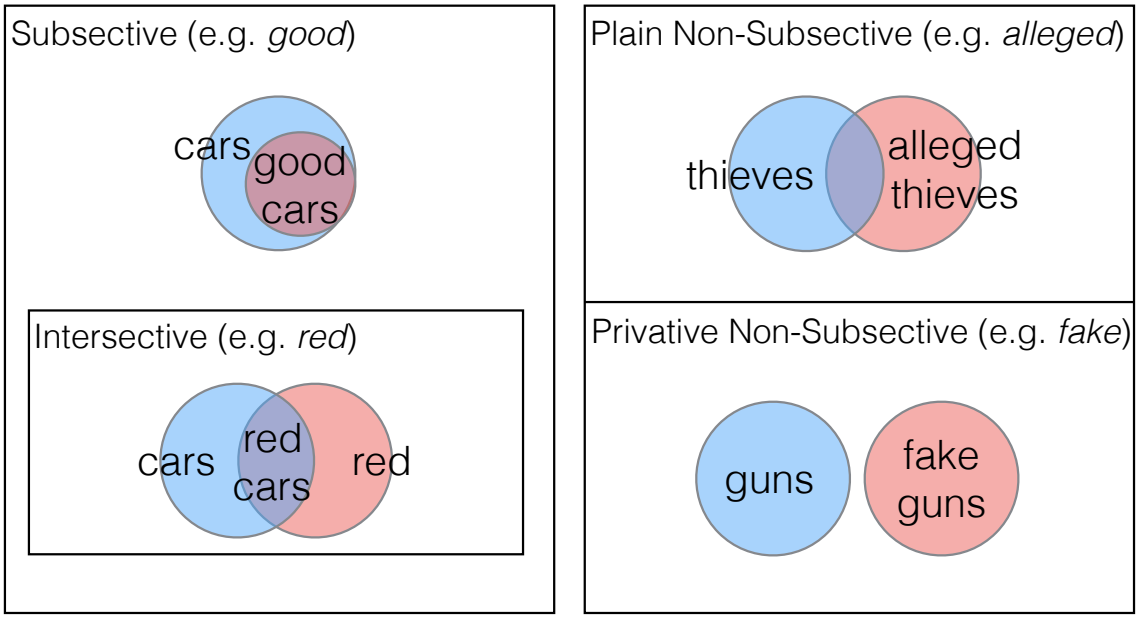


Figure 5: Classes of modifiers in formal semantics.

Subsective Modifiers

Subsective modifiers are modifiers which pick out a subset of the set denoted by the unmodified noun; that is:

$$M \text{ is subsective} \Leftrightarrow [\![MH]\!] \subset [\![H]\!] \tag{2.3}$$

Subsective modifiers are formalized as a functions of type $<< e,t >, < e,t >>$ which take as input the set denoted by the head noun H and return the set denoted by the modified noun MH, i.e. the subset of elements in $[\![H]\!]$ for which the modifier holds:

$$[\![MH]\!] = [\![M]\!]([\![H]\!]) = \{e \in [\![H]\!] \mid e \text{ is } M\} \tag{2.4}$$

We will discuss computational ways to operationalize Equation 5.2 in Chapter 5.

Note that, often, the meaning of a modifier is dependent on the meaning of the head noun being modified. For example, *"good"* is subsective: if e is a *"good student"*, e is not necessarily a *"good person"* in general. This example illustrates the difficulty in modeling

whether "e is *good*" in an absolute sense, independent of the context H in which e is being considered. In general, determining whether or not "e is M" can be deeply context dependent. We discuss this issue further in Section 2.3.3 and in Chapters 4 and 5.

Intersective Modifiers. Some subsective modifiers are considered *intersective*. Intersective modifiers can be interpreted as the characteristic function of a set, in the same way that a noun can. That is, intersective adjectives can naturally be formalized as functions of type $< e, t >$. For intersective adjectives, the denotation of MH is simply the intersection of the set denoted by M and the set denoted by H:

$$M \text{ is intersective} \Leftrightarrow [\![MH]\!] = [\![M]\!] \cap [\![H]\!] \tag{2.5}$$

For example, *"red"* is intersective in the context of *"car"*, so if e is a *"red car"*, e is both *"red"* and a *"car"*. Note that intersective modifies are just a special case of subsective modifiers as defined in Equation 5.2 in which the question of whether "e is M" equates to determining whether $e \in [\![M]\!]$.

Non-Subsective Modifiers

For non-subsective modifiers, in contrast, the denotation of MH is not a subset of H. Thus, formalizing the denotation of M as a function of type $<< e, t >, < e, t >>$ is less straightforward, as determining the domain of this function presents challenges: if $e \in [\![MH]\!]$ does not guarantee $e \in [\![H]\!]$, we cannot constrain the domain of $[\![M]\!]$ to be $[\![H]\!]$, as we could when M was subsective (Eq. 5.2). Formalizing the denotation of $[\![M]\!]$ in general is beyond the scope of this thesis. Rather, for non-subsective modifiers, we will focus only on the entailment relations between the set denoted by $[\![MH]\!]$ and that denoted by $[\![H]\!]$.

Plain Non-Subsective Modifiers. When M is plain non-subsective, there is no guaranteed entailment relationship between the set denoted by the modified noun and that

denoted by the noun alone, i.e.:

$$M \text{ is plain nonsubsective} \Leftrightarrow (\llbracket MH \rrbracket \not\subset \llbracket H \rrbracket) \wedge (\llbracket MH \rrbracket \cap \llbracket H \rrbracket \neq \emptyset) \tag{2.6}$$

In other words, if $e \in \llbracket MH \rrbracket$, there are possible worlds in which $e \in \llbracket H \rrbracket$ and there are possible worlds in which $e \notin \llbracket H \rrbracket$. The same is true if $e \notin \llbracket MH \rrbracket$: there are possible worlds in which $e \in \llbracket H \rrbracket$ and there are possible worlds in which $e \notin \llbracket H \rrbracket$. The modifier *"alleged"* is quintessentially plain non-subsective, since, for example, an *"alleged thief"* may or may not be a true *"thief"* and a true *"thief"* may or may not be an an *"alleged thief"*.

Privative Modifiers. Privative modifiers are defined as modifiers for which the set denoted by the modified noun is completely disjoint from the set denoted by the unmodified noun: i.e.:

$$M \text{ is privative} \Leftrightarrow \llbracket MH \rrbracket \cap \llbracket H \rrbracket = \emptyset \tag{2.7}$$

"Fake" is considered to be a quintessentially privative adjective since, by the usual definition of *"fake"*, a *"fake gun"* is expressly *not* a *"gun"*.

While the notion of privative adjectives as described above is widely accepted and often applied in NLP tasks (Amoia and Gardent (2006, 2007); Boleda et al. (2012); McCrae et al. (2014); Angeli et al. (2015)), recent linguistic theories have taken the position that in fact privative adjectives should be considered as simply another type of subsective adjective (Partee (2003); McNally and Boleda (2004); Abdullah and Frost (2005); Partee (2007)). Under this argument, the denotation of the noun H should be expanded to include those entities which belong to MH, so that the domain of $\llbracket M \rrbracket$ can be simply $\llbracket H \rrbracket$, as when M is subsective. This expanded denotation is used to account for the acceptability of the sentence *"Is that gun real or fake?"*, which is difficult to analyze if $e \in \llbracket$ *"gun"* $\rrbracket$ entails $e \notin \llbracket$ *"fake gun"* $\rrbracket$, as holds under the traditional definition of privatives. In more recent theoretical work, Del Pinal (2015) has argued that common nouns have a "dual semantic structure" and that non-subsective adjectives modify part of this meaning (e.g. the

functional features of the noun) without modifying the extension of the noun. Under this analysis, we can interpret a *"fake gun"* as having many, but not all, of the properties of a *"gun"*. Within NLP, there have been similar efforts to characterize privative modifiers more robustly. Nayak et al. (2014) categorize non-subsective adjectives in terms of the proportion of properties that are shared between H and MH and Pustejovsky (2013) focus on syntactic cues for exactly which properties are shared.

2.3.2. Adjective Noun Composition in Natural Logic

Set-Theoretic Basic Relations

Recall that the natural logic formalism as described in MacCartney (2009) defines seven basic entailment relations, which capture all of the possible ways that two sets defined in the same universe might relate to one another (see Section 2.2.3). Using these basic entailment relations to partition the space of possible relationships between $[\![H]\!]$ and $[\![MH]\!]$ results in a more fine-grained characterization of MH compositions than does the traditional taxonomy of modifiers just described. Table 6 depicts the relationship between these two set-theoretic classifications of modified noun phrases.

<table>
<tr><td rowspan="2">Subsective
$[\![MH]\!] \subset [\![H]\!]$</td><td>$[\![MH]\!] = [\![H]\!]$</td><td></td><td></td><td>Equiv. ($\equiv$)</td></tr>
<tr><td>$[\![MH]\!] \neq [\![H]\!]$</td><td></td><td></td><td>Forward ($\sqsubset$)</td></tr>
<tr><td rowspan="5">Non-Sub.
$[\![MH]\!] \not\subset [\![H]\!]$</td><td rowspan="2">Privative
$[\![MH]\!] \cap [\![H]\!] = \emptyset$</td><td>$[\![MH]\!] \cup [\![H]\!] = \mathcal{U}$</td><td></td><td>Negation ($\hat{}$)</td></tr>
<tr><td>$[\![MH]\!] \cup [\![H]\!] \neq \mathcal{U}$</td><td></td><td>Alt. ($|$)</td></tr>
<tr><td rowspan="3">Plain
$[\![MH]\!] \cap [\![H]\!] \neq \emptyset$</td><td>$[\![H]\!] \subset [\![MH]\!]$</td><td></td><td>Reverse ($\sqsupset$)</td></tr>
<tr><td rowspan="2">$[\![H]\!] \not\subset [\![MH]\!]$</td><td>$[\![MH]\!] \cup [\![H]\!] = \mathcal{U}$</td><td>Cover ($\smile$)</td></tr>
<tr><td>$[\![MH]\!] \cup [\![H]\!] \neq \mathcal{U}$</td><td>Indep. (#)</td></tr>
</table>

Table 6: Relationship between natural logic relations and formal semantics adjective classes. Table reads as a decision tree from left to right.

Basic Relations Defined on String Operations

We discussed in Section 2.2.3 that while the basic relations are defined between sets, it is often preferable, especially for NLP applications, to focus on the inference rules implied by the underlying set relations, rather than on the set relations themselves. In natural logic

(MacCartney (2009)), this is accomplished by focusing on the relations *generated* by atomic edits applied to natural language strings. For example, while formal semantics focuses on specifying the relationship between $[\![$ *"brown dog"* $]\!]$ and $[\![$ *"dog"* $]\!]$ in the abstract (across all possible worlds), natural logic focuses on determining the entailment relationship between a sentence s containing the word *"dog"* and a sentence $e(s)$ into which the word *"brown"* has been inserted in front of *"dog"*.

Table 7 shows examples of sentences and edits in which the composition of a modifier M with a noun H can generate each of the basic entailment relations previously described. The relation ($\beta(e)$) generated by the atomic edit is determined by the inferences that hold between s and $e(s)$. See Table 5 from Section 2.2.3 for a summary of the inference rules associated with each relation.

MH	s	e	$\beta(e)$
entire world	It is her favorite book in the world.	$INS($ *"entire"*$)$	$\equiv$
brown dog	Fido is a dog.	$INS($ *"brown"*$)$	$\sqsupset$
potential successor	She is the president's successor.	$INS($ *"potential"*$)$	$\sqsubset$
former senator	She is a senator.	$INS($ *"former"*$)$	$\mid$
alleged hacker	She is a hacker.	$INS($ *"alleged"*$)$	$\#$

Table 7: Basic entailment relations generated by modifier-noun composition–i.e. inserting modifiers in front of nouns in context.

Note that we do not offer examples for which MH composition generates the Cover ($\smile$) or the Negation ($\hat{}$) relation, as these two relations have the requirement that $[\![H]\!] \cup [\![MH]\!] = \mathcal{U}$, a difficult condition to meet in practice. I discuss our treatment of this "exhaustivity" constraint as it pertains to our work in Section 2.4.

Comparison to Formal Semantics Approach

Regarding modifier-noun composition, the atomic edit approach taken in natural logic makes no attempt to assign meaning to the modifier itself. This is unlike the function application approach taken in formal semantics. In formal semantics, a subsective modifier like *"brown"* carries some intrinsic meaning, which is used, for example, to discriminate the set of *"brown*

dogs" from the set of *"dogs"* more generally. MacCartney (2009)'s formulation does not require we assign any intrinsic meaning to words nor that we ground words to entities and relations either concretely (in terms of the real world) or abstractly (in terms of possible worlds).

This decrease in representational power, however, makes natural logic quite flexible for tasks like RTE, as it is able to avoid the need to address difficult theoretical problems, like the issue of domains and function types for non-subsective modifiers (Section 2.3.1). The edit-based formalization makes it possible for the natural logic framework to support the handling of issues like word sense, context, and pragmatics when reasoning about entailment, without needing a complete logical formalization of these complex phenomena. The examples in Table 7 illustrate how, by focusing on the relation generated by an edit in specific context, the natural logic framework sidesteps any formal treatment of issues such as definite and indefinite reference, temporal reasoning, and hyperbole. That is not to say that systems based on natural logic would not need to deal with these issues, but rather that the general natural logic framework outsources[4] these issues to whatever subroutine determines $\beta(e)$ for a given edit e and context s. Thus, the goal is simply to recognize *that* the context generates a particular relation, not to model *why* the context warrants that relation. It is an open question as to whether automatic systems can master the former without the latter. We return to this distinction again in our discussion of modifier-noun composition and semantic containment in Chapter 4.

2.3.3. Pragmatic Factors Affecting Modifier-Noun Composition

It has been widely observed that modifier-noun composition is a complex process. Baroni and Zamparelli (2010) observes that even for seemingly simple intersective modifiers like *"red"*, the meaning can change dramatically based on the noun being modified: e.g. a *"red Ferrari"* has a red outside, while a *"red watermelon"* has a red inside. This complexity

[4]It is worth noting that, while in theory the framework could support arbitrarily subtle attention to context, in practice, systems based on natural logic like that described in MacCartney (2009) do not address these complex issues at all.

exists even before considering less literal uses of *"red"*: e.g. *"red hand"*, *"red pen"*, *"red tape"*. Weiskopf (2007) has argued that modifier-noun composition is infinitely productive, and that modifiers are capable of expressing any possible semantic relation, given the right context. Weiskopf (2007) uses the differing interpretations of the MH *"Elvis stamp"* in the following two contexts to argue that pragmatics, not semantics, determines the meaning of the modifier:

- The US Postal Service is issuing a new stamp bearing an image of the older, fatter Elvis. The *Elvis stamp* is not expected to sell very well.
- The curators of Graceland are auctioning a stamp that is believed to have been licked by Elvis himself. The *Elvis stamp* is expected to fetch a high price.

Because of the wide range of ways in which modifiers can affect the meaning of the underlying noun, a lot of attention has been devoted to the pragmatic factors that enable humans to assign meaning to compositional noun phrases. Some have argued that compound noun phrases are not compositional at all, meaning that there is no transferable meaning that can be built into the M or the H themselves, and every unique modifier-noun pair requires specific handling (Lahav (1989); Fodor (1998)).

Dismissing compositionality altogether is extreme. But even among theories which support the compositionality of modifier-nouns, there is little disagreement that the composition is carried out in a context-sensitive way, and that people rely on shared context in order to understand and be understood (Reimer (2002); Rothschild and Segal (2009); Bach (2012)). Reimer (2002) argues that each noun has an "ordinary context" which is understood by interlocutors, and modifiers are interpreted with respect to this ordinary context unless otherwise specified. For example, in the sentence *"She is a good student"*, the adjective *"good"* is by default taken to mean that she studies hard and gets good grades, since studying and grades is the "ordinary context" surrounding *"student"*. If one intends to communicate that in fact she is a student who is *"good"* in the sense that she is an all-around kind,

charitable person, this sense of *"good"* would need to be specified more explicitly. Abdullah and Frost (2005) takes a similar approach for analyzing privative modifiers, stating that "the compound *"real fur"* is deemed necessary only when there is *"fake fur"* in the vicinity." That is to say, the "ordinary context" of *"fur"* (like that of most nouns) is that it is real, but this does not preclude *"fake fur"* from also being *"fur"*. We investigate pragmatic issues related to modifier-noun composition in Chapter 4.

2.4. Definition of Basic Entailment Relations used in this Thesis

The basic entailment relations defined by MacCartney (2009)'s natural logic (summarized in Section 2.2.3) provide a clear and simple vocabulary for talking about entailment in natural language. In Chapters 3 and 4, we would like to frame our analyses largely in terms of these relations. In order to do so, we make a few modifications to MacCartney (2009)'s definitions, as described below.

2.4.1. Relaxing Requirements of Exhaustivity

The definitions of two of MacCartney (2009)'s basic entailment relations, Negation ($\hat{}$) and Cover ($\smile$), require that the union of the sets standing in the relation be "exhaustive". That is, in order for x and y to stand in either of these two relations, everything in the universe U must be either x or y (or possibly both). While this is a meaningful theoretical distinction to make, its relevance to natural language, in practice, is arguably very limited. We therefore disregard these two relations, as described below.

Negation Relation. Recall the definition of Negation ($x \hat{} y$) is that $(X \cap Y = \emptyset) \wedge (X \cup Y = U)$. That is, everything in the universe U is either x or it is y, and it cannot be both simultaneously. This Negation relation allows one to make the strong inference that not only does $x \Rightarrow \neg y$ but also $\neg x \Rightarrow y$. This relation is primarily used to deal with explicit negation, e.g. it is often the basic entailment relation generated by a $DEL($*"not"*$)$ or $INS($*"not"*$)$ edit. However, in this thesis, we focus on lexical substitutions (Chapter 3) and on modifier-noun compositions (Chapter 4). It is difficult to come up with cases of in either of these

two settings for which ˆ is relevant. For example, in the context of lexical substitution, even indisputably antonymous words do not generate the negation relation: *"not good"* $\not\Rightarrow$ *"bad"* and *"not bad"* $\not\Rightarrow$ *"good"*. Even for words which are explicitly negated through a prefix (*"intelligent"*/*"unintelligent"*), people may perceive the dimensions under discussion as having a "middle ground": just because *"she isn't intelligent"*, we don't necessarily assume *"she is unintelligent"*. There is undoubtedly room to explore the pragmatic circumstances under which a lexical substitution may or may not yield a ˆ relation, but that is well beyond the scope of this thesis.

Therefore, we only use one relation to represent mutual exclusion, which we will refer to as *Exclusion* and represent with the $\dashv$ symbol. Our $\dashv$ relation signifies semantic exclusion in which $x \Rightarrow \neg y$ and $y \Rightarrow \neg x$ but it is not necessarily the case that $\neg x \Rightarrow y$ or that $\neg y \Rightarrow x$. Note that our exclusion relation is definitionally equivalent to MacCartney (2009)'s Alternation (|); the new name and symbol are for clarity only, so that we can still refer to MacCartney (2009)'s symbols when needed without confusion.

We acknowledge that removing the distinction between | and ˆ weakens the strength of the inferences we are able to make. For example, using the natural logic framework, and given the premise/hypothesis pair *"The claim is not true"*/*"The claim is false"*, a system which models only non-exhaustive exclusion (*"true"*$\dashv$*"false"*) can only conclude NONENTAILMENT. In contrast, a system which models exhaustive exclusion (*"true"*ˆ*"false"*) can draw the stronger conclusion of CONTRADICTION. In practice, however, these types of cases are rare, and we therefore see little disadvantage to simplifying the exclusion relations to remove the focus on exhaustivity.

Cover Relation. Recall the definition of Cover ($x \smile y$) is that $(X \cap Y \neq \emptyset) \wedge (X \cup Y = U)$. Like for the ˆ relation, the $\smile$ relation requires that everything in the universe U is either x or y, but in $\smile$ it is possible for something in the universe to be both x and y. MacCartney (2009) describes a canonical case of this relation as a word x paired with the negation of a hyponym of x, e.g. *"animal"*$\smile$*"nonhuman"*. MacCartney (2009) acknowledges that the

application of this relation is "not immediately obvious." The relation has since been shown to be applicable to the analysis of insertions and deletions when such edits involve one-way implicative verbs (Karttunen (2016)). However, for simple lexical substitutions or modifier insertions, which is the focus of this thesis, the Cover relation is unlikely to arise. Thus, we disregard this relation completely going forward.

2.4.2. Definitions

We define five basic entailment relations which we use to describe the relationship between two natural language strings: Equivalence ($\equiv$), Forward Entailment ($\sqsubset$), Reverse Entailment ($\sqsupset$), Exclusion ($\dashv$), and Independent ($\#$).

Within our definition of Exclusion ($\dashv$), there are some pairs which are intuitively interpreted as falling along a single dimension and thus are naturally interpreted as "opposites" (e.g. *"good/bad"*) and others which are clearly mutually exclusive but more categorical than bipolar (e.g. *"dog/cat"*). When it is necessary to distinguish, we use $\dashv_{opp}$ to refer to natural opposites and $\dashv_{alt}$ to refer to mutually-exclusive alternatives under a common category.

Within our definition of Independent ($\#$), there are some paris which are semantically related but not through entailment, for example meronyms (*"eye"*/ *"face"*) or derivationally related terms (*"academy"*/ *"academia"*). When relevant, we will refer to these pairs as "Otherwise Related" and denote this type of relatedness with the $\sim$ symbol. We will refer to the remaining independent relations as "Unrelated", denoted using the $\not\sim$ symbol.

Throughout this thesis, unless otherwise specified, the below names, symbols, and associated inference rules will refer to the definitions given here.

- Equivalence ($x \equiv y$): $x \Leftrightarrow y$. E.g. *"couch"*/ *"sofa"*
- Forward Entailment ($x \sqsubset y$): $x \Rightarrow y$ E.g. *"couch"*/ *"furniture"*
- Reverse Entailment ($x \sqsupset y$): $y \Rightarrow x$ E.g. *"furniture"*/ *"couch"*

- Exclusion ($x \dashv y$): $y \Rightarrow \neg x \wedge y \Rightarrow \neg x$
  - Opposites ($x \dashv_{opp} y$): $x \dashv y$, and humans typically view x and y as "opposites", or the two ends of a single bipolar dimension. E.g. *"good"/"bad"*
  - Alternatives ($x \dashv_{alt} y$): $x \dashv y$, and humans typically view x and y as mutually exclusive types under a common category (and there are typically more than two alternatives under that category). E.g. *"couch"/"table"*
- Independent ($x \# y$) : $x \not\Rightarrow y \wedge y \not\Rightarrow x$
  - Otherwise Related ($x \sim y$) : $x \# y$ but there is some natural relationship between x and y that can't be captured by entailment. E.g. *"couch"/"cushion"*
  - Unrelated ($x \not\sim y$) : $x \# y$ and not $x \sim y$. E.g. *"couch"/"antique"*

CHAPTER 3 : Lexical and Non-Compositional Entailment

Lexical semantics is the subproblem of RTE concerned with the semantic relations that hold between individual words. For example, lexical semantics addresses whether similar words should be treated as effectively equivalent (*"couch"*/*"sofa"*) or as mutually exclusive (*"couch"*/*"table"*). In this chapter, I discuss our work on assigning lexical entailment relations to pairs of words and phrases. The ultimate goal of the work presented here is to build a web-scale lexical entailment resource entirely automatically while ensuring the level of semantic interpretability available in hand-built resources. As discussed in Section 2.2.4, most existing automatically-constructed resources do not provide a clear definition of the semantics they capture. We aim to address this limitation by training a statistical classifier to differentiate between the five distinct basic entailment relations defined in Section 2.4. We use this classifier to assign an entailment relation to each paraphrase rule in the Paraphrase Database (PPDB), transforming it into a valuable lexical semantics resource for language understanding tasks. We illustrate at the end of the chapter that, in a downstream RTE task, our automatically-constructed resource performs as well as manually-constructed resources.

In Section 3.1, we build dataset of pairs of natural language expressions labeled according to the basic entailment relations defined in Section 2.4. In Section 3.2 we use this data to train a supervised lexical entailment classifier, which we use to assign a explicit entailment relation to every paraphrase rule in PPDB. We evaluate the quality of the resulting lexical entailment resource in two ways: first, by analyzing how well the relations assigned to arbitrary pairs in PPDB agree with human labels (Section 3.3) and second, by measuring the extent to which the automatically-annotated PPDB improves performance of an end-to-end RTE system (Section 3.4). Section 3.5 concludes with a discussion of directions for future exploration.

3.1. Annotating Basic Entailment Relations

In this Section, we construct labeled datasets consisting of pairs of lexical expressions each associated with one of the seven basic fine-grained relations defined in Section 2.4: $\equiv$, $\sqsubset$, $\sqsupset$, $\dashv_{opp}$, $\dashv_{alt}$, $\sim$, and $\not\sim$. These datasets are used to train and evaluate our lexical entailment classifier in Section 3.2.

3.1.1. Assumptions about Context

We would like to label the pairs in PPDB using the entailment relations defined in Section 2.4, derived from MacCartney (2009)'s formulation of natural logic. However, PPDB is a static resource consisting of pairs of context-independent natural language strings. Thus, we will have to make assumptions about context and word sense, and it is inevitable that these assumptions will not hold in all the cases in which the resource might be used. This does not mean that building a static lexical entailment resource is fruitless, and in fact, context-independent lexical entailment resources are pervasive in NLP (see Section 2.2.4). Still, determining which basic entailment relation should be assigned to two strings without knowledge of the context in which it will be applied presents challenges. We follow the precedent set in the original definition of the RTE task (Section 2.1.2) in which we appeal to human intuition to handle difficult or ambiguous cases. We acknowledge that doing so makes our basic entailment relations less well-defined than would be ideal. Nonetheless, we show in Section 3.4 that, despite these imperfect definitions, the basic entailment relations as we define them provide useful signal for the downstream RTE task.

In this chapter, we will say that the basic entailment relation from Section 2.4 that holds between two out-of-context linguistic expressions x and y is equivalent to the basic entailment relation that would be generated by the atomic edit $e = SUB(x, y)$ if it were applied to a context s having the following properties:

1. **Co-Reference**: Given the sentences s and $e(s)$ (the substitution edit applied to s), x and y corefer (if x and y are nouns) or the arguments of x and y corefer (if x and y are

functional types such as modifiers or verbs). E.g. the noun pair *"mother"/"father"* should be evaluated in a context such as *"Sam is my mother"* rather than a context such as *"I had dinner with my mother"*; the verb pair *"sitting"/"running"* should be evaluated in a context such as *"Sam is sitting in the park"* rather than a context such as *"Someone is sitting in the park"*.

2. **Most Relevant Word Sense**: If there is any reasonable context in which x and y stand in a relation other than the independence (#) relation, s captures that context. E.g. the noun pair *"bank"/"riverside"* should be evaluated in a context such as *"We used to picnic along the bank"* rather than a context such as *"I had to run to the bank to get money"*, whereas the noun pair *"bank"/"credit union"* should be evaluated in the latter context and not the former.

3. **Most Frequent Word Sense**: If x and y can stand in multiple non-independent relations, s captures the most typical sense of x in which it is not independent of y. E.g. the noun pair *"bird"/"woman"* should be evaluated in a context like *"He studies migration patterns of North American birds"* rather than a context such as *"Dorothy is a silly old bird"*. We defer to human intuition about what constitutes "most typical".

The above assumptions are incorporated implicitly into our annotation task, described below.

3.1.2. Design of Annotation Task

We use Amazon Mechanical Turk (MTurk) to gather manual annotations. Our task design is described below. A comparison of alternative designs for the annotation task is given in Appendix A.1.

Task Parameters and Setup

Our interface shows each paraphrase pair out of context and asks workers to choose between the options shown in Table 8. Our full annotation guidelines are shown in Appendix A.2.

Basic Entailment Relation	Symbol	MTurk Description
Equivalence	$\equiv$	x is the same as y
Forward Entailment	$\sqsubset$	x is more specific than/is a type of y
Reverse Entailment	$\sqsupset$	x is more general than/encompasses y
Exclusion, Opposites	$\dashv_{opp}$	x is the opposite of y
Exclusion, Alternatives	$\dashv_{alt}$	x is mutually exclusive with y
Independent, Otherwise Related	$\sim$	x is related in some other way to y
Independent, Unrelated	$\not\sim$	x is not related to y
Not Applicable		Relation cannot be determined

Table 8: Descriptions of basic entailment relations from Section 2.4 shown to annotators on Amazon Mechanical Turk.

The options shown correspond to our definitions of basic relations, but are simplified to be easily understood by non-expert annotators. Annotators are also given the option to indicate "not applicable" if the relation cannot be determined, for example if either x or y is in a language other than English. The distinction between Otherwise Related ($\sim$) and Unrelated ($\not\sim$) is admittedly vague. Ultimately, the distinction between these to classes is not important, since, from an entailment perspective, both are considered Independent (#). However, we distinguish these two in our annotation interface, largely to prevent annotators from over-assigning to the Equivalence and Entailment relations, e.g. by labeling meronymy (*"eye"*/*"face"*) as Forward Entailment ($\sqsubset$).

Each of our annotation tasks (known on MTurk as a "human intelligence tasks" or "HITs") contains 10 pairs to label: eight pairs from PPDB for which we need labels and two quality control pairs (described below). We pay \$0.15 per HIT, or \$0.015 per labeled pair. We make our tasks available only to workers based in the US, and only to workers who have completed at least 50 HITs and have at least a 90% approval rate. We show each pair to 5 workers, taking the majority label as truth.

Quality Control

In order to measure worker reliability, we embed synonyms and antonyms drawn from WordNet as gold-standard examples of the Equivalence ($\equiv$) and Opposite ($\dashv_{opp}$) relations, respectively. We draw random pairs of words which we use as gold standard examples of the Unrelated ($\not\sim$) relation. After inspecting the WordNet hypernym and hyponym pairs ourselves, we determined they were too unclear to be used as gold-standard examples of the Forward and Reverse Entailment relations ($\sqsubset$ and $\sqsupset$), so workers' accuracy is not explicitly evaluated for those relations. We considered a worker's answer to be correct if they labeled the WordNet synonyms as any of Equivalent, Forward Entailment, or Reverse Entailment. We chose to be lenient in this regard after inspecting a sample of synonyms extracted from WordNet and determining that many could be accurately labeled as Reverse Entailment (e.g. *"morning"/"sunrise"*) or as Forward Entailment (e.g. *"fabric"/"material"*).

Each HIT consisted of two control questions, and workers who fell below 50% accuracy were rejected. Workers achieved 82% accuracies on our controls overall: 92% on the Unrelated pairs, 70% on the Equivalence pairs, and 64% on the Opposite pairs. Of the Equivalence pairs, 50% were labeled Equivalence and another 20% were labeled either Forward Entailment or Reverse Entailment.

3.1.3. Labeled Datasets for Training and Evaluation

We use the above-described annotation task on Amazon Mechanical Turk to create several labeled datasets for training and evaluating the classification model that will be described in Section 3.2. These datasets are described below and summarized in Table 12. Each dataset consists of tuples of the form $\langle (w_1, t_1), (w_2, t_2) \rangle$ where w_i is a lexical expression–either a single word or a multiword phrase–and t_i is a syntactic category.

Random Sample of Pairs in PPDB

Our first dataset, PPDBSAMPLE, contains a random sample of pairs appearing in the Paraphrase Database. To build PPDBSAMPLE, we take a stratified random sample across the six sizes of PPDB (see Section 2.2.5), so as to bias the sample toward good and interesting paraphrases, rather than noisy paraphrases, which would likely dominate if drawn uniformly at random from PPDB-XXXL. We assign a syntactic category to each pair by mapping the syntactic category associated with the pair in PPDB coarsely onto 'noun', 'verb', 'adjective/adverb', or 'other'. Our sample consists of 22,817 paraphrase pairs: 10,800 lexical paraphrases, 10,126 "one-to-many" paraphrases in which one phrase in the pair is lexical and the other is phrasal, and a small sample of 1,932 phrasal paraphrases. Table 9 shows a sample of pairs from the dataset.

achieve/get, active/formal, appeal/appeal board, boards/executive boards, constitute/fill, cover/cure, danger/grave danger, defence property/military goods, energetic/serious, enforcement/running, floor/your word, objectivity/subject, proper operation of the internal market/smooth functioning of the internal market, radioactive materials/radioactivity, redo/restore, refuse/revoke, remote/short, space/outer space, steam/this trend, week/last week

Table 9: Random sample of noun pairs in the PPDBSAMPLE dataset.

Each pair in PPDBSAMPLE was annotated by 3 workers. We take the true label of a pair to be the majority label across workers, breaking ties randomly.

Exhaustive Set of Pairs from RTE Benchmark Data

We design another two datasets which consist solely of paraphrase pairs in PPDB which also appear in established benchmark datasets for the RTE task. The intent of these sets is to test our model on paraphrase pairs that are likely to be "relevant" for RTE systems. Specifically, we intersect PPDB separately with the vocabulary of two benchmark RTE datasets, SICK and RTE2, and refer to the resulting datasets as PPDBSICK and PPDBRTE2, respectively.

See Section 2.1.3 for a description of these benchmark datasets.

Both of our benchmark datasets consist of pair of sentences, a premise p and a hypothesis h. We tokenize, POS tag, and parse all of the sentences in each dataset using the Stanford CoreNLP pipeline (Manning et al. (2014)). Given a set of parsed p/h pairs, we select all tuples $\langle (w_1, t_1), (w_2, t_2) \rangle$ such that:

1. Both w_1 and w_2 contain three words or fewer.

2. There is some p/h pair such that w_1 appears in p and w_2 appears in h.

3. $\langle w_1, w_2 \rangle$ appears in PPDB-XXXL.

Tables 10 and 11 show sample pairs from PPDBSICK and PPDBRTE, respectively.

a group/camera, aircraft/an airplane, baby/the little, ball/snowball, band/boy, clear water/water, come/racing, cross/trunk, current/water, edge/sand, event/person, full/milk, group/restaurant, man/talk, person/tail, playing/ride, race/the track, reading/sing, side/stand, surfboard/wall

Table 10: Random sample of noun pairs in the PPDBSICK dataset.

bill/day, business/talk, community/live, completing/give, construction/propose, court/federal, declaration/be, division/member, early/israel, economy/more, force/promotion, health/people, in prison/jail, iran/tehran, israeli/leader, issue/estate, meeting/representative, organization/response, senator/speak, terrorist/terrorist attack

Table 11: Random sample of noun pairs in the PPDBRTE dataset.

The POS tag t_i associated with w_i is the tag or tag sequence assigned by the parser to w_i in context of the full sentence in which it appeared (either p or h). This means that for these datasets, the same phrase pair might appear multiple times with different POS tags. We allow $\langle w_1, w_2 \rangle$ to appear with any syntactic category in PPDB, we do not require that it match the category with which it appears in the sentence. Each pair was annotated by 5 workers and we take the true label to be the majority label.

Label Distributions and Annotator Agreement

Table 12 shows the distribution of labels obtained for the pairs in each of the described datasets. Together, the Independent classes ($\sim$ and $\not\sim$) constitute the majority of pairs in all three datasets. The PPDBSAMPLE dataset contains proportionally fewer Unrelated ($\not\sim$) pairs (24%) than do the RTE-filtered datasets. In all three datasets, the Exclusion class ($\dashv$) is infrequent, in total constituting about 7% of the pairs in PPDBSAMPLE and in PPDBSICK, and only 3% in PPDBRTE.

	$\equiv$	$\sqsubset$	$\dashv_{alt}$	$\dashv_{opp}$	$\not\sim$	$\sim$	NA	Total
PPDBSAMPLE	15%	25%	5%	2%	24%	25%	5%	22,817
	3,414	5,695	1,189	397	5,401	5,711	1,051	
PPDBSICK								
Train	8%	26%	3%	5%	39%	19%	<1%	4,790
	394	1,240	136	220	1,871	920	9	
Test	9%	26%	4%	3%	39%	19%	<1%	5,084
	443	1,321	228	147	1,976	956	13	
PPDBRTE2								
Dev	7%	21%	2%	1%	51%	17%	1%	9,299
	651	1,945	163	98	4,783	1,548	111	
Test	7%	20%	2%	1%	54%	18%	1%	8,835
	636	1,776	151	81	4,783	1,603	78	

Table 12: Distribution of basic entailment relations appearing in our annotated datasets. These datasets are used for training and evaluating our lexical entailment classifier.

On inspection, we do see that annotators commonly assign pairs to Unrelated ($\not\sim$) that ideally would be labeled as Alternatives ($\dashv_{alt}$). Table 13 provides several examples. Based on the examples shown, it appears that humans struggle to conceptualize two words as alternatives under a common category when the category is too abstract or too far removed from the words under consideration: e.g. humans to no consider *"dog"* and *"dirt"* be be alternatives under the category *"thing"*. In practice, this error does not seem to translate into errors in the downstream RTE task (Section 3.4), as systems (like humans) are rarely asked to make inferences which hinge on recognizing, for example, that *"Fido is a dog"* is incompatible with *"Fido is dirt"*. The relatively frequent presence of these errors, however,

is interesting and may be relevant to future work on lexical entailment in general and on taxonomies in particular.

bank/country, bird/boy, blade/man, conference/police, clothing/hand, dirt/dog, football/table, gun/kid, man/sky, people/time, water/wood

Table 13: Examples of pairs labeled as Unrelated ($\not\sim$) which would have been better labeled as Alternatives ($\dashv_{alt}$).

The inter-annotator agreement for each dataset, measured using Fleiss's κ (Fleiss et al. (2013)), is shown in Table 59. Note that κ is lower when label distributions are skewed, since the computation assumes that the probability of randomly choosing a label is equal to that label's frequency in the dataset. The observed agreement measures support the intuition that lexical entailment annotation is more straightforward when the vocabulary is more concrete. Agreement is highest on the PPDBSICK dataset, which is based on image captions and covers a vocabulary of mostly common nouns and simple adjectives (Table 10), and is lower for PPDBSAMPLE, which contains paraphrases extracted from a variety of corpora and contains a greater proportion of abstract phrases (Table 9).

	κ	# Pairs	# Annotators
PPDBSAMPLE	0.20	22,817	599
PPDBSICK	0.36	9,874	648
PPDBRTE2	0.31	18,134	697

Table 14: Inter-annotator agreement for each of the labelled datasets.

3.2. Supervised Model for Lexical Entailment Classification

We now turn to the task of automatically determining the basic entailment relation that holds between two natural language strings. We aim to build a statistical classifier which takes as input a pair of linguistic expressions and returns one of the basic entailment relations defined in Section 2.4. We will use this classifier to automatically add fine-grained semantic relations to each of the phrase pairs in PPDB in Section 3.3.

3.2.1. Classifier Configuration

We will train our classifier using the labeled datasets collected in Section 3.1. Because of the low frequency of exclusion relations in PPDB (Table 12), we do not attempt to automatically differentiate between the finer-grained $\dashv_{opp}$ and $\dashv_{alt}$ relations. Additionally, for simplicity, we fix the direction of the $\sqsubset$ and $\sqsupset$ pairs so that all are considered as $\sqsupset$ relations. Thus, we build our classifier to distinguish between 5 classes: $\{\equiv, \sqsupset, \dashv, \sim, \not\sim\}$.

We use the scikit-learn toolkit (`http://scikit-learn.org`) to train a logistic regression classifier. In order to overcome the imbalanced distribution of our data, we subsample training examples from each class inversely proportionally to the class's frequency in the training data (Table 12); this is corresponds to the `class_weight='auto'` parameter setting. We tune the regularization parameter using cross-validation on the training data.

3.2.2. Feature Groups

We compute a variety of features, which we organize into six feature groups, named as follows and described below: LEXICAL, WORDNET, DISTRIBUTIONAL, PATTERN, PARAPHRASE, and TRANSLATION. For more precise definitions and feature templates, see Appendix A.3.

For analysis purposes, we differentiate between features which rely on patterns derived from large monolingual corpora and those which rely on patterns derived from bilingual parallel corpora. When relevant, MONOLINGUAL refers to the combination of the DISTRIBUTIONAL and PATTERN feature groups, and BILINGUAL refers to the combination of the PARAPHRASE and TRANSLATION feature groups.

In the descriptions below, w_1 and w_2 refer to lexical items and t_1 and t_2 are their respective syntactic categories.

Lexical Features

We compute a variety of simple lexical features for each phrase pair, including: the lemmas, part-of-speech tags, and phrase lengths of w_1 and w_2; the substrings shared by w_1 and w_2; and the Levenstein, Jaccard, and Hamming distances between w_1 and w_2. This feature group is referred to as LEXICAL.

WordNet Features

For each pair $\langle (w_1, t_1), (w_2, t_2) \rangle$, we include indicator features to capture the relation or relations to which the pair can be assigned according to WordNet. This feature group is referred to as WORDNET.

Distributional Features

We follow Lin and Pantel (2001) in building distributional context vectors from dependency-parsed corpora. Given a dependency context vectors for w_1 and w_2, we compute the number of shared contexts, as well as the cosine similarity, Jaccard distance, and several perviously-proposed distributional similarities measures. Specifically, we compute `lin_similarity`, a symmetric similarity measure proposed by Lin (1998) as defined below:

$$\texttt{lin\_similarity} = \frac{\sum\limits_{c \in W_1 \cap W_2} W_1(c) + W_2(c)}{\sum\limits_{c \in W_1} W_1(c) + \sum\limits_{c \in W_2} W_2(c)} \tag{3.1}$$

where W_i is the set of contexts in which w_i appears and $W_i(c)$ is the number of times w_i has been observed in context c. We also compute `weeds_similarity`, a variation proposed by Weeds et al. (2004) and aimed at capturing asymmetric similarity, as defined below.

$$\texttt{weeds\_similarity} = \frac{\sum\limits_{c \in W_1 \cap W_2} W_1(c)}{\sum\limits_{c \in W_1} W_1(c)} \tag{3.2}$$

This group of features is referred to collectively as DISTRIBUTIONAL.

Lexico-Syntactic Pattern Features

Hearst (1992) and Snow et al. (2004) exploit certain textual patterns (e.g. *"x and other y"*) in order to infer hypernym relations from text. We follow Snow et al. (2004) in using dependency parsed corpora to automatically recognize these "lexico-syntactic patterns", but extend it to include all of our basic relations. We refer to the features in this group collectively as PATTERN.

Paraphrase Features

There are a variety of features distributed with PPDB, which we include in our classifier. These include 33 different measures used to sort the goodness of the paraphrases, including distributional similarity, bilingual alignment probabilities, and lexical similarity. These features combined are referred to as PARAPHRASE features.

Translation Features

PPDB is based on the "bilingual pivoting" method, in which two phrases are considered paraphrases if they share a foreign translation. The English PPDB was built by pivoting through 24 foreign languages. We use the pivot words from all of these languages to derive a set of features, including the number of foreign language translations shared by w_1 and w_2 for each of the languages separately and collectively. We compute `translation_similarity`, an asymmetric measure of the bilingual similarity of two words, as follows.

$$\texttt{translation\_similarity} = \frac{\mid \tau_*(w_1) \cap \tau_*(w_2) \mid}{\mid \tau_*(w_1) \mid} \tag{3.3}$$

where $\tau_*(w_i)$ is the set of all the translations of w_i across all 24 languages. We refer to this group as TRANSLATION features.

3.2.3. Feature Analysis

The features used in our classifier are largely based on previously-used methods for automatically inferring related words from text. However, in most prior work, these methods are used in isolation, or in applications which focus on a specific type of semantic relation (e.g. synonymy or hypernymy). It is therefore interesting to analyze the strengths and weaknesses of each feature group for differentiating between our five fine-grained entailment relations.

All of the below results are obtained by running ten-fold cross validation on the training split of the PPDBSICK dataset (Section 3.1.3).

Ablation Analysis

Table 15 shows the classifier's overall performance. The classifier achieves good overall performance, even for relations which are relatively infrequent in the training data.

	Frequency	Accuracy	F1
Unrelated ($\not\sim$)	39%	88%	0.79
Equivalence ($\equiv$)	8%	81%	0.57
Entailment ($\sqsupset$)	26%	76%	0.68
Exclusion ($\dashv$)	8%	73%	0.49
Otherwise Related ($\sim$)	19%	64%	0.51

Table 15: Accuracy and F1 score by classifier on 10-fold cross validation over PPDBSICK training data.

Table 16 shows the performance when ablating each of the feature groups. The BILINGUAL features (PARAPHRASE and TRANSLATION) are especially important for distinguishing the Equivalence class ($\equiv$), and the PATTERN and WORDNET features are important for the Exclusion class ($\dashv$). The LEXICAL feature group exhibits strong performance for classifying all relation types; this is likely because this group indirectly captures both negation words (e.g. *"no"*) and substring features (*"little boy"* $\sqsubset$ *"boy"*).

		Δ F1 when excluding					
	All	LEXICAL	DISTR.	PATTERN	PARA.	TRANS.	WORDNET
$\not\sim$	79.0	-1.99	-0.24	-1.23	-1.67	-0.24	-0.12
$\equiv$	56.8	-3.53	+0.22	-0.75	-2.44	-3.67	+0.46
$\sqsupset$	67.9	-4.58	-0.25	-0.83	-0.76	-0.65	-1.59
$\dashv$	48.5	-4.02	-0.76	-2.88	+0.29	-0.00	-2.23
$\sim$	50.6	-4.93	-0.46	-0.75	-1.19	-0.89	-0.32

Table 16: Change in F1 score (× 100) achieved by classifier when ablating each feature group.

Monolingual vs. Bilingual Similarity Metrics

Table 17 shows the "most similar" pairs in the PPDBSICK training set, according to the various types of similarity metric defined among our features (see Section 3.2.2). Our symmetric monolingual score (`lin_similarity`, Eq. 3.1) consistently identifies Exclusion ($\dashv$) pairs, while our asymmetric monolingual score (`weeds_similarity`, Eq. 3.2) is good for identifying Entailment ($\sqsupset$) pairs; none of the monolingual scores we explored were effective in making the subtle distinction between Equivalent and Entailment. In contrast, the bilingual similarity metric (`translation_similarity`, Eq. 3.3) is fairly precise for identifying Equivalent pairs, but provides less information for distinguishing between the different types of non-equivalent relations, such as distinguishing Entailment ($\sqsupset$) from Unrelated ($\not\sim$).

These differences are further exhibited in the confusion matrices shown in Figure 6: when the classifier is trained using only the MONOLINGUAL feature groups, it misclassifies 26% of Exclusion pairs as Equivalent, whereas the classifier trained with the BILINGUAL feature groups makes this error only 6% of the time. However, the classifier trained with the BILINGUAL feature groups completely fails to predict the Entailment class, calling over 80% of such pairs Equivalent or Otherwise Related ($\sim$).

3.3. Intrinsic Evaluation of Predicted Relations

We now evaluate the predictions of our classification model by measuring how well its predictions match human judgements of lexical entailment relations for pairs in a held out

`cosine_similarity` monolingual, symmetric		`lin_similarity` monolingual, symmetric	
$\sqsupset$	shades/the shade	$\dashv$	large/small
$\sqsupset$	yard/backyard	$\equiv$	few/several
$\not\sim$	each other/man	$\dashv$	different/same
$\sqsupset$	picture/drawing	$\dashv$	other/same
$\sim$	practice/target	$\dashv$	put/take
`weeds_similarlity` monolingual, asymmetric		**`translation_similarity`** bilingual, asymmetric	
$\sqsupset$	boy/little boy	$\equiv$	dad/father
$\sqsupset$	man/two men	$\sqsupset$	some kid/child
$\sqsupset$	child/three children	$\equiv$	a lot of/many
$\equiv$	is playing/play	$\equiv$	female/woman
$\sqsupset$	side/both sides	$\equiv$	male/man

Table 17: Most similar pairs (x/y) in PPDBSICK training data, according to various similarity measures, along with their manually classified entailment labels.

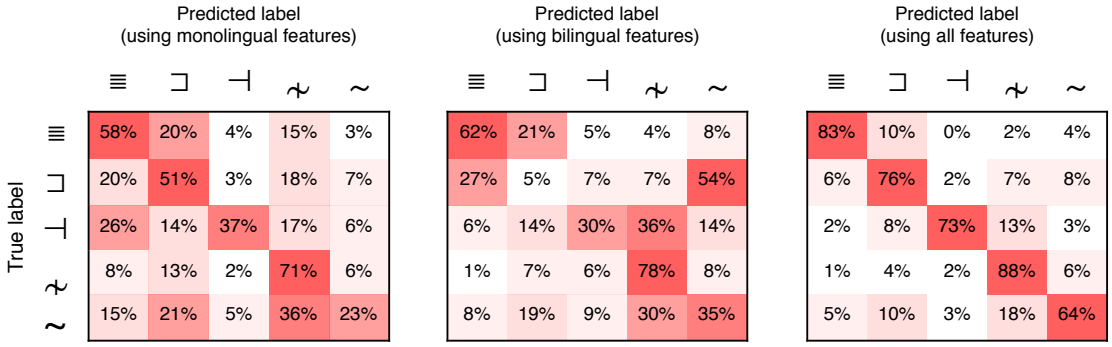


Figure 6: Confusion matrices for classifier trained using only MONOLINGUAL versus only BILINGUAL. True labels are shown along rows, predicted along columns.

test set. We consider two different settings. First, we evaluate on the subset of pairs in PPDB which also appear in the standard RTE benchmark dataset. Second, we evaluate the quality of the predictions when we use the classifier to label all pairs occurring in PPDB. PPDB overall contains many abstract expressions (e.g. *"go back"/"start all over again"*) and phrases with complex syntactic categories (e.g. *"which have resulted"/"and that have led"*). Such pairs don't necessarily lend themselves well to categorization according to the features on which our classifier relies (Section 3.2.2). In contrast, the vocabulary of the benchmark RTE datasets is more concrete and better suited to our lexical entailment

classification. Thus, we expect our classifier's predictions for phrase pairs occurring in the RTE data to be better than its predictions when applied to the entirety of PPDB.

3.3.1. Performance on Paraphrase Pairs Occurring in RTE Data

We evaluate the classifier on the PPDBSICK and PPDBRTE2 datasets described in Section 3.1.3. These datasets are built by intersecting the phrase pairs in PPDB with the vocabularies of the SICK (Marelli et al. (2014b)) and RTE2 (Bar Haim et al. (2006)) datasets, respectively. The train and test splits for PPDBSICK come from intersecting PPDB separately with the standard train split of SICK and the standard test split of SICK. To evaluate on PPDBSICK we train and tune the classifier using only the training split of PPDBSICK, and test on the test split. The process is analogous for evaluating on PPDBRTE2.

Tables 18 and 19 show the precision and recall achieved by the classifier on each of the basic entailment relations for the held out test sets from PPDBSICK and PPDBRTE2. Performances are overall higher for PPDBSICK than for PPDBRTE. This is likely largely due to that fact that the SICK dataset is derived from image captions, and thus covers a much simpler vocabulary than the RTE2 dataset, which is drawn from news text.

	Freq.	Precision	Recall	F score
$\not\sim$	39%	0.842	0.876	0.859
$\equiv$	8%	0.704	0.831	0.762
$\sqsupset$	26%	0.798	0.760	0.779
$\dashv$	7%	0.737	0.733	0.735
$\sim$	19%	0.706	0.637	0.670

Table 18: Precision, recall, and F1 score achieved by entailment classifier trained on the training split of PPDBSICK and tested on the test split.

Table 20 shows some examples of common and interesting error cases taken from the classifier evaluated on PPDBSICK. A complete confusion matrix is given in the previous section in Figure 6. The majority of errors (26%) come from confusing the Otherwise Related ($\sim$) class with the Unrelated ($\not\sim$) class. This mistake is not too concerning from an RTE perspective since both are subtypes of the more general Independence (#) relation (Sec-

	Freq.	Precision	Recall	F score
$\not\sim$	52%	0.790	0.870	0.828
$\equiv$	7%	0.536	0.629	0.579
$\sqsupset$	20%	0.610	0.597	0.603
$\dashv$	3%	0.421	0.289	0.343
$\sim$	18%	0.492	0.353	0.411

Table 19: Precision, recall, and F1 achieved by entailment classifier trained on the training split of PPDBRTE2 and tested on the test split.

tion 2.4), and frequently involve pairs for which w_1 and w_2 belong to different semantic types.[5] There are very few cases in which the classifier makes extreme errors, e.g. confusing Equivalence ($\equiv$) with Exclusion ($\dashv$). Some interesting examples of such errors arise when the phrases contain pronouns (e.g. *"girl"* $\equiv$ *"she"*) or when the relation uses an infrequent word sense (e.g. *"photo"* $\equiv$ *"still"*).

True	Pred.	N	Example misclassifications
$\sim$	$\not\sim$	169	boy/little, an empty/the air
$\not\sim$	$\sim$	114	little/toy, color/hair
$\sqsupset$	$\sim$	108	drink/juice, ocean/surf
$\sqsupset$	$\not\sim$	97	in front of/the face of, vehicle/horse
$\sqsupset$	$\equiv$	83	cat/kitten, pavement/sidewalk
$\equiv$	$\sqsupset$	46	big/grand, a girl/a young lady
$\sqsupset$	$\dashv$	29	kid/teenager, no small/a large
$\dashv$	$\sqsupset$	29	old man/young man, a car/a window
$\not\sim$	$\equiv$	15	a person/one, a crowd/a large
$\equiv$	$\not\sim$	9	he is/man is, photo/still
$\equiv$	$\dashv$	1	girl is/she is

Table 20: Example misclassifications from some of the most frequent and most interesting error categories.

3.3.2. Labeling All Paraphrase Pairs in PPDB

We next measure the classifier's performance when assigning basic relations to arbitrary paraphrase pairs from PPDB. We train our classifier on the combination of all of our annotated datasets described in Section 3.1.3: PPDBSICK, PPDBRTE, and PPDBSAMPLE. We then run the trained model over the entire set of word and phrase pairs in PPDB (we

[5]While PPDB attempts to only extract pairs belonging to the same syntactic category, the process for doing so is noisy and thus many errors exist in the database.

leave out the syntactic paraphrase templates, described in Section 2.2.5). We associate every paraphrase pair in the database with a predicted probability distribution over the 5 entailment relations in the classifier's output label set ($\equiv$, $\sqsubset$, $\dashv$, $\sim$, $\not\sim$). We assign each pair to the basic relation that receives the highest probability according to the classifier. To handle the directionality of the $\sqsubset$ relation, we run the classifier over every pair in both directions, and we choose whichever direction ($\sqsubset$ or $\sqsupset$) receives a higher confidence score to be the final prediction.

To evaluate these predicted labels, we take a random sample of 1,000 of the pairs that the model assigned to each relation. We take a stratified sample across confidence levels (i.e the model's predicted probability for the assigned relation). Specifically, for each relation, we take all the pairs to which the classifier assigned that relation and sort this list on the classifier's confidence in the prediction. We then sample 250 pairs from the top $1/10^{th}$ of the list, 250 from the top $1/4^{th}$, 250 from the top half of the list, and 250 from the entire list. In the case of Exclusion ($\dashv$), since only 430 pairs across all of PPDB were assigned to this relation, we take the entire list. We gather labels on MTurk as described in Section 3.1.2 in order to compute accuracy.

Table 21 shows the precisions for each relation at varying levels of confidence. The classifier's results are very good for the $\equiv$ and $\dashv$ classes. The performance is lower for the $\sqsubset$ relation. Most of these errors come from misclassifying $\equiv$ as $\sqsubset$.

Predicted	N	Top 10%	Top 25%	Top 50%	All Pairs
$\not\sim$	13.M	0.42	0.44	0.40	0.34
$\equiv$	3.1M	0.89	0.74	0.73	0.67
$\sqsubset$	6.4M	0.40	0.32	0.30	0.17
$\dashv$	430	1.00	0.90	0.84	0.82
$\sim$	1.2M	0.29	0.24	0.24	0.20

Table 21: Precision of each predicted class, at varying confidence cutoffs, for all 24M word and phrase pairs in PPDB.

3.4. Using Lexical Entailment Classifier to Improve End-to-End RTE

The original goal of classifying the phrase pairs in PPDB according to our basic entailment relations was to transform PPDB into a useful lexical entailment resource, comparable to hand-built resources like WordNet, but constructed completely automatically. We now test whether we have achieved that goal by comparing the lexical entailment relations added to PPDB with those available in WordNet, in the context of an end-to-end system for RTE. Specifically, our experiments use the Nutcracker RTE System (Bjerva et al. (2014)) on the SICK dataset (Marelli et al. (2014b)). Recall that, in the RTE task, a system receives as input a premise/hypothesis pair p/h and returns one of three classifications describing the relationship between p and h: ENTAILMENT, CONTRADICTION, or UNKNOWN (Section 2.1.2).

3.4.1. The Nutcracker RTE System

We run our experiments using Nutcracker, a state-of-the-art RTE system based on formal semantics developed by Bjerva et al. (2014). In the SemEval 2014 RTE challenge, this system performed in the top 5 out of the more than 20 participating systems (Marelli et al. (2014a)). Given a premise/hypothesis (p/h) pair, Nutcracker (NC) uses the Boxer semantic parser (Bos (2008)) to produce a formal semantic representation of both p and h. These formal semantic representations are translated deterministically into standard first-order logic formulae, which are passed to both an off-the-shelf theorem prover and an off-the-shelf model builder. The theorem prover searches for a logical entailment while the model builder searches for a logical contradiction. If an entailment is found, the system predicts ENTAILMENT, and if a contradiction is found, the system predicts CONTRADICTION. When the system fails to find a proof for either entailment or contradiction, it predicts UNKNOWN.

Use of Lexical Entailment Information. NC can incorporate information from external lexical entailment resources in the form of logical axioms which are given as additional input to the theorem prover and model builder. NC uses three types of background knowledge axioms: *syn* axioms of the form $x \Leftrightarrow y$, *isa* axioms of the form $x \Rightarrow y$, and *isnota*

axioms of the form $x \Rightarrow \neg y$. Without access to an external resource providing knowledge of lexical entailments (i.e. providing these background knowledge axioms), NC's theorem prover and model builder are only capable of connecting symbols in p to symbols in h when the symbols are identical. By default, NC uses WordNet synonyms, hypernyms, and antonyms as a source of *syn*, *isa*, and *isnota* axioms, respectively.

Configuration. We run NC using the Paradox model builder (Claessen and Sorensson (2003)) and the Vampire theorem prover (Riazanov and Voronkov (2002)). The configuration of NC used to produce the results reported in Marelli et al. (2014a) includes a paraphrasing preprocessing step which substitutes words and phrases from p and h with possible paraphrases according to PPDB before any semantic parsing in performed. We remove this step before running our experiments, since the use of PPDB at this point interferes with our ability to isolate the effect of our entailment annotations on the end-to-end performance of the system. As a result, the numbers we report here differ slightly from the state-of-the-art performance reported for Nutcracker elsewhere in the literature.

3.4.2. Experimental Setup

We evaluate the performance of Nutcracker when using several external lexical entailment resources as a source of *syn*, *isa*, and *isnota* background knowledge axioms. Our baseline systems and the different lexical entailment resources are described below.

Most Frequent Class (MFC) Baseline. The most frequent class baseline is obtained by labeling every sentence pair as UNKNOWN, and results in an accuracy of 56%.

NC+$\emptyset$ Baseline. The NC+$\emptyset$ baseline is obtained by running NC alone, without any source of background knowledge axioms. In this case, words are only equivalent to the theorem prover if they are lemma-identical, and contradictions can only arise through explicit negation: i.e. *"no"* or *"not"* inserted in front of otherwise identical words.

NC+WN Baseline. The NC+WN baseline is obtained by using WordNet as a source of background knowledge. This is the default used by NC. We generate *syn*, *isnota*, and *isa* axioms respectively for each of the synonym pairs, antonym pairs, and hypernym pairs in WordNet.

NC+PPDB-XL Baseline. To differentiate between the benefits added by our entailment classifier versus those added by PPDB alone, we test an NC+PPDB-XL baseline, which uses PPDB-XL as a source of background knowledge axioms. This baseline is obtained by generating a *syn* axiom for every phrase pair in PPDB-XL. This baseline does not have any *isa* or *isnota* axioms. We tested similar baselines for all six sizes of PPDB, but XL performed best.

NC+PPDB⋆. We convert our classifier's predictions into a set of axioms for NC. When our classifier predicts $\equiv$ we generate an *syn* axiom, when it predicts $\sqsupset$ we generate an *isa* axiom, and when it predicts $\dashv$ we generate an *isnota* axiom. The Independence relations $\not\sim$ and $\sim$ do not generate any axioms. To handle the directionality of the $\sqsupset$ relation, we run the classifier over every pair in both directions, and we choose whichever direction and relation receives the highest confidence score to be the final prediction. We refer to this set of automatically-predicted axioms as PPDB⋆.

NC+PPDB-Human Oracle. To further calibrate our improvements, we also generate axioms using the human labels collected from MTurk. Our process for doing so is the same as we used to generate PPDB⋆. We refer to this lexical entailment resource as PPDB-Human.

3.4.3. Results

Table 22 reports NC's overall prediction accuracy and the number of proofs found using each of the described sources of background knowledge axioms. Using PPDB⋆, NC is able to find proofs for 25% of the p/h pairs, a substantial increase over WordNet, with which NC was able to find proofs only 21% of the time. These additional proofs lead NC to make a greater

number of correct predictions for the "right reasons" (i.e. finding a proof/contradiction) rather than by lucky guessing (recall NC guesses the most frequent class when it cannot find a proof).

	Acc.	# Proofs	Coverage
MFC	56.4	0	0%
NC + $\emptyset$	74.3	878	17.8%
NC + WN	77.5	1,051	21.3%
NC + PPDB-XL	77.5	1,091	22.1%
NC + PPDB$\star$	78.0	1,197	24.3%
NC + WN + PPDB$\star$	**78.4**	**1,230**	**25.0%**
NC + WN + PPDB-H	78.6	1,232	25.0%

Table 22: Nutcracker's overall system accuracy and proof coverage when using different sources of lexical entailment axioms.

Table 23 shows the performance in terms of the precision and recall achieved for each of the three output classes. The automatically predicted entailment relations in PPDB$\star$ contain more noise than the entailment relations provided by the manually-constructed WordNet. However, PPDB$\star$ makes up for the drop in precision with significantly improved recall: e.g. on the ENTAILMENT class, NC achieves 51% recall when using PPDB$\star$, compared to only 44% when using WordNet. Moreover, using PPDB$\star$, NC comes very close to the performance achieved when using PPDB-Human, demonstrating that the automatically generated PPDB$\star$ provides as much utility to the end-to-end system as does a gold-standard resource.

	ENTAILMENT			CONTRADICTION			NEUTRAL		
	P	R	F1	P	R	F1	P	R	F1
MFC	0.00	0.00	0.00	0.00	0.00	0.00	0.56	1.00	0.72
$\emptyset$	**0.99**	0.33	0.49	**0.99**	0.57	0.72	0.69	**0.99**	0.82
WN	0.99	0.44	0.61	0.99	0.58	0.73	0.72	0.99	0.83
PPDB-XL	0.96	0.45	0.61	0.98	0.58	0.73	0.72	0.99	0.83
PPDB$\star$	0.92	0.49	0.64	0.97	0.59	0.73	0.73	0.97	0.83
WN+PPDB$\star$	0.91	**0.51**	**0.66**	0.97	**0.59**	**0.74**	**0.73**	0.97	**0.84**
WN+PPDB-H	0.92	0.50	0.65	0.97	0.59	0.73	0.73	0.98	0.84

Table 23: Precision, recall, and F1 measures achieved by Nutcracker on SICK test data when using different sources of lexical entailment axioms.

3.5. Discussion

The goal of this chapter was to build a lexical entailment resource entirely automatically, but with precision and interpretability comparable to that offered by automatically constructed resources. We did this by assigning basic entailment relations, based on those defined in natural logic, to each of the paraphrase rules in the Paraphrase Database. Our results demonstrate that by combining a variety of signals of semantic relatedness–including signals derived from both monolingual and bilingual corpora–we are able to build a classifier for distinguishing these fine-grained relations with high accuracy. We demonstrated that the resulting, automatically-annotated PPDB improves the performance of an end-to-end RTE system by offering greater coverage of lexical entailment relations than manually-built resources like WordNet.

In constructing the above resource, we made multiple significant assumptions. Notably, we ignored the important issues of word sense and context when assigning basic entailment relations. That is, we assumed that words only have one sense, and that there is one relation that holds between x and y which is applicable in every context in which x or y appears. These assumptions did not significantly hinder the practical utility of the lexical entailment resource we constructed, evidenced by the experimental results in Section 3.4. However, this is likely due in large part to the nature of the task and the particular dataset on which we evaluate. The SICK dataset, in particular, consists of a very simple vocabulary for which the one-sense-per-word assumption is generally valid, and the majority of the words in the dataset are concrete nouns or action verbs for which there is one clear prevailing sense. For example, while there are technically multiple senses of *"man"* or *"bird"*, they are indisputably obscure and esoteric compared to the dominant sense, and thus unlikely to substantively taint our human annotation or our classifier's feature extraction. For language understanding more generally, however, these assumptions will not always hold. Thus, in moving toward more general applicability to NLU tasks, it would be necessary to incorporate models of word sense into the resource itself, or to incorporate a component for

addressing issues of context at runtime, or a combination of both.

Another important and limiting assumption that we made in this chapter was that of non-compositionality. That is, we assumed that all of the natural language expressions in PPDB could be treated as atomic units of meaning and we reasoned about the semantics of each expression by looking, for example, at the distributional context of the expression or at the ways in which the expression is translated. Many of the phrases in the database, however, are multiword phrases, e.g. *"little boy"*, *"radioactive materials"*, and *"proper operation of the internal market"*. Modeling these longer phrases in the same way as we do single words results in a number of weaknesses. First, there are infinitely many possible natural language expressions, and it is impossible to learn and store all pairwise combinations and the basic entailment relations that relate them. Second, as expressions become longer, the probability that we will observe them as-is in a corpus, and thus extract good features, becomes increasingly low: while we can probably build a good model of the distributional context of *"operation"*, building one for *"proper operation of the internal market"* is likely to be more difficult. Finally, non-compositionality limits generalizability. Our model must separately learn *"little boy"*$\sqsubset$*"boy"*, *"little girl"*$\sqsubset$*"girl"*, *"little baby"*$\sqsubset$*"baby"*, etc. Compositional models would allow us learn just one representation for *"little"*, which could be used to infer each of these pairwise relations. We address the issue of compositionality, especially in the case of modifier-noun compounds, at length in Chapters 4 and 5.

CHAPTER 4 : Semantic Containment in Compositional Noun Phrases

Among the most powerful properties of language is its compositionality. Compositionality makes it possible for us to understand an infinite number of sentences by deriving the meaning of the whole from the meaning of the parts. The remainder of this thesis will focus on one particular type of composition in the English language: combining modifiers with nouns. Modifier-noun composition enables us to make sense of phrases we have never encountered before. For example, we can easily conceptualize a *"purple cat"*, even if we have never actually seen a purple cat, because we understand the meaning of *"purple"* and of *"cat"*, and because we are generally systematic about what it means to describe an animal with a color.

In this chapter, we will focus specifically on *semantic containment* as it relates to modifier-noun composition. That is, the central question in this chapter is: how do we–and the systems we build–decide whether a *"purple cat"* is a *"cat"*? What about a *"toy cat"* or an *"imaginary cat"*? The ways in which people reason about compositional entailment, even in the simple case of noun-phrase modification, are complex and varied. Often, humans draw on common sense knowledge and contextual cues, rather than strict linguistic or logical reasoning, in order to make decisions about entailment. Thus, for automatic systems aiming to emulate human inference, it is necessary to adopt similarly informal approaches. However, the processes which govern these types of inferences are not yet well understood, preventing NLP researchers from building systems that can handle such inferences robustly.

In this chapter, we deepen our understanding of modifier-noun composition by analyzing human inferences and by assessing the competency of current NLP systems to make human-like judgements about modifier-noun compounds. In Section 4.1, we describe our methodology for analyzing modifier-noun composition in context, using the concept of atomic edits as defined by MacCartney (2009). Section 4.2 describes our annotation and the resulting annotated dataset, which we use for our analyses. Section 4.3 presents experimental results and

analysis of human inferences regarding modifier-noun composition, and Section 4.4 looks specifically at inferences involving non-subsective modifiers (e.g. *"fake"* and *"imaginary"*). Finally, in Section 4.5, we evaluate the performance of state-of-the-art RTE systems on a simplified RTE task designed to isolate the phenomena associated with modifier-noun composition, and discuss the limitations of current NLP approaches to natural language inference. Section 4.6 concludes with a discussion of practical and theoretical implications and directions for future work.

4.1. Annotating Compositional Noun Phrases in Context

In this section, we describe our methodology for annotating and analyzing modifier-noun compositions. Our focus is on characterizing modifier-noun (MH) compounds in a way that promotes better natural language inference by automatic systems. As discussed in Section 2.3, many factors contribute to the interpretation of an MH, including context, common sense assumptions, and cultural conventions. Rather than attempt to control for these confounding factors, we choose instead to embrace them and treat them as inseparable from the MH composition itself.

4.1.1. Focusing on Denotations vs. Focusing on Inferences

As we discussed in Section 2.3.2, there are two broad approaches to the study of natural language semantics. The first, commonly taken in linguistics, aims primarily to model the underlying denotations of words and phrases: e.g. where does the set of *"imaginary cats"* stand in relation to the set of *"cats"*? The second approach, predominate in NLP, aims primarily to make correct inferences about natural language statements. This latter approach is agnostic about the underlying representation of individual words beyond what is necessary to produce the right behavior in a given situation or on a given task. That is, the main concern from the point of view of an NLP system is not whether the set of *"imaginary cats"* is a subset of the set of *"cats"*, but rather: can we infer that a particular mention of *"cat"* is an *"imaginary cat"*? Or, relatedly, if we replace the phrase *"imaginary*

cat" with *"cat"* in a particular context, will it change the meaning of the utterance?

In this thesis, we adopt this inference-focused approach. As a result, in our experimental design, rather then asking humans "Is any/every instance of MH an instance of H?" we instead ask "Is this statement that is true of MH also true of H?" We accept that this design openly conflates semantic inference with pragmatic reasoning, and that it prevents us from drawing conclusions about the underlying set theoretic relationship between the denotation of MH and that of H. However, the benefit is that it enables us to explore the types of inferences that automatic systems will be expected to make in the "real world".

4.1.2. Studying Composition through Atomic Edits

Our goal is to determine which of our five basic entailment relations, as defined in Section 2.4, is generated by composing M with H. To do this, we want to design a task for studying modifier-noun composition that is as simple as possible, while still capturing realistic complexities that exist in natural language inference. To the extent possible, we would like to isolate the effect of the modifier-noun composition on the meaning of the noun phrase. However, we want to avoid collecting annotations in the "laboratory" setting, for example by studying MH pairs out of context, or in contrived, overly-simplistic sentences (e.g. *"Fido is a dog"*). Our intention is to design a task that is not unnaturally easier or unnaturally harder than what is found in the real world. Thus, if humans exploit context in order to make inferences that may not be explicitly justified by formal reasoning, our automatic systems should learn to do the same.

We define a simplified RTE task, which is identical to the standard RTE task (Section 2.1.2) but has the additional constraint that p and h differ only by the insertion of a single modifier. Specifically, if $p = s$, then $h = e(s)$ where $e = INS(M)$ and M is a single modifier. To determine the relation generated by the modifier-noun composition for a given sentence s, we must determine whether s entails or contradicts $e(s)$ and similarly whether $e(s)$ entails or contradicts s. We use the three-way entailment classification as in Section

2.1.2. That is, given a p/h pair, we must determine which of the three relationships holds: $p \Rightarrow h$ (ENTAILMENT), $p \Rightarrow \neg h$ (CONTRADICTION), or $p \not\Rightarrow h$ (UNKNOWN). By determining the classification in both the forward ($s \rightarrow e(s)$) and the reverse ($e(s) \rightarrow s$) directions, we are able to determine which of the five basic entailment relations is generated by the insertion of the modifier in the chosen context (Table 24).

Equivalence	$\equiv$	$s \Rightarrow e(s)$	$e(s) \Rightarrow s$
Forward Entailment	$\sqsubset$	$s \Rightarrow e(s)$	$e(s) \not\Rightarrow s$
Reverse Entailment	$\sqsupset$	$s \not\Rightarrow e(s)$	$e(s) \Rightarrow s$
Independence	$\#$	$s \not\Rightarrow e(s)$	$e(s) \not\Rightarrow s$
Exclusion	$\dashv$	$s \Rightarrow \neg e(s)$	$e(s) \Rightarrow \neg s$

Table 24: Inference conditions used to determine which of the basic entailment relations is generated by the composition of M with H.

For example, if $s =$ *"She wore a dress"* and $e = INS($*"red"*$)$ then $e(s) =$ *"She wore a red dress"*. In this case, since $s \not\Rightarrow e(s)$ and $e(s) \Rightarrow s$, we can determine that $\beta(e)$ is $\sqsupset$.

4.1.3. Limitations of our Methodology

In the above-described simplified RTE task, we assume that the entailment relation that holds overall between s and $e(s)$ is attributable wholly to the atomic edit (i.e. the inserted modifier). This is an over-simplification. In practice, several factors can cause the entailment relation that holds between the sentences overall to differ from the relation that is generated by the $INS(M)$ edit. For example, negation, quantifiers, or other downward-monotone operators can block or reverse entailments (*"brown dog"* entails *"dog"*, but *"no brown dog"* does not entail *"no dog"*). We make an effort to avoid selecting such sentences for our analysis (Section 4.2.1), but fully identifying and handling such cases is beyond the scope of this thesis. We acknowledge that downward monotone operators and other complicating factors (e.g. multiword expressions) are present in our data. However, based on manual inspection, they do not occur frequently enough to substantially effect our analyses.

4.1.4. Treating Entailment as a Continuum

Very often, humans draw conclusions about natural language based on "assumptions that seem plausible, rather than assumptions that are known to be true" (Kadmon (2001)). For example, given s and $e(s)$ below, most readers would agree that, while it cannot be *guaranteed* that $s \Rightarrow e(s)$, it seems artificially naive to say $s \not\Rightarrow e(s)$

s: A cat sitting on the ground looks out through a clear door screen.

$e(s)$: A domestic cat sitting on the ground looks out through a clear door screen.

While RTE has thus far always been treated as a discrete classification task by the NLP community (Section 2.1.2), systems are increasingly expected to make informal and probabilistic inferences like the one above (see Table 4 in Section 2.1.3). There is thus a strong case for treating entailment as a continuum rather than as a discrete classification. Doing so provides a clearer treatment for "edge case" inferences and is arguably better aligned with the way humans reason about language.

Therefore, when collecting humans annotations for the simplified RTE task just described in Section 4.1.2, we replace the hard three-way classification (ENTAILMENT, CONTRADICTION, or UNKNOWN) with a softer 5-point scale in which 1 corresponds to definite CONTRADICTION, 3 corresponds to UNKNOWN, and 5 corresponds to definite ENTAILMENT, but scores of 2 and 4 allow humans to specify likely (but not certain) contradiction and entailment, respectively. Allowing for weak judgments of probable entailments and contradictions allows us to more naturally capture inferences like that *"cat"* very likely entails *"domestic cat"* in the example above. When necessary, for example to interface with existing RTE systems, we collapse this 5-point scale to the standard three-way classification.

4.2. Labeled Datasets for Analysis

Using the above-described methodology, we collect a large dataset of human judgements about modifier-noun composition. This dataset is used for our analyses in Section 4.3.

4.2.1. Data Selection

In order to collect annotations using our simplified RTE task, we must select 1) a set of modifier-noun pairs to study and 2) a set of contexts in which to display each of the chosen modifier-noun pairs. Our process for selecting this data is described below. More information about our selection strategies, as well as a comparison with alternative strategies, is given in Appendix A.4.

We look at four different corpora capturing four different genres: the Annotated Gigaword corpus from Napoles et al. (2012) (henceforth referred to as News), a collection of crowd-sourced image captions from Young et al. (2014) (Image Captions), the Internet Argument Corpus from Walker et al. (2012) (Forums), and the prose fiction subset of GutenTag dataset from Brooke et al. (2015) (Literature).

Choosing Modifier-Noun Pairs

We collect annotations for the most frequent modifiers of the most frequent nouns. Specifically, from each corpus, we select the 100 nouns which occur with the largest number of unique adjectives. Then, for each of these nouns, we select the 10 modifiers with which the noun occurs most often. A pilot study revealed that focusing on frequent MHs did not introduce any significant bias, and that MHs from the tail of the distribution do not behave notably differently than those from the head (see Appendix A.4.1).

We process the sentences in each corpus using the Stanford CoreNLP POS tagger and dependency parser (Manning et al. (2014)). We look only at adjectival modifiers (JJs) and common nouns (NNs). We consider only modifier-noun pairs in which 1) the JJ occurs immediately before the NN, 2) the JJ is linked via an *amod* dependency relation to the NN, and 3) the NN is not a modifier of (in an *nmod* dependency relation with) any other word. Table 25 shows a random sample from modifier-noun pairs chosen by our method.

Image Captions asian man · beige vest · black collar · black dog · blond girl · blue ball · blue slide · blue sweater · blue wall · blurry background · brown coat · brown ground · brown horse · colorful clothing · colorful tent · colorful truck · curly hair · dirty snow · empty beach · goofy face · green bench · green chair · little child · long hair · long view · modern setting · muddy track · multiracial couple · red bench · red plane · red vehicle · red vest · red wall · shirtless man · small hill · small tree · small yard · snowy area · snowy grass · striped top · striped vest · tall cliff · white dog · white toy · wooded area · wooded hill · yellow bird · yellow coat · yellow grass · young sheep
News accepted form · american history · american model · american style · annual production · criminal organization · deadly attack · documentary film · easy victory · electrical power · final game · financial section · fine form · fine line · first man · first section · first time · funeral service · genetic material · good performance · huge crowd · international trade · key piece · left side · legal work · local time · main town · major operation · mountainous region · new company · overall strategy · own set · own show · own version · palestinian man · pharmaceutical industry · political nature · pragmatic approach · prominent figure · public figure · public image · real power · second victory · socialist party · southern region · southern state · special unit · strained relationship · tarnished image · web site
Literature bad sort · black hair · bright spot · calm sea · civilized world · clever fellow · curly head · deep thought · electric light · extreme youth · fair face · fierce desire · final word · fine piece · first day · first time · generous nature · german youth · great effect · great mind · great moment · grim smile · human body · human soul · humble home · immediate action · little horse · little laugh · little thing · low laugh · much talk · naked eye · natural order · new country · new friend · new piece · new spirit · old boy · open sea · other night · other person · own person · own power · private business · quick glance · real pleasure · same sort · strange feeling · third day · whole country
Forums administrative cost · american economy · american market · bad form · basic knowledge · big money · correct answer · current conflict · different definition · economic plan · efficient manner · entire nation · federal control · final approach · financial support · first point · first thing · good idea · good leader · good life · hard work · humble opinion · individual basis · intellectual effort · last thing · legal system · limited government · low cost · moral issue · new class · only person · only problem · only reason · original post · other organization · own argument · own experience · palestenian land · palestinian society · past experience · perfect example · personal information · political position · powerful nation · public money · same language · same situation · same way · socialist agenda · whole story

Table 25: Examples of modifier-noun pairs selected from each corpus for annotation.

Choosing Contexts

For each of the MHs chosen above, we select three sentences from the corpus in which to judge the MH. Specifically, we select a sentence from the corpus in which the noun H appears unmodified–i.e. does not participate in any *amod* or *nmod* dependency relations.

A pilot study revealed that building our p/h pairs from contexts in which H appears unmodified, rather than from contexts in which MH appears natively, leads to more diverse contexts and varied annotations. A discussion of this pilot study is given in Appendix A.4.2.

Before sampling, we apply several filters in order to reduce the noise in the selected sentences. First, we use two POS taggers, the one distributed with Stanford CoreNLP (Manning et al. (2014)) and the one distributed with NLTK (Loper and Bird (2002)), and only consider sentences in which both taggers agree that H is used as a common noun (NN) in the sentence. Second, since negations invert the true entailment associated with an atomic edit (Section 4.1.3), we omit sentences containing obvious negations (i.e. *no*, *not*, *n't*) before sampling. This is admittedly a crude way of controlling for confounds resulting from monotonicity: many of the removed sentences do not negate the noun in question, and many sentences that do indeed negate the noun still pass through our filter. However, since identifying and scoping negations is a very hard problem , we resort to this simple heuristic.

When sampling, we try to prefer short sentences. This is intended to reduce the cognitive load on the annotators and simplify the task. Specifically, for a given H, we first try to sample sentences containing H (and meeting the other conditions above) that are less than 15 words long. If we are unable to select 3 sentences meeting these criteria, we try to sample from sentences less than 20 words long. We continue raising the upper limit until the sample has been filled.

4.2.2. Annotation

We recruit annotators from Amazon Mechanical Turk to participate in our study. We gather annotations via the simplified RTE task from Section 4.1.2 using the following configuration and setup.

Constructing p/h Pairs

Following the sampling procedures described above, we collect a set of 11,910 $\langle M, H, s \rangle$ tuples: 1,000 MHs from each of our four corpora and up to[6] three contexts per MH. From these $\langle M, H, s \rangle$ tuples, we generate $e(s)$, the atomic edit $e = INS(M)$ applied to s, by inserting the adjective M directly in front of H in s. We ask annotators to provide entailment judgements in both "forward" direction in which $p = s$ and $h = e(s)$, and the "reverse" direction in which $p = e(s)$ and $h = s$. In Section 4.3, we will use the combination of these two three-way judgements in order to infer the basic entailment relation generated by the modifier-noun composition.

Interface and Task Parameters

We have three independent annotators judge each p/h pair. Our interface presents each annotator with the premise p followed by the hypothesis h. Our instructions tell the annotator to assume that p "is true or describes a true scenario" and asks them to indicate, on a scale from 1 to 5, how likely it is that h "is also true or describes the same scenario." We provide several examples of p/h pairs and their expected annotations. Our exact guidelines and examples are shown in Appendix A.5. A screenshot of our annotation interface is given in Figure 7.

Annotators are paid \$0.20 to annotate a set of 10 p/h pairs, or \$0.02 per annotation. We restrict to workers who are located in the US, and have had completed at least 1,000 HITs with at an approval rate of at least 98%.

Quality Control

While we want to keep our annotation task open enough to allow for differences in interpretation, we want to ensure that workers perform the task conscientiously enough that their

[6]Since the Image Caption corpus is small, we are not able to select three sentences for every MH, and thus have a total of 2,910 sentences rather than 3,000 sentences from that corpus.

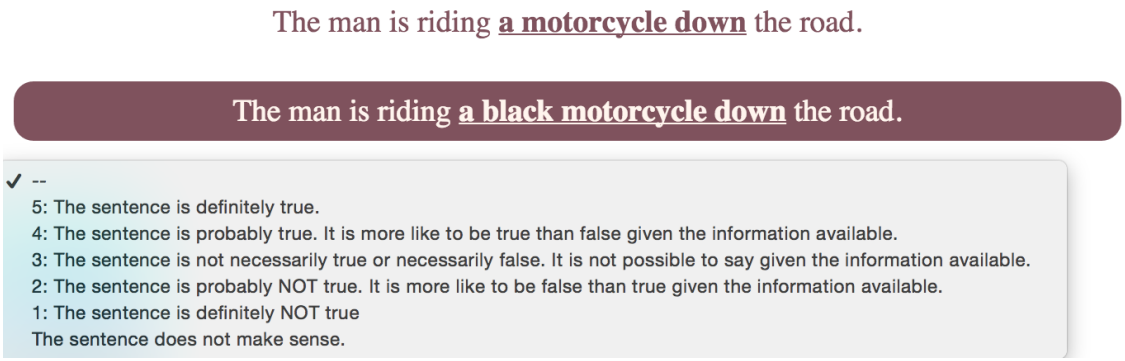


Figure 7: Annotation interface used to collect entailment judgements for p/h pairs.

work can be trusted. To do this, we embed a small number of quality control questions (one out of every 10 questions) into each task. We remove workers who fail to answer these questions correctly. We strive to keep the control questions unambiguous and as similar as possible to the examples we show in the instructions. Our intent is for the quality control questions to serve as solely an attention check rather than a filter which biases our pool of annotators toward a particular type of interpretation.

Although the aim of our annotation is to study MH composition, several of our control questions include implicative verbs (e.g. *"manage to"*, *"refuse to"*). Including implicatives allows us to construct unambiguous and fair controls for the cases where the expected rating is 1 ("p definitely entails *not* h") or 5 ("p definitely entails h"). When we attempted to build controls for these categories using MHs, the resulting examples still left room for argument (see Section 4.4 on for an in-depth discussion of why privative adjectives do not reliably result in judgements of CONTRADICTION). Some example QC questions are shown in Table 26, and a full list is given in Appendix A.5.

CONTRADICTION (1 or 2)	p: Denver went 40-42 and **failed to make** the playoffs.
	h: Denver went 40-42 and **made** the playoffs.
UNKNOWN (3)	p: Three people are pulling a **rope** on a hillside.
	h: Three people are pulling a **white rope** on a hillside.
ENTAILMENT (4 or 5)	p: Police say about 25 passengers **managed to escape**.
	h: Police say about 25 passengers **escaped**.

Table 26: Examples of some of the quality control questions embedded in our tasks.

Our policy for accepting or rejecting an annotator's judgements is as follows. We accept every annotator's first 10 tasks (equivalent to 90 real p/h pairs and 10 quality control pairs) automatically. After that, if a worker's accuracy on our controls falls below a chosen threshold τ_{reject}, we reject all of their work. If at any point worker's accuracy surpasses a chosen threshold τ_{accept}, we accept all of their future work. When a worker's accuracy is between τ_{reject} and τ_{accept}, we accept proportionally to their accuracy on the controls. We set $\tau_{reject} = 0.3$ and $\tau_{accept} = 0.7$. These thresholds were set after manually inspecting varying accuracy levels and determining levels that seemed to differentiate conscientious work from spam. However, we iterate on our controls throughout the annotation, in response to questions and feedback from workers. At times, we manually override decisions about whether or not to reject an annotator's judgements, when annotators provided rationale for their annotations. In the end, we rejected work from only 11 out of a total of 192 annotators. Work that was rejected was reposted to be completed by another annotator.

4.2.3. Filtering and Post-Processing

We aggregate annotations by taking a simple average of the three independent annotations for each p/h pair. For analyses that require categorical labels, we collapse this average entailment score such that scores less than 2.5 are considered to be CONTRADICTION, scores between 2.5 and 3.5 to be UNKNOWN, and scores greater than 3.5 to be ENTAILMENT.

Before running our analyses, we remove sentences for which one or more annotators selected the "does not make sense" option. In addition, we remove sentences in which we don't have at least two out of the three annotators in agreement on the 5-way rating. For example, if the three annotators give ratings 4, 4, and 5, we keep the pair and consider the true label to be ENTAILMENT (with score 4.33), whereas we omit a pair in which the three annotators give ratings of 3, 4, and 5. If either of our criteria is not met in either the forward ($s \rightarrow e(s)$) or the reverse ($e(s) \rightarrow s$) direction, we remove the p/h pair altogether. We admit that these criteria are quite strict, but choose to err on the side of agreement and reproducibility. In the end, our "does not make sense" filter removes nearly half of the p/h pairs, and our "2

out of 3" filter removes 10% of the remaining pairs. Thus, our final set of sentences consists of 5,560 sentence pairs, coming roughly evenly from the four corpora. Table 27 gives a breakdown of our dataset in terms of the number of p/h pairs and the number of unique MHs from each corpus. Table 28 shows examples of sentences that were removed by our filtering criteria.

Genre	p/h pairs	MHs
News	1,398	834
Literature	1,203	754
Forums	1,270	812
Image Captions	1,689	864

Table 27: Number of p/h pairs and unique MHs in our dataset coming from each corpus.

Removed by "does not make sense" filter
No **certain amount** of government regulation changes this.
Which means they can handle more than one **poor job** at a time.
The answer is that they are both **own right**.
Orlando Cutter usually drove **beautiful home** with her when the class was over.
Removed by "2 out of 3" filter
That depends on how the **judicial system** is ran. (3,4,5)
It is one of the **developing world**'s great tourist destinations. (2,3,4)
City officials hope to complete the **next project** by the end of the year. (2,3,4)
To suspect a **poor woman** is a crime in love. (1,2,3)

Table 28: Examples of sentences removed by our filtering.

4.2.4. Reproducibility

To ensure that our judgements are reproducible, we re-annotate a random 10% of our pairs, using the same annotation setup but a different set of annotators. We compute the intra-class correlation (ICC) between the scores received on the first round of annotation, and those received in the second pass. ICC is related to Pearson correlation, and is used to measure consistency among annotations when the group of annotators measuring each observation is not fixed, as opposed to metrics like Fleiss's κ which assume a fixed set of annotators. On our data, the ICC is 0.77 (95% CI 0.73 - 0.81) indicating very high agreement. These twice-annotated pairs will become our test set in Section 4.5 when we

evaluate the performance of automatic systems on this simplified RTE task.

4.3. Analysis of Human Inferences

In this section, we use the data collected in Section 4.2 to gain a better understanding of how modifier-noun composition effects the inferences humans make. In Section 4.5, we will analyze how well current state-of-the-art RTE systems align with these human inferences.

Note that, in this section and the next, I may refer to judgements and relations as holding for H and MH rather than for s and $e(s)$. This is for convenience and compactness. For example, I might say that "humans judged H to entail MH", when in fact, humans judged s, a specific sentence containing H, to entail $e(s)$, that same sentence but with M inserted in front of H. Similarly, a statement like "$H \equiv MH$" should be understood to mean that the atomic edit $INS(M)$, applied to a context containing H, generated the $\equiv$ relation.

4.3.1. Basic Entailment Relations Generated by MH Composition

We first look at the basic entailment relations that are generated when modifiers are inserted. Recall from Section 4.1.2 (Table 24) that the relation generated is determined by considering inferences in both the "forward" direction (whether s entails or contradicts $e(s)$) and the "reverse" direction (whether $e(s)$ entails or contradicts s).

Distribution of Generated Relations

Figure 8 shows the distribution of entailment judgements associated with the forward and reverse inferences for each of the four genres we study. For most modifiers in most contexts, the forward direction yields judgements of UNKNOWN (i.e. $s \not\Rightarrow e(s)$) while the reverse direction yields judgements of ENTAILMENT ($s \Rightarrow e(s)$). This is the pattern we would expect to see when modifiers are subsective and humans are reasoning consistently with standard rules of logical inference: for example, if the set of red dresses is a subset of the set of all dresses, we expect that in most sentences *"red dress"* entails *"dress"* but *"dress"* does not necessarily entail *"red dress"*. While this "subsectivity" pattern is the

dominant one, however, it does not hold in all cases. In every genre, we see a range of entailment judgements provided for both the forward and the reverse inference. We see an especially large number judgements in which the forward direction ($s \rightarrow e(s)$) is judged as ENTAILMENT. The degree of variability in judgements differs substantially across genres.

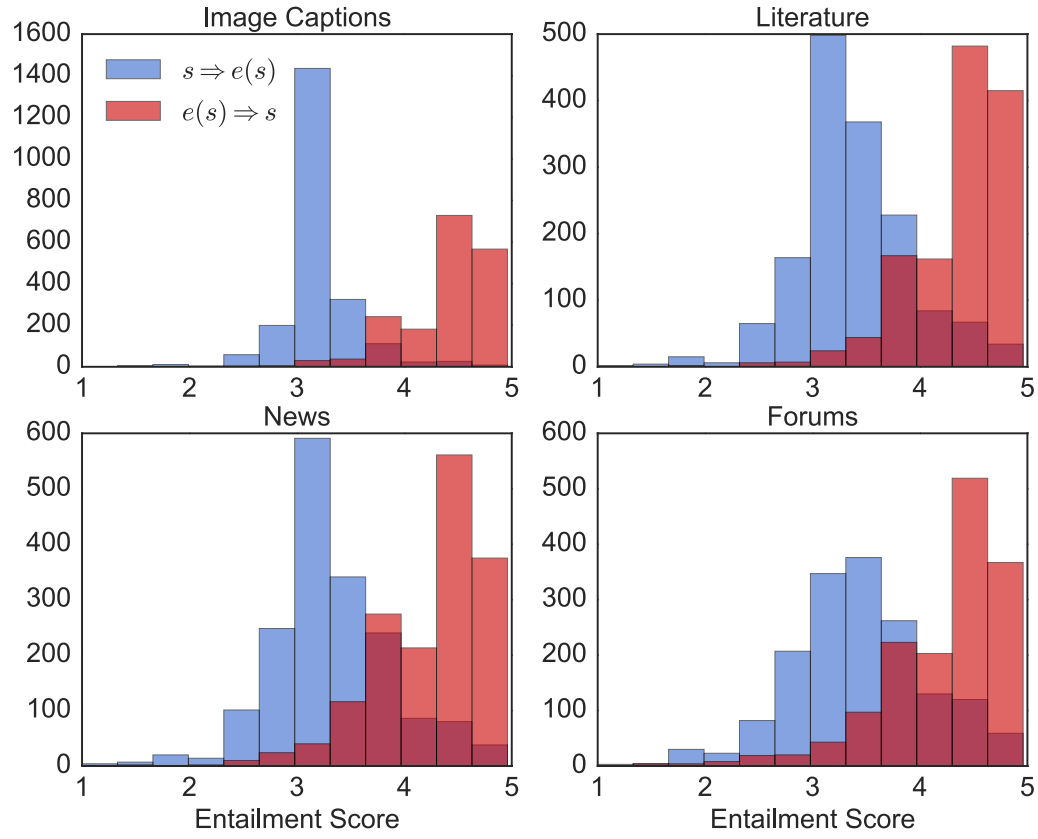


Figure 8: Distribution of human entailments judgements (on a five-point scale) for "forward" inferences ($s \rightarrow e(s)$) and for "reverse" inferences ($e(s) \rightarrow s$) where $e = \text{INS}(M)$.

Figure 9 shows the basic entailment relations generated by the modifier insertions. In Image Captions, which consist of fairly simple sentences and a very concrete vocabulary, the vast majority of modifiers (approximately 85%) generate the Reverse Entailment ($\sqsupset$) relation when inserted. However, in genres with more complex language, modifiers generate a wider range of relations. In Forums, for example, 36% modifier-noun compositions were judged as generating the Equivalence ($\equiv$) relation, indicating that inserting the modifier did not add new information beyond what was already entailed when the sentence contained the

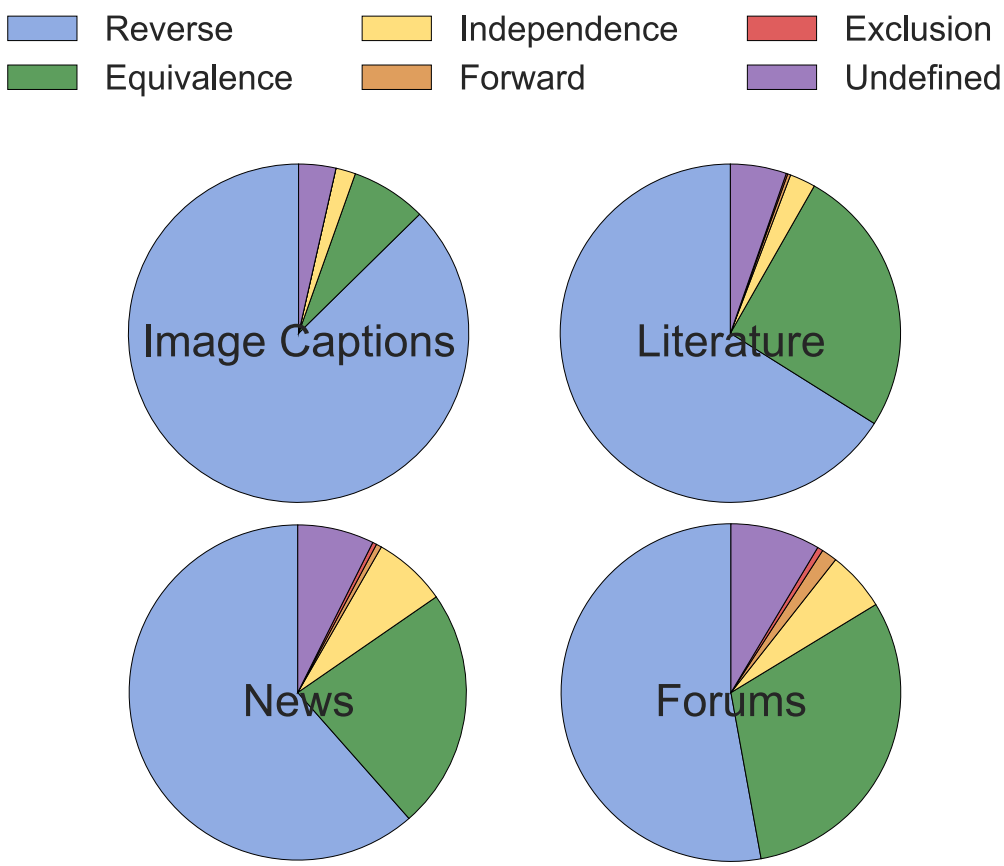


Figure 9: Basic entailment relations generated by INS(M) edits across four genres.

noun unmodified. Table 29 shows examples of MHs and contexts in which each of the basic entailment relations is generated. Some entailment inferences depend entirely on contextual information (Example 1) while others arise from common-sense inference (Example 2).

Examples in which the Independence (#) relation is generated are especially interesting. Recall from Section 2.3.2 that Independence, according to the set-theoretic definitions, should cover MHs such as *"alleged criminal"*, in which the MH may or may not entail the H and vice-versa. In practice, the cases we observe which generate the Independence relation tend to be those in which the unmodified noun has a particularly strong default interpretation. For example, in Example 5 in Table 29, *"local economy"* is considered to be independent of *"economy"* when used in the context of *"President Obama"*: i.e. the assumption that the president would be discussing the national economy is so strong that

1.	$MH \equiv H$	The **deadly attack** killed at least 12 civilians.
2.	$MH \equiv H$	The **entire bill** is now subject to approval by the parliament.
3.	$MH \sqsubset H$	He underwent a **successful operation** on his leg.
4.	$MH \sqsubset H$	From those surveyed, 255 were selected for the **informal meeting.**
5.	$MH \# H$	Obama cited the data as evidence that the **local economy** is improving.
6.	$MH \# H$	Some went for the history and **political culture**.
7.	$MH \sqsupset H$	The **militant movement** was crushed by the People's Liberation Army.
8.	$MH \sqsupset H$	There are questions as to whether our **traditional culture** has changed.
9.	$MH \dashv H$	Red numbers spelled out their **perfect record**: 9-2.
10.	$MH \dashv H$	Schilling stayed busy after serving Epstein turkey at his **former home**.

Table 29: Examples of different types of basic entailment relations ($\beta(e)$) generated by inserting a modifier in front of a noun ($e = \text{INS}(M)$).

even when the president says *"the local economy is improving"*, people do not understand it to mean that he has said *"the economy is improving"* more generally. Similarly in Example 6, *"culture"* seems to carry such a strong default meaning on its own that, in context, it is not seen as a hypernym of *"political culture"* but rather the two phrases are interpreted as referring to independent concepts.

Context Sensitivity

The examples in Table 29 illustrate that at least some judgements of whether H entails MH depend on the context s itself, and cannot be determined solely by the modifier and the noun being composed. Specifically, there are 480 unique modifiers which appear in our dataset as modifiers of at least two unique nouns. Of these modifiers, 62% generate different entailment relations depending on the noun in front of which they are inserted. Furthermore, there are 1,215 unique modifier-noun pairs in our dataset which were judged in two or more contexts. Of these, 32% are observed generating different entailment relations depending on the context in which they are composed. Note that these percentages are specific to our sampling methods, and are not necessarily indicative of how modifiers and nouns behave in general in natural language. Nonetheless, these figures indicate that entailment properties of MHs can be highly context dependent. Table 30 provides examples of how the same MH composition can lead to different entailment judgments, depending on context.

MH	Inference	Context
enthusiastic crowd	$H \Rightarrow MH$	The **crowd** roared.
enthusiastic crowd	$H \not\Rightarrow MH$	I look around at the **crowd**.
ridiculous claim	$H \Rightarrow MH$	This **claim** is a lie.
ridiculous claim	$H \not\Rightarrow MH$	Freshman level astronomy takes care of this **claim**.
ridiculous claim	$H \Rightarrow \neg MH$	It seems important to you that I support my **claim**.

Table 30: Examples when composing the same modifier M with the same noun H generates different entailment relations depending on context.

4.3.2. Generalizations for when H entails MH

The prevalence of the Equivalence ($\equiv$) relation in Figure 9 reveals that, at times, it is possible to modify a noun without actually adding any new information beyond what was already communicated by the noun alone. Sometimes, this occurs because the sentence in which the noun appears entails the modifier by definition. Example 1 in Table 29 provides one such example. However, in other cases, the context does not explicitly justify the inference of MH from H. Among such cases, a few patterns stand out which appear to hold in general, independent of the particular context in which the MH appears.

Communicating Presence vs. Absence Our data suggests that, in general, nouns are assumed to be present, salient, and relevant. As a result, modifiers that communicate presence and saliency tend to be entailed, regardless of the noun with which they are being composed or the context in which it appears, while modifiers that communicate absence or irrelevance tend to generate contradictions. Figure 10 shows, for a given modifier M, the distribution over judgments for whether H entails MH for various Hs. For example, the modifier *"false"* generates judgments of definite or likely CONTRADICTION in nearly every context in which it is judged. In contrast, the modifier *"real"* generates judgements of definite or likely ENTAILMENT in nearly every context.

Whether a modifier communicates presence or absence can be noun-dependent, when the modifier relates to the core meaning of the noun and the properties that would make that noun relevant for discussion. For example, *"answers"* are assumed (perhaps naively) to

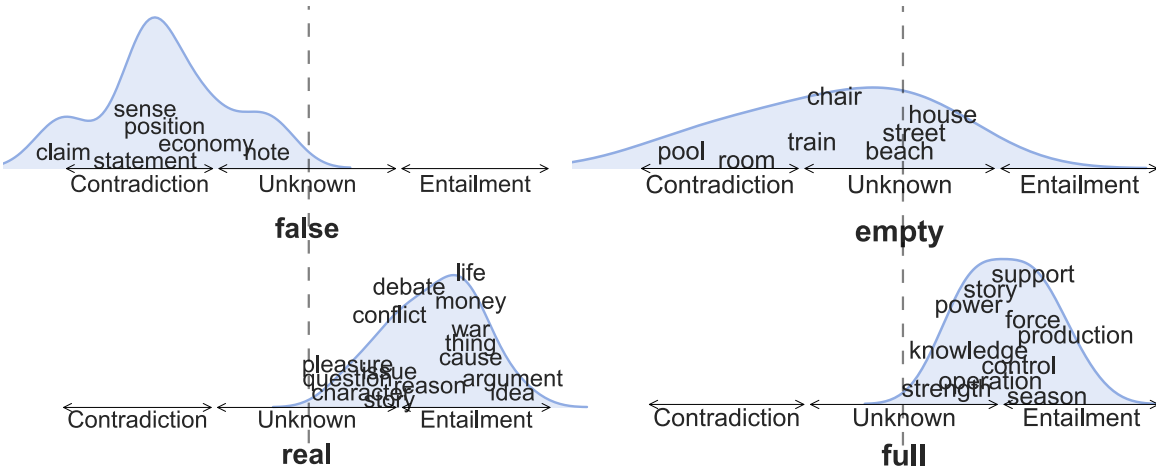


Figure 10: Distribution over entailment scores generated when composing several "presence" modifiers and several "absence" modifiers with various nouns.

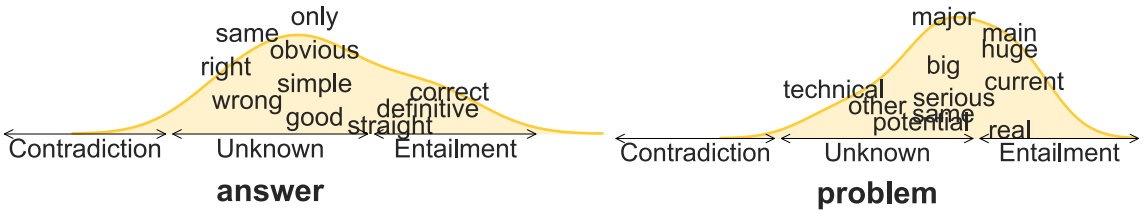


Figure 11: Unless otherwise specified, nouns are considered to be salient and relevant. *"Answers"* are assumed to be *"correct"*, and *"problems"* to be *"current"*.

be *"correct"* and *"definitive"*, and *"problems"* are assumed (perhaps melodramatically) to be *"current"* and *"huge"*. This notion of noun-specific properties is closely related to the notion of prototypicality, described below.

Communicating Prototypicality In general, we see that H is assumed to entail MH when M captures attributes of the "prototypical" instance of the H. For example, people are generally comfortable concluding that *"beach"* entails *"sandy beach"* (Figure 12) and that *"baby"* entails *"little baby"* (Figure 13). Prototype-based inferences are dependent on both M and H: i.e. same modifier may be prototypical and thus entailed in the context of one noun, but generate a contradiction when composed with a different noun. For example, if *"she has a baby"*, it is probably fine to say to infer that *"she has little baby"*, but if *"she has control"*, it would be inconsistent to infer that *"she has little control"* (Figure 13). In fact,

in this particular example, one could argue that the composition of *"little"* with *"control"* generates a contradiction because it fails the presence/absence test just described.

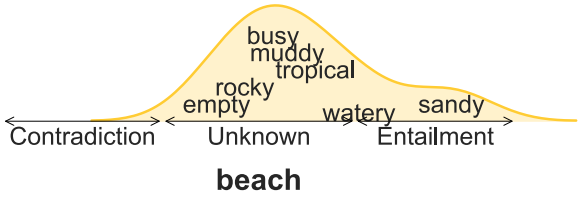


Figure 12: Distribution over entailment scores generated when composing various modifiers with the noun *"beach"*.

Figure 13: Distribution over entailment scores generated when composing the modifier *"little"* with various nouns.

4.3.3. Undefined Entailment Relations

Our annotation methodology does not ensure that all of the MH compositions will generate one of the five basic entailment relations defined in Section 2.4. In fact, for roughly 5% of our p/h pairs for which we collected annotations, the aggregated human judgements do not correspond to any well-defined set-theoretic relation (Table 31).

$s \rightarrow e(s)$	$e(s) \rightarrow s$	Frequency
CONTRADICTION	ENTAILMENT	4.26%
CONTRADICTION	UNKNOWN	0.89%
UNKNOWN	CONTRADICTION	0.36%
ENTAILMENT	CONTRADICTION	0.10%

Table 31: Frequency of p/h pairs in which human's entailment judgements result in INS(M) generating an "undefined" basic entailment relation.

Many of these cases occur infrequently and could be attributed to the imprecision of our experimental design or to human error. However, one pattern appears with high enough

frequency to warrant further investigation. Specifically, in more than 4% of our contexts, the forward inference ($s \rightarrow e(s)$) is judged as CONTRADICTION but the reverse inference ($e(s) \rightarrow s$) is judged as ENTAILMENT. If were were to try to interpret these judgments in terms of the set-theoretic relationship between the denotations of H and MH, as in the prior work described in Section 2.3, we would arrive in the (non-sensical) situation in which $([\![MH]\!] \subset [\![H]\!]) \wedge ([\![MH]\!] \cap [\![H]\!] = \emptyset)$, i.e. "every MH is an H, but no H is an MH." Since this relation occurs frequently here and in Section 4.4, we will designate it as Undefined and denote it using the $\varnothing$ symbol.

Table 32 provides examples of MHs and contexts which generate the Undefined relation. In general, these sentences capture cases in which common-sense assumptions about what is most often the case in the real world dominate the inference. For example, given the premise *"Bush travels to Michigan to remark on the economy"*, humans are confident enough that *"economy"* refers to *"American economy"* that they label the insertion of *"Japanese"* as generating a contradiction. However, when told that *"Bush travels to Michigan to remark on the Japanese economy"*, annotators have no difficulty concluding that *"Bush travels to Michigan to remark on the economy"*.

MH	Context
Japanese economy	Bush travels Monday to Michigan to remark on the **economy**.
small government	**Government** is the only thing holding back large corporations.
homeless man	A child rides on a **man**'s shoulders.

Table 32: Examples of contexts in which generate the Undefined ($\varnothing$) relation: i.e. MH was judged to entail H but H was judge to entail $\neg MH$.

The Undefined relation is particularly interesting as a case study for when people reason based on context and convention rather than logic and semantics. In Section 4.4, we will present evidence that inferences involving privative adjectives often give rise to this same pattern. We will discuss the implications more generally in our discussion in Section 4.6.

4.4. Privative and Non-Subsective Adjectives

Recall from Section 2.3.1 that formal semantic theory defines three classes of modifiers: subsective, plain non-subsective, and privative. Privative modifiers (e.g. *"fake"*) are defined as modifiers for which the set denoted by the modified noun MH is disjoint from that denoted by the unmodified noun H. Plain non-subsective modifiers (e.g. *"alleged"*) are defined as modifiers for which the denotation of MH is neither a subset of, nor disjoint from, the denotation of H.

In this section, we look specifically at the behavior of privative and plain non-subsective modifiers in terms of the inferences they do and don't permit in our simplified RTE setting. We observe that, in practice, both privative and plain non-subsective modifiers often behave differently than what we would expect based on their formal semantics definitions. In particular, plain non-subsective modifiers tend to behave like subsective modifiers: i.e. they often generate the Reverse Entailment ($\sqsupset$) relation rather than the expected Independence (#) relation. Privative modifiers tend to lead to asymmetric entailment judgements, generating the Undefined ($\varnothing$) relation just discussed in Section 4.3.3, rather than the expected Exclusion ($\dashv$) relation.

4.4.1. Experimental Design

We begin with the set of 60 non-subsective adjectives identified by Nayak et al. (2014). We split this list into 24 privatives and 36 plain non-subsectives (Table 33). This division of the 60 adjectives into privative and plain is based on our own understanding of the literature, not on Nayak et al. (2014).

For each of the adjectives in the list, we find sentences in the Annotated Gigaword corpus (Napoles et al. (2012)) in which the adjective appears as a direct modifier of a common noun. That is, we find sentences in which the modifier appears directly to the left of, and in an *amod* dependency relation with, a noun tagged as NN. For each adjective, we choose

Plain Non-Subsective			Privative		
alleged	apparent	arguable	anti-	artificial	counterfeit
assumed	believed	debatable	deputy	erstwhile	ex-
disputed	doubtful	dubious	fabricated	fake	false
erroneous	expected	faulty	fictional	fictitious	former
future	historic	impossible	hypothetical	imaginary	mock
improbable	likely	mistaken	mythical	onetime	past
ostensible	plausible	possible	phony	pseudo-	simulated
potential	predicted	presumed	spurious	virtual	would-be
probable	proposed	putative			
questionable	seeming	so-called			
supposed	suspicious	theoretical			
uncertain	unlikely	unsuccessful			

Table 33: 60 privative and plain non-subsective adjectives from Nayak et al. (2014).

up to[7] 10 sentences, requiring that the adjective modifies a different noun in each. As a control, we take a small random sample of 100 MHs for which M does not appear in our list of privative and non-subsective modifiers. We expect these to contain almost entirely subsective adjectives. Table 34 shows a random sample of modifier-noun pairs occurring in our dataset for each class.

As before, we use the sentences to generate p/h pairs. Note that for these experiments, we select sentences from our corpus in which MH appears natively, rather than sentences in which H appears unmodified. Our rationale is that we want to look specifically at the way these modifiers are actually used in practice, and whether they substantively affect the overall meaning of the sentences in which they appear. Thus, in order to generate p/h pairs, we apply the deletion edit $e = DEL(M)$ to s. We then, as before, gather entailment judgements in two directions. For consistency with previous sections, the "forward" direction will refer to the insertion of M (i.e. $p = e(s)$ and $h = s$), and the "reverse" direction will refer to the deletion of M (i.e. $p = s$ and $h = e(s)$). For clarity, we may refer to the forward direction as "inserting" and the reverse direction as "deleting".

In total, we collect judgements for 459 p/h pairs, covering 54 of the 60 adjectives. We omit

[7] After imposing the described constraints on the sentences, not every modifier is observed with 10 unique nouns.

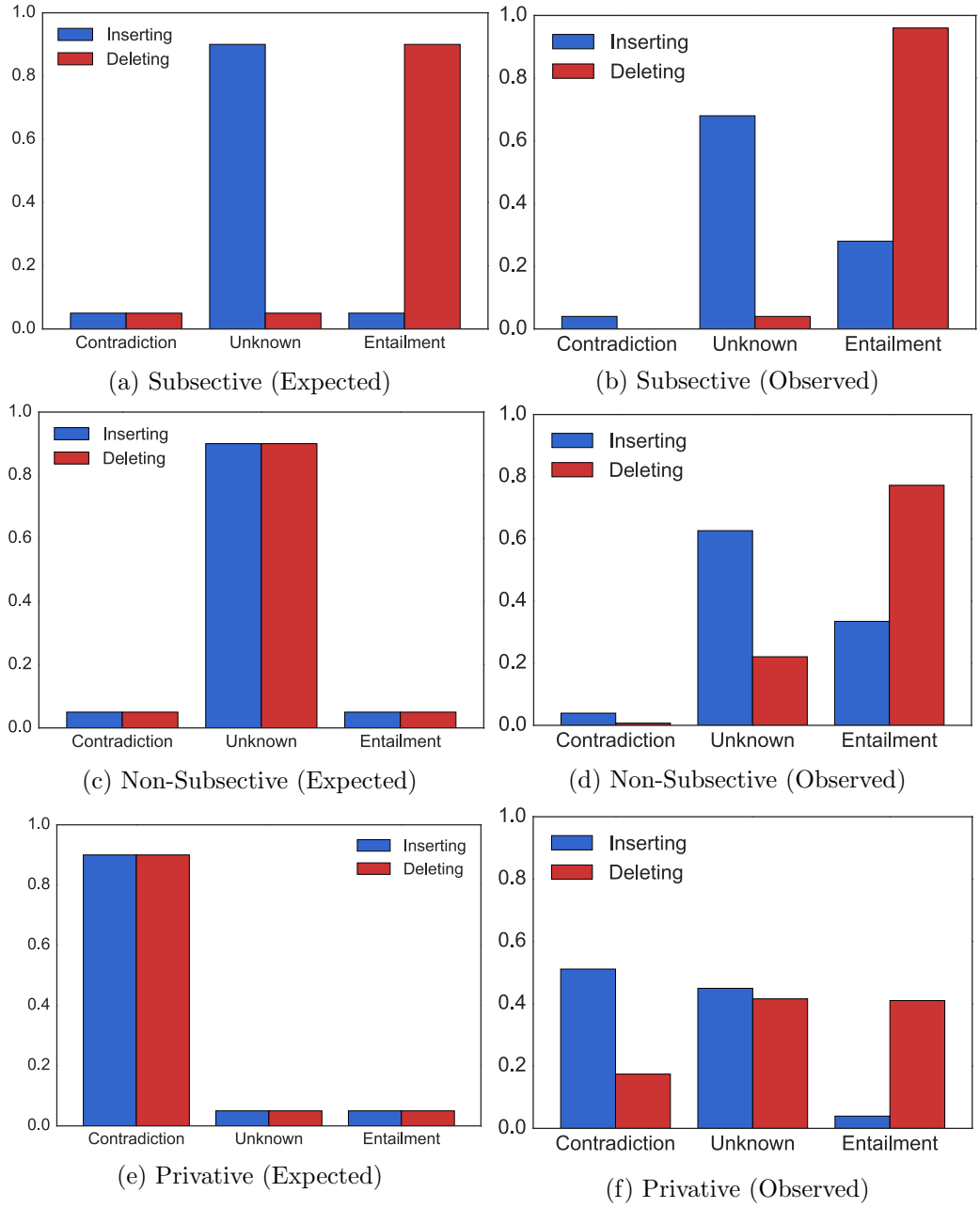
Control (Subsective) current problem · deep snow · good character · gray coat · green bag · financial position · hollow sound · immediate effect · jewish character · large piece · legal system · local economy · much power · new business · new position · old train · personal experience · personal relationship · pink shirt · pink toy · right side · significant role · third person · white boat · white track
Plain Non-Subsective alleged attempt · alleged support · apparent lack · arguable exception · doubtful proposition · dubious claim · erroneous reference · expected visit · impossible goal · impossible position · likely outcome · mistaken assumption · ostensible mission · plausible alternative · plausible deniability · plausible explanation · potential danger · potential market · probable return · proposed change · suspicious behavior · theoretical possibility · theoretical risk · unlikely coalition · unlikely source
Privative artificial leg · artificial snow · counterfeit merchandise · counterfeit money · fake checkpoint · fake fur · false sense · fictitious account · fictitious country · former governor · former leader · hypothetical question · imaginary country · imaginary line · imaginary world · mythical beast · mythical city · onetime rival · phony money · phony passport · phony story · phony war · simulated battle · spurious argument · virtual standstill

Table 34: Examples of modifier-noun pairs for each modifier class appearing in our sample.

two adjectives (*"ex-"* and *"pseudo-"*) due to tokenization problems and three adjectives (*"fabricated"*, *"predicted"*, and *"believed"*) due to frequent POS tag errors in the corpus. One adjective (*"deputy"*) was included in the sample, but eventually omitted from our analysis, since, upon inspecting very simple sentences, it was apparent that workers did not understand its meaning, e.g. that *"deputy chairman"* and *"chairman"* are distinctly different positions.

We gather judgements using the same annotation interface described in Section 4.2, again collecting entailment judgements on a continuous 5-point scale. We collect five independent judgements for each p/h pair and take a simple average of the five entailment ratings to be the true score.

It is worth reiterating the limitations of this experimental design, which were discussed in Section 4.1.1. Namely, we do not claim to directly assess the relationship between the underlying (set-theoretic) denotations of MH and H. Rather, we test how these modifiers effect the inferences permitted in the setting of the RTE task.



(a) Subsective (Expected)

(b) Subsective (Observed)

(c) Non-Subsective (Expected)

(d) Non-Subsective (Observed)

(e) Privative (Expected)

(f) Privative (Observed)

Figure 14: Expected vs. observed distributions of entailment judgements for inserting and deleting of modifiers by modifier class.

4.4.2. Results

Based on the theoretical adjective classes described in Section 2.3.1, we expect the composition of a privative modifier with a noun to generate the Exclusion (⊣) relation and

the composition of a plain non-subsective modifier with a noun to generate the Independence (#) relation. That is, for privative modifiers, we expect that both the insertion and the deletion direction should yield judgments of CONTRADICTION: e.g. *"fake gun"* $\Rightarrow$ $\neg$ *"gun"* and *"gun"* $\Rightarrow$ $\neg$ *"fake gun"*. Similarly, we expect plain non-subsective modifiers to yield judgments of UNKNOWN in both directions: e.g. *"alleged criminal"* $\not\Rightarrow$ *"criminal"* and *"criminal"* $\not\Rightarrow$ *"alleged criminal"*. We expect the subsective adjectives to yield ENTAILMENT in the deletion direction (*"red car"* $\Rightarrow$ *"car"*) and UNKNOWN in the insertion direction (*"car"* $\not\Rightarrow$ *"red car"*). Figures 14a, 14c, and 14e depict these expected distributions.

The entailment patterns that we actually observe for insertion and deletion within each class of modifiers are shown in Figures 14b, 14d, and 14f. Our control sample of subsective adjectives largely matched expectations (Figure 14a), with 96% of deletions producing ENTAILMENT and 73% of insertions producing NON-ENTAILMENT. The entailment patterns produced by the non-subsective adjectives, however, did not match our predictions. The plain non-subsective adjectives (e.g. *"alleged"*) behave nearly identically to how we expect regular, subsective adjectives to behave (Figure 14d). That is, in 80% of cases, deleting the plain non-subsective adjective was judged to produce ENTAILMENT, rather than the expected UNKNOWN. The privative adjectives (e.g. *"fake"*) also fail to match the predicted distribution. While insertions often produce the expected CONTRADICTION, deletions were judged to produce ENTAILMENT in a surprising number of cases (Figure 14f).

The basic entailment relations generated by modifier-noun compositions in each class are shown in Figure 15. While there is a clear difference between the three classes of modifiers, the differences are not as stark as originally expected, with all three classes producing a mix of entailment relations. Subsective modifiers generate predominately Reverse Entailments ($\sqsupset$), which accords with their formal semantics definition. In addition, plain non-subsectives generate notably more Independence (#) and Forward Entailment ($\sqsubset$) relations than do subsective modifiers, which is what we expect from plain non-subsective modifiers like *"alleged"* and *"possible"*, respectively. However, plain non-subsective modifiers also generate

Reverse Entailment ($\sqsupset$) and Equivalence ($\equiv$) relations with unexpectedly high frequency. Privative modifiers, expectedly, generate the Exclusion ($\neg$) relation substantially more frequently than either of the other two classes. However, the most frequently generated relation among the privatives is the Undefined ($\varnothing$) relation introduced in Section 4.3.3, in which $H \Rightarrow \neg MH$ but $MH \Rightarrow H$.

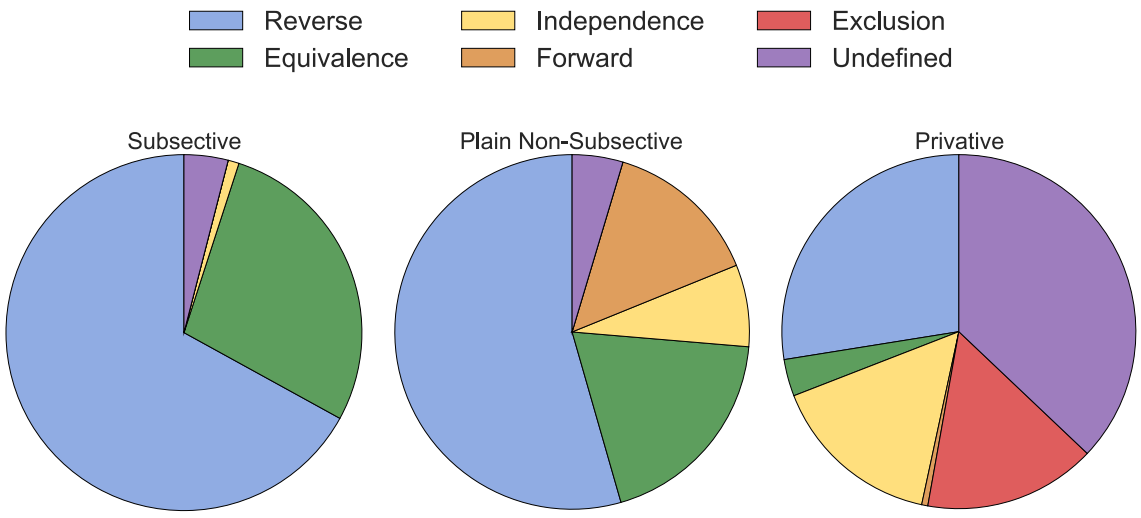


Figure 15: Distribution of entailment relations generated by the modifier-noun compositions for modifiers of different classes. The subsective (control) chart reflects the distribution over 100 p/h pairs, the plain non-subsective chart reflects 281 p/h pairs, and the privative chart reflects 178 p/h pairs. See Table 6 for theoretical relationship between modifier classes and generated natural logic relations.

4.4.3. Analysis

Plain Non-Subsectives

Table 35 gives several contexts in which plain non-subsective adjectives were judged to behave like subsective adjectives. In the examples shown, humans seem to agree that removing the modifier removes information, but their inferences do not indicate that removing it substantively alters the the underlying truth conditions of the sentence. That is, human consensus is that $MH \Rightarrow H$ but $H \not\Rightarrow MH$.

The examples shown illustrate how non-subsective modifiers often provide additional, but

Inference	Context
alleged role ⇒ role	Officials said they've launched an investigation into Urs Tinner's **alleged role**.
theoretical chance ⇒ chance	They kept it close and had a **theoretical chance** come the third quarter.
fictitious town ⇒ town	The show depicts eight officers patrolling the **fictitious town** of El Camino.
expected surge ⇒ surge	To deal with an **expected surge** in unemployment, the plan includes a huge temporary jobs program.

Table 35: Examples of sentences containing plain non-subsective modifiers; these modifiers are judged to behave the same way as subsective modifiers, i.e. to generate the Reverse Entailment (⊐) relation.

not essential, information about the particular mention of the noun they modify. For example, if a jobs program is created *"to deal with an expected surge in unemployment"*, then that program indisputably was created *"to deal with a surge in unemployment"*, just with the caveat that that surge has not yet occurred and is not guaranteed to occur in the future. For many of the non-subsective modifiers, the additional information provided relates to the speaker's belief in, or the general uncertainty of, the underlying proposition: e.g. an *"investigation into an alleged role"* entails an *"investigation into a role"* with the caveat that the role may not have existed; a team having *"a theoretical chance"* entails that the team has *"a chance"* but suggests the speaker is hedging as to how likely she believes this chance to be.

This tendency of plain non-subsectives to act as hedges may account in large part for their observed subsective-like behavior. As shown in Table 34, many of the nouns which plain non-subsectives modify in practice themselves communicate uncertainty (*"possibility"*, *"alternative"*, *"claim"*). As a result, removing the modifier is not judged to substantively alter the truth conditions of the sentence in which the noun phrase appears. It is very likely, however, that we would observe different patterns if the non-subsectives appeared as modifiers of more concrete nouns: a *"potential danger"* is still a *"danger"*, but it is unlikely that a *"potential president"* would be judged to still be a *"president"*. The fact that, in practice, these plain non-subsective modifiers co-occur so frequently with the types of

abstract nouns that they do, and so infrequently with the types of concrete nouns for which the formal semantics set-theoretic definition would be especially relevant, likely warrants further investigation. However, exploring this phenomenon is beyond the scope of this dissertation. Rather, we conclude based on these results that, for the purposes of a practical NLP system, there is no clear advantage to differentiating between plain non-subsective modifiers and subsective modifiers during inference.

Privatives

Based on the formal semantics definition, we expect that the composition of privative modifiers with nouns would, for the most part, generate the Exclusion ($\dashv$) relation. What we observe, however, is that privative modifier-noun compositions are capable of generating a range of basic entailment relations (Figure 15). In fact, Exclusion relations are generated in only 16% of cases in our dataset, and the most frequent relation generated (37% of cases) is not any of the basic relations defined in Section 2.4, but rather the Undefined ($\varnothing$) relation described in Section 4.3.3. That is, for most of the privative modifier-noun compounds in our data, MH is judged to entail H but H is judged to contradict MH.

Table 36 provides several examples of contexts for each of the generated relations. The examples reveal a very high degree of context dependence: whether or not the insertion of the privative modifier changes the relevent truth conditions of the sentence seems more dependent on the pragmatic purpose of the sentence overall than on either the modifier or the noun itself. For example, *"mock debate"* is judged to entail *"debate"* while *"mock execution"* is judged to be contradictory with *"execution"*. Such inferences make sense if the choice of whether MH entails or contradicts H is governed by the sentence's effectiveness at communicating a situation, rather than by faithfulness to an underlying set-theoretic model of the noun phrases' denotations. I.e. a listener who hears *"debate"* in place of *"mock debate"* will have a near-perfect understanding of what took place, whereas a listener who hears *"execution"* in place of *"mock execution"* will be drastically misinformed.

MH	$\beta(e)$	Context
virtual unknown	$\equiv$	Aponte went from an **unknown** to toast of the town.
erstwhile rival	$\equiv$	Sarkozy sent Strauss-Kahn away, ridding himself of a **rival**.
artificial light	$\sqsupset$	The plants were grown under **light**.
would-be terrorist	$\sqsupset$	They disputed the portrayal of Siddiqui as a **terrorist**.
artificial intelligence	#	Leaps in **intelligence** would lead to driverless cars.
past season	#	He had four goals in 24 games this **season**.
mock execution	$\dashv$	The prisoner had been subjected to an **execution**.
fake bomb	$\dashv$	He'd actually been provided a **bomb** in an FBI sting.
imaginary friend	$\dashv$	Emily has a new **friend**.
counterfeit medicine	$\varnothing$	Pharmacists denied selling **medicine** in their stores.
mock debate	$\varnothing$	He also took part in a **debate** Sunday.
fictitious company	$\varnothing$	Wilson signed off to pay the debts to the **company**.

Table 36: Examples of contexts in which privative modifier-noun composition results in each of the basic entailment relations plus the Undefined ($\varnothing$) relation.

Again, our focus in this thesis is on improving automatic systems for natural language inference. Thus, we do not attempt to provide a theoretical model to account for the observations presented here. Rather, we emphasize the fact that privative modifiers, in practice, can generate any of the basic entailment relations, not only the Exclusion relations. In fact, as a class, the privative modifiers appear to be the most context sensitive, and the least amenable to a naive "most frequent class" treatment. As a result, a model which makes assumptions about inferences based solely on the class of the modifier involved is likely to make more incorrect predictions than correct ones. We will return to this point, as it relates to modifier-noun composition more generally, in our discussion in Section 4.6.

4.5. Performance of Current RTE Systems

Sections 4.3 and 4.4 have illustrated that modifier-noun composition is a complex process. Human inferences about modified noun phrases often rely on subtle contextual cues or on common sense assumptions not readily available given the text alone. As a result, inferences involving modifier-noun composition are likely to be very challenging for automatic NLU systems. In this section, we evaluate the performance of several state-of-the-art RTE systems, representing a range of approaches to automated natural language inference, in

order to quantify how well modifier-noun composition is handled by current technology.

4.5.1. The Add-One Entailment Task

We reframe our simplified RTE task, described in Section 4.1.2 and used for our human annotation, as a challenge task to be performed by automatic RTE systems. Specifically, we define the "Add-One Entailment" task, which is identical to the normal RTE task, except with the constraint that the premise p and the hypothesis h differ only by the atomic insertion of a modifier: $h = e(p)$ where $e = INS(M)$ and M is a single adjective. To provide a consistent interface with a range of different RTE systems, we use a binary label set: NON-ENTAILMENT (which encompasses both CONTRADICTION and UNKNOWN) and ENTAILMENT.

We build training, development, and test sets for the Add-One RTE task using the data collected in Section 4.2. In evaluating systems, we want to test only on straightforward p/h pairs, so as not to punish systems for failing to classify examples which humans themselves find difficult to judge. Therefore, to build our test set, we consider pairs with mean human scores ≤ 3 as NON-ENTAILMENT and pairs with scores ≥ 4 as ENTAILMENT, omitting the pairs which fall into the ambiguous range in between. For our training and development sets, we include all pairs, considering scores < 3.5 as NON-ENTAILMENT and scores ≥ 3.5 as ENTAILMENT. We tried removing the "ambiguous" pairs from the training and development sets as well, but it did not affect the systems' performances. Our resulting train, dev, and test sets contain 4,481, 510, and 387 pairs, respectively. These splits cover disjoint sets of MHs, i.e. none of the MHs appearing in test were seen in train. Individual adjectives and/or nouns may appear in both train and test. The dataset consists of roughly 85% NON-ENTAILMENT and 15% ENTAILMENT. Human agreement achieves 93% accuracy on the test set (Section 4.2.4).

4.5.2. Description of Evaluated RTE Systems

We test a variety of state-of-the-art RTE systems, covering several popular approaches to RTE. These systems are described in more detail below.

Classifier-Based Systems

The Excitement Open RTE platform (Magnini et al. (2014)) includes a suite of RTE systems, including baseline systems as well as feature-rich supervised systems which have been shown to achieve state-of-the-art performance on the RTE3 datasets (Giampiccolo et al. (2007)). We test two systems from Excitement: the simple Maximum Entropy model (henceforth referred to as **MaxEnt**) which uses a suite of dense, similarity-based features (e.g. word overlap, cosine similarity), and the more sophisticated Maximum Entropy model (**MaxEnt+LR**) which uses the same similarity-based features but additionally incorporates features from external lexical resources such as WordNet (Fellbaum (1998)) and VerbOcean (Chklovski and Pantel (2004)). We also train a standard unigram model (**BOW**), with regularization determined using cross-validation on the training data.

Transformation-Based System

The Excitement platform also includes a transformation-based RTE system called **BIUTEE** (Stern and Dagan (2012)). The BIUTEE system derives a sequence of edits that can be used to transform the premise into the hypothesis. These edits are represented using feature vectors, and the system searches over edit sequences for the lowest cost "proof" of either entailment or non-entailment. The feature weights are set by logistic regression during training. BIUTEE can be viewed as a robust hybrid between logical systems (like the Nutcracker system used in Section 3.4) and the above classifier-based systems.

Deep Learning Systems

Bowman et al. (2015) recently reported promising results on benchmark RTE datasets by using deep learning architectures trained on very large training data. We test the performance of the same implementations on our Add-One task. Specifically, we test three models: **Sum**: a basic sum-of-words model, which represents both p and h as the sum of their word embeddings; **RNN**: a vanilla recurrent neural network; and **LSTM**: a vanilla long short term memory network. We also train a bag-of-vectors model (**BOV**), which is simply a logistic regression whose features are the concatenated averaged word embeddings of p and h.

For the LSTM, in addition to the normal training setting–i.e. training only on the 5,000 Add-One training pairs–we test a transfer-learning setting (**Transfer**). In transfer learning, the model trains first on a large general dataset before fine-tuning its parameters on the smaller set of target-domain training data. For our Transfer model, we train first on the 500,000 pair SNLI dataset (Bowman et al. (2015)) until convergence, and then fine-tune on the 5,000 Add-One pairs. This setup enabled Bowman et al. (2015) to train a high-performance LSTM for the SICK dataset (Marelli et al. (2014b)), which is of similar size to our Add-One dataset. See Section 2.1.3 for a description of the SNLI and SICK datasets.

4.5.3. Results and Analysis

Out-Of-The-Box Performances

To calibrate expectations, we measure the performance of each of the above systems on the datasets for which they were originally designed. For the Excitement systems, this is the RTE3 dataset (Table 16). For the deep learning systems, this is the SNLI dataset (Table 17). For the deep learning systems, in addition to reporting performance when trained on the full SNLI corpus (500,000 p/h pairs), we report the performance in a reduced training setting in which systems only have access to 5,000 p/h pairs, drawn randomly from the full

training set. This is equivalent to the amount of data we have available for the Add-One task, and is intended to give a sense of the performance improvements we should expect from these systems given the size of the training data.

System	Accuracy
Majority Class Baseline	51.3
BOW	51.0
MaxEnt	61.5
Edit Dist.	61.9
MaxEnt+LR	63.6
BIUTEE	65.6

Figure 16: Performance of systems from Magnini et al. (2014) on the RTE3 dataset (the dataset on which they were originally developed).

System	SNLI 500K	SNLI 5K
Majority Class Baseline	65.7	
BOV	74.4	71.5
RNN	82.1	67.0
Sum	85.3	69.2
LSTM	86.2	68.0

Figure 17: Performance of systems from Bowman et al. (2015) on the SNLI dataset (the dataset on which they were originally developed) using both the full training set (500K pairs) and a reduced training set (5K pairs).

Performance on Add-One RTE.

We train each of the systems on the training and development sets of Add-One p/h pairs and test on our held-out set of 387 pairs. For the deep learning systems which require separate training and development sets, we use our splits of 4,481 training pairs and 510 development pairs. For the remaining systems, we combine training and development into a single training set containing 4,991 pairs.

Figure 18 shows the overall accuracy achieved by each system. The baseline strategy of predicting the majority class for each adjective, based on the training data, reaches close to human performance[8] (92% accuracy). Given the simplicity of the task (p and h differ

[8]It is worth noting that this baseline performs very well due to the strict filtering we applied to our test

by a single word), this baseline strategy should be learnable. However, none of the systems tested come close to this level of performance, indicating that they fail to learn even the most-likely entailment generated by each adjective: e.g. that in general, $INS(\textit{"brown"})$ probably generates NON-ENTAILMENT in most contexts and $INS(\textit{"real"})$ probably generates ENTAILMENT in most contexts. The best performing system is the RNN, which achieves 87% accuracy, only two points above the baseline of always guessing NON-ENTAILMENT.

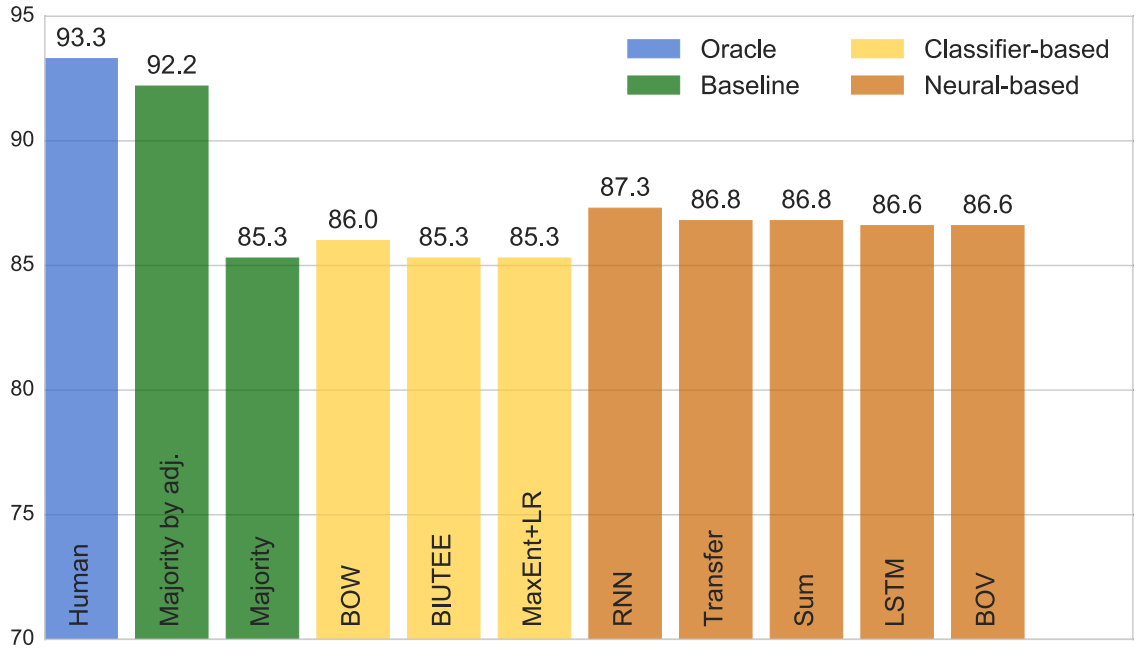


Figure 18: Accuracy achieved by all tested RTE systems on AddOne RTE task.

Table 37 reports the results in terms of precision, recall, and F1 score for the ENTAILMENT class which, recall, comprises 15% of all p/h pairs in our data. The feature-rich classification-based systems, BIUTEE and MaxEnt+LR, fail to learn any way of distinguishing the ENTAILMENT class from NON-ENTAILMENT, leading both to resort to the baseline strategy of classifying everything as NON-ENTAILMENT. This failure to learn is likely attributable to the fact that their features do not capture differences between individual adjectives. In fact,

sets in order to ensure fairness and reproducibility of results. As described in Section 4.3, many inferences depend on both the M and the H, and many depend further on the particular context s in which the MH appears. However, in treating entailment as a discrete classification task, it is difficult to capture these especially interesting and challenging cases while still maintaining reproducibility.

BIUTEE contains a feature explicitly intended to penalize the insertion of modifiers into the hypothesis that do not appear in the premise, under the assumption that modification always results in the introduction of new (not otherwise entailed) information. The basic bag of words classifier (BOW) actually outperforms the more complex feature-engineered models. Inspecting the feature weights, we see that the BOW model does manage to assign positive weights when "presence/saliency" modifiers (discussed in Section 4.3.2) appear in the hypothesis and negative weights when more clearly restrictive modifiers do (Table 38). However, the signal is weak and does not account for deeper context dependencies, leading to overall performance that is still disappointingly low.

Model	P	R	F
Majority	0.00	0.00	0.00
BIUTEE	0.00	0.00	0.00
MaxEnt+LR	0.00	0.00	0.00
BOW	0.58	0.19	0.29
LSTM	0.56	0.39	0.46
BOV	0.60	0.26	0.37
Transfer	0.59	0.33	0.43
Sum	0.65	0.23	0.34
RNN	0.60	0.44	0.51
Majority by adj.	0.86	0.56	0.68
Human	0.84	0.64	0.73

Table 37: Precision, recall, and F1 score for all systems on AddOne RTE task.

Indicative of entailment: own · real · whole · entire · human · general · same · current · single · personal · full · major · such · strong · direct · much · fine · your · bright · every
Indicative of non-entailment: black · white · small · green · red · old · third · yellow · second · little · brown · jewish · american · first · other · pink · international · purple · gray · chinese

Table 38: Top 20 modifiers most likely to correspond to the ENTAILMENT class and most likely to correspond to the NON-ENTAILMENT class when appearing in the hypothesis, according to the basic BOW classifier.

The deep learning systems perform slightly better, likely due to their ability to generalize over individual modifiers and to better capture, to a small extent, the context of the sentence overall. However, the absolute performance of even the best-performing model, the RNN,

True	Pred.	Hypothesis
NON.	ENT.	Gold has always held some **moral value**, even during near anarchy.
NON.	ENT.	The act could clearly cost him his **private life**.
NON.	ENT.	Human rights groups put the **popular figure** closer to 30,000.
NON.	ENT.	A hiker walking on a **sandy path** at the foot of snow capped mountains.
ENT.	NON.	Bush 's spending made the **entire economy** worse.
ENT.	NON.	But the **main reason** he got spanked was being a total jerk to his cousin.
ENT.	NON.	Those without an **good education** will be left behind.
ENT.	NON.	The **enthusiastic crowd** roared: "slobo Saddam."

Table 39: Examples of false positive predictions and false negative predictions of the RNN (the best-performing of the systems we tested) on AddOne RTE test data.

is low: it achieves an F1 of only 0.51, compared to 0.68 which could have been achieved by simply memorizing the most likely class given the inserted modifier. Table 39 provides several examples of incorrect predictions made by the RNN. While no obvious pattern stands out among the incorrect predictions, several of the false positive predictions involve common collocations (e.g. *"moral value"*, *"popular figure"*) that, while frequent in general, do not apply in the specific context given.

4.6. Discussion

In this chapter, we performed an in-depth study of human inferences surrounding modifier-noun composition. We showed that the conventional wisdom, that inferences about modifiers are systematic and governed by the class to which the modifier belongs, leads to incorrect predictions a significant portion of the time. We provided evidence that context and common sense, rather than principled logical reasoning, affect many inferences that humans make. This is especially true in the case of privative and non-subsective modifiers, for which inferences appear the least formulaic and the most situation-specific. Our evaluation of several state-of-the-art RTE systems revealed that the level of pragmatic reasoning required to accurately model modifier-noun composition in context is beyond the capability of current technologies.

Taking everything together, the examples and analyses presented in this chapter reveal a

consistent pattern in which pragmatic inferences supersede semantic ones. That is, a system which perfectly adheres to logical models of inference will certainly make incorrect inferences. The question of how often is dependent on the genre (Figure 9) and likely the style of the language (whether it is concrete or abstract, definitive or speculative), but errors will inevitably occur. As we emphasized throughout the chapter, the priority in the field of NLP generally (and in this thesis in particular) has not been to model language for its own sake, but rather to make correct inferences about language in practice. From this perspective, then, our results lean toward the conclusion that there is no practical utility to differentiating between modifier classes. That is, while the formal semantics classification can provide a non-trivial prior distribution on the inferences to be made (e.g. when inserted, subsective modifiers *in general* generate non-entailment and privative modifiers *in general* generate contradiction) the accuracy a system achieves by following the prior alone is prohibitively low for more advanced natural language understanding. Thus, in order for NLP systems to improve inference beyond baseline accuracy, they will need a functioning model of pragmatic and common sense reasoning, and once this model of pragmatics is in place, our results suggest little advantage to considering the formal semantics class of the modifier when making inference decisions.

The relevant question then becomes: how do we build such a model of pragmatic reasoning? A complete answer to this question is fundamental to the study of language, and is far beyond the scope of this thesis. Thus, we comment only on the patterns which are observable from the examples presented in this chapter. Specifically, the problem central to the analyses here is: given *one particular mention* of a noun in a given context, how do we decide whether that mention belongs to the set denoted by some particular noun phrase MH. For example, given the mention of *"economy"* in the sentence *"President Bush spoke in Michigan about the economy"*, how do we decide whether the entity referenced by this mention of *"economy"* falls within the denotation of *"Japanese economy"*?

Based on the observations in this chapter, we see that, consistently, a particular mention of a

noun (i.e. an entity) is assigned a number of properties by default. These default properties, in turn, effect the inferences that are made about the sets to which that entity belongs, and the noun phrases (MHs) which can be used to refer to that entity. Some of these default properties seem to be assigned uniformly regardless of the noun or the context: for example, every entity is assumed to be *"real"*, unless otherwise specified. Other default properties are assigned based on the noun alone, according to the noun's so-called[9] "prototypical" form: for example, *"beaches"* are assumed to be *"sandy"*, unless otherwise specified. Still other default properties are assigned based on the specific context in which the mention occurs: for example, the *"economy"* about which the president of the United States is speaking is the *"American economy"*, unless otherwise specified.

In general, the human inferences we observe tend to accord with these default assumptions. Inferences which are entailed by the default assumptions are judged as entailments (*"problem"*$\Rightarrow$ *"real problem"*, *"economy"*$\Rightarrow$ *"American economy"*), and those which contradict the default assumptions are judged as contradictions (*"problem"*$\Rightarrow \neg$ *"imaginary problem"*, *"economy"*$\Rightarrow \neg$ *"Japanese economy"*). That is, our observations accord with a pattern of conversational implicature: while H does not *technically* entail MH in this context, it is so highly likely that H means MH (given the prototypical interpretation of H, in general and in this context specifically) that if H does not mean MH, the speaker was obligated to say so explicitly; thus, the listener can reasonably conclude that H entails MH.

In order for a computational system to accurately handle such context-specific inferences about modifier-noun phrases, it will at a minimum need to do two things. First, it needs to recognize, for a specific entity referenced by a noun H in context, what properties are likely to hold for that entity. (More likely, the system will need to recognize not just what properties hold for the entity in general, but rather, what properties are important for the central point that the speaker is trying to communicate.) Second, the system needs to understand what properties are entailed by the modifier. That is, the system needs to

[9] :)

know which properties must necessarily hold for the entity in question if that entity is to be referenced using the noun phrase MH.

The first of the above points, modeling common sense inference and pragmatic intent of an utterance, is a weighted question which we leave for future work. The second, modeling the intrinsic semantics of the modifiers, in terms of the properties they entail about the entities for which they hold, is the focus of our work in Chapter 5.

CHAPTER 5 : Noun Phase Composition for Class-Instance Identification

The focus of the previous chapter was semantic containment: e.g. how do we decide whether a *"red dress"* is a *"dress"* or a *"beach"* is a *"sandy beach"*? Full language understanding, however, requires more than an understanding of the set-theoretic relationships between noun phrases. For example, a system cannot be said to understand the meaning of a phrase like *"American composer"* if it knows only that an *"American composer"* is a *"composer"*, but cannot describe what it is, exactly, that makes a composer *"American"*.

In this chapter, we focus on modeling the meaning of modifiers themselves such that it is possible to determine, for an individual entity, whether or not the modifier holds. There are two central problems addressed in this chapter. The first is *modifier interpretation*: what does it mean for a *"composer"* to be *"American"*? The second is *class-instance identification*: how do we decide whether or not an individual *"composer"* is an *"American composer"*? We focus solely on subsective modifiers, and instantiate the formalization given in Section 2.3.1, in which modifiers are functions which operate on the sets denoted by the noun phrases they modify. We build representations of these functions automatically from natural language text, and apply them to the task class-instance identification for arbitrary classes which may involve many modifiers: e.g. *"1950s American jazz composer"*.

In Section 5.1, we restate the formalization of modifier-noun composition from formal semantics, and outline desiderata for our operationalization of this formal semantics definition. In Section 5.2, we describe our approach for assigning meaning to modifiers using paraphrasing techniques. In Section 5.3, we use our modifier interpretations to recognize class-instance relations: given an specific entity (e.g. *"Charles Mingus"*), determine whether or not it is an instance of a specific class (e.g. *"jazz composer"*). In Section 5.4, we evaluate our method's performance on the task of identifying class-instance relations by using it to discover instances of categories listed on Wikipedia category pages given only unstructured natural language text.

5.1. Modeling the Semantics of Noun Phrases

This section highlights some of the major theoretical implications of the formal semantics definitions of modifier-noun composition, and discusses the relevance of these concepts for practical NLP systems. We describe some shortcomings of existing computational approaches to modeling compositional noun phrases.

5.1.1. Modifiers in Formal Linguistics

Recall from Section 2.3.1 that in formal semantics, subsective modifiers are modeled as functions which map between sets: that is, they take as input the set denoted by the unmodified noun phrase, and return the more narrow subset denoted by the modified noun phrase. Specifically, let MH be a noun phrase consisting of a head noun phrase H, preceded by a modifier M. The interpretation of the head H is a set of entities in the universe $\mathcal{U}$, as below:

$$[\![H]\!] = \{e \in \mathcal{U} \mid e \text{ is a } H\} \tag{5.1}$$

The interpretation of the modifier M, then, is a function that selects the subset of entities from $[\![H]\!]$ which meet the criteria specified by the modifier:

$$[\![MH]\!] = [\![M]\!](H) = \{e \in H \mid e \text{ is } M\} \tag{5.2}$$

This formalization leaves open how one decides whether or not "e is M". Determining whether this statement holds is non-trivial, as the meaning of a modifier often varies depending on the class being modified. For example, just because e is a *"good student"*, it is not necessarily the case that e is a *"good person"*, making it difficult to model whether "e is *good*" in general. Thus, determining whether or not "e is M" requires a model of the core "meaning" of the modifier M in the context of H. That is, it requires a representation of M which encapsulates all of the properties which are entailed by M and which differentiate members of the output class MH from members of the more general input class H. In Sections 5.2 and 5.3, we will propose a concrete instantiation of Equation 5.2.

5.1.2. Desiderata

The above formal semantics framework has two important properties, which we aim to preserve when we operationalize it for use in NLP.

1. **The modifier has an intrinsic "meaning".** That is, there are properties entailed by the modifier that are independent of the particular state of the world. E.g. the premise *"Eddy is a composer born in America"* entails the hypothesis *"Eddy is an American composer"* in all possible worlds, regardless of the particular entity to which *"Eddy"* refers. Said differently, the statement "e is M" has semantic meaning, regardless of the extension of H, and even if $[\![M]\!](H)$ returns the empty set.

2. **The modifier is a function that can be applied in a truth-theoretic setting.** That is, given a particular model of the world, it is possible to apply the meaning of the modifier in order to assign truth values to propositions. E.g. applying *"American"* to the set of *"composers"* returns exactly the set of *"American composers"*, enabling a system to determine whether or not *"Eddy is an American composer"* is in fact true.

The above properties make it possible to reason about entailment in the strict formal linguistics sense (Section 2.1.1) as well as in the informal NLP sense (Section 2.1.2).

For the majority of *current* NLP applications, it is admittedly difficult to motivate why meeting the above desiderata is important. Most work at present focuses on tasks which operate entirely in the realm of one particular world (the "real world") and on applications which can benefit from big data and massive redundancy: e.g. answering factoid questions or characterizing widely-discussed current events. As we attempt to apply NLP to more advanced applications, however, we can expect the demands on systems to shift significantly. Eventually, we will want NLP systems to understand and precisely summarize individuals' accounts and opinions, like those presented in personal anecdotes, scientific papers, or legal arguments. Doing so will undoubtedly require the ability to understand hypothetical scenarios and to reason about entities and concepts which are not described elsewhere,

or which cannot be grounded to the real world at all. Certainly such applications will require systems to perform close readings of one-off documents, without the ability to exploit redundant expressions of the same proposition. Thus, meeting the above desiderata now positions models to better handle these more advanced applications in the future.

5.1.3. Weaknesses of Existing Computational Approaches

The notion of modifiers as functions has been incorporated into computational models previously. However, existing approaches have not simultaneously satisfied both of the desiderata discussed above. For example, Baroni and Zamparelli (2010) focuses exclusively on modifier interpretation, and on learning an explicit semantic representation for a given modifier. Specifically, their proposed model represents modifiers as $n \times n$ matrices which map the n-dimensional vector corresponding to H to the n-dimensional vector corresponding to MH. This model meets the first of the desiderata above: the $n \times n$ matrix is an explicit representation of the "meaning" of the modifier M. However, their model focuses exclusively on measuring the similarity between noun phrases–e.g., to say that *"important routes"* is similar to *"major roads"*–and it is not obvious how the method could be operationalized in a truth-theoretic setting. That is, given a specific instance $e \in H$ and a specific model of the world, it is not clear how a system might use the representation of M in order to determine whether or not $e \in MH$.

Young et al. (2014), in contrast, focuses exclusively on class-instance identification. Specifically, their proposed model consists of a graph of noun phrases derived from images, in which nodes in the graph correspond to entities or to sets and are linked by instance-of and set-containment relations. As a result, given a set of instances H and a modifier M, the method presented in Young et al. (2014) can return the subset constituting MH. However, their method does not model any intrinsic meaning for the modifier itself. Thus, if there were no red cars in their model of the world, the phrase *"red cars"* would have no meaning.

The majority of related prior work does not model compositionality at all. Rather, most

work on class-instance identification treats noun phrases as atomic non-compositional units, and relies on lexico-syntactic patterns as the primary signal of whether a given instance e is a member of the class MH. That is, the most common approaches require that patterns like *"e is a MH"* and *"MH such as e"* occur sufficiently frequently in text (Snow et al. (2006); Shwartz et al. (2016)). Ignoring the compositionality of MH performs sufficiently well, for practical purposes, when MH contains only one or two words (e.g. *"composer"* or *"American composer"*), but as phrases become longer (e.g. *"1950s American jazz composer"*), the probability of occurring, even in a very large corpus, becomes prohibitively low. Sparsity-related concerns aside, such non-compositional models have the additional weakness that they cannot model the meaning of the modifier alone. As a result, such models can only reason about the *extension* of the MH, and not about the entailments intrinsic to the MH itself, a severe limitation as language tasks become more complex.

In this thesis, we model the semantics of M intrinsically, but in a way that permits application in the model theoretic setting. Specifically, we learn an explicit model of the "meaning" of a modifier M relative to a head H, represented as a distribution over properties which differentiate the members of the class MH from those of the class H. We then use this representation to identify the subset of instances of H which constitute the subclass MH.

5.2. Modifier Interpretation

The task of modifier interpretation is: given a modifier M applied to a head noun H, determine what it means when M holds for a given $e \in H$. Our goal, then, is to determine what it means to say "e is M" in the context of Equation 5.2. We aim to do this by determining the set of properties that are entailed by M, and thus must be true of any e for which "e is M" is true. This section describes our method for learning such a set of properties. In Section 5.3, we will apply these properties to the task of class-instance identification.

Note that, throughout this chapter, we may use the terms "set" and "class" interchangeably.

Specifically, "class" simply refers to the set of entities in $\mathcal{U}$ which are assigned some name (or *class label*) in natural language. E.g. a the set of every entity which is an American composer constitutes a class with the label *"American composers"*.

5.2.1. Assumptions of our Approach

In general, there is no systematic way to determine the implied relation between M and H: is an *"American composer"* a *"composer born in America"*, a *"composer who lives in America"*, or simply any *"composer who has visited America"*? It has been argued that, given the right context, modifiers can express any possible semantic relation (Section 2.3.3). We therefore model the semantic relation between M and H as a distribution over properties which could potentially define the subclass $MH \subseteq H$. We will refer to this distribution as a *property profile* [10] for M relative to H.

We make the assumption that relations between M and H that are discussed more often in a large text corpus are more likely to capture the important properties of the subclass MH. This assumption is not perfect (Section 5.2.4) but has lead to good results in prior work on noun phrase paraphrasing (Nakov and Hearst (2013); Pasca (2015)). Specifically, our method for learning property profiles is based on the unsupervised method proposed by Pasca (2015), which uses query logs as a source of common sense knowledge, and rewrites noun compounds by matching MH (*"American composers"*) to queries of the form "HrM" (*"composers from America"*), where r (*"from"*) can be any natural language expression.

5.2.2. Data Processing

We assume two inputs: 1) an IsA repository, $\mathcal{O}$, containing $\langle e, C \rangle$ tuples where C is a class label[11] and e is an instance of C, and 2) a fact repository, $\mathcal{D}$, containing $\langle s, r, o, w \rangle$ tuples where s and o are noun phrases, r is a predicate, and w is a score reflecting the confidence

[10] Note that the more commonly used terminology in existing work is "interpretation": the interpretation of M relative to H. However, we use the phrase "property profile" to avoid overloading the formal semantics definition of "interpretation", as discussed in Section 2.1.1.

[11] We abuse notation slightly and use C to represent both the class (a set of entities in $\mathcal{U}$) and the class label (a natural language string).

that r expresses a true relation between s and o.

IsA Repository

We instantiate $\mathcal{O}$ with an IsA repository extracted from a sample of around 1 billion Web documents in English. $\mathcal{O}$ is constructed by applying the following four lexico-syntactic patterns to the Web corpus: *"C such as E"*, *"E is a C"*, *"C including E"*, and *"C especially E"*, where E is either a single entity (e.g. *"composers such as Mingus"*) or an enumeration of entities (e.g. *"composers such as Ellington, Davis, and Mingus"*). All patterns receive equal weight, but the second is the most productive in practice. Instances in $\mathcal{O}$ are represented as automatically-disambiguated entity mentions. We use the entity linker described in Lazic et al. (2015). Each entity is assigned to a unique ID and may correspond to an individual, like *"Charles Mingus"*, or to a concept, like *"jazz"*. When possible, entities are resolved to Wikipedia pages; for example, *"America"* and *"USA"* will be mapped to the same Wikipedia article. Classes in $\mathcal{O}$ are represented as non-disambiguated strings in natural language. In building the repository, we retain every $\langle e, C \rangle$ tuple which is supported by 5 or more sentences and has a confidence of at least 0.9. We compute "confidence" using a weighted combination of a handful of metrics, including the number of supporting sentences and the overall frequency of the category C. Weights are set automatically, using a hand-labeled tuning set and optimized to give a good trade-off between precision and recall of true pairs. The resulting repository contains 1.1M IsA relations, covering 412K instances and 9K categories. Some examples of $\langle e, C \rangle$ tuples from $\mathcal{O}$ are given in Table 40.

⟨ Alice Starmore, designer⟩	⟨ Leroy Hutson, artist⟩
⟨ Beeb Birtles, character⟩	⟨ Psychoneuroimmunology, field⟩
⟨ Bodoland People 27s Front, party⟩	⟨ Richard Davison, champion⟩
⟨ Clymer repair manual, tool⟩	⟨ SLC31A1, protein⟩
⟨ Dwayne Russell, player⟩	⟨ Vélo'v, service⟩
⟨ Hu Shih, philosopher ⟩	⟨ Whispers II, album⟩

Table 40: Example ⟨instance e, Class C⟩ tuples from our IsA repository $\mathcal{O}$.

Fact Repository

We instantiate $\mathcal{D}$ with a large repository of facts extracted from a sample of around 1 billion Web documents in English. $\mathcal{D}$ is extracted using an in-house open information extraction system, based on the methods proposed by Fader et al. (2011) and Mausam et al. (2012), applied to the Web corpus. We leave predicates as natural language strings, but remove stop words and apply basic lemmatization (e.g. the predicate *"is an important part of"* becomes *"be important part of"*). Subjects and objects may be either disambiguated entity references, as in $\mathcal{O}$ described above, or may be natural language strings. Every tuple is included in both the forward and the reverse direction. For example, the tuple ⟨ *"jazz"*, *"perform at"*, *"venue"*⟩ also appears as ⟨ *"venue"*, ← *"perform at"*, *"jazz"*⟩, where ← is a special character signifying inverted predicates. The weight w associated with each subject-predicate-object tuple is the number of times the tuple was extracted from the corpus. In total, our fact repository contains 30M tuples. Some examples of $\langle s, r, o \rangle$ tuples from $\mathcal{D}$ are given in Table 41.

⟨actress, detain in, Pakistan ⟩	⟨ game, publish in, Japanese ⟩
⟨ champion, ←produce, wrestling ⟩	⟨ laureate, ←go to, nobel ⟩
⟨ channel, ←scan, television ⟩	⟨ painter, visit, Germany ⟩
⟨ converter, consume, wave energy ⟩	⟨ protein, ←mediate, ring finger⟩
⟨ designer, do in, Malaysia ⟩	⟨ show, ←give, Fox soccer channel⟩
⟨ film, ←be great year for, 1962 ⟩	⟨ system, install in, medicine ⟩

Table 41: Example ⟨subject s, predicate r, object o⟩ tuples from our fact repository $\mathcal{D}$.

5.2.3. Associating Properties with Modifiers

Given a particular modifier M associated with a particular head H, and given $\mathcal{O}$ and $\mathcal{D}$ as described above, we build a *property profile* which reflects the relationship between M and H. That is, this profile ideally captures the properties which discriminate the instances of the class MH from instances of the more general class H. Below, let I be a function which takes as input a noun phrase MH and returns a property profile for M relative to H.

Definition of "Property"

We define a "property" to be an subject-predicate-object tuple in which the subject position is a wildcard, e.g. $\langle *, \textit{born in}, \textit{America} \rangle$. Because our fact repository includes inverse predicates, this definition is still capable of capturing properties in which the wildcard is conceptually the object of the relation, although it technically occupies the subject slot in the tuple. For example, $\langle \textit{venue}, \leftarrow \textit{perform at}, \textit{jazz} \rangle$ captures that a *"jazz venue"* is a *"venue"* e such that *"jazz performed at e"*. We say that any entity e which has been observed filling the wildcard slot in $\mathcal{D}$ "has" the property. For convenience, we will often display properties as strings (*"* born in America"*) rather than as tuples ($\langle *, \textit{born in}, \textit{America} \rangle$). When doing so, we will spell out the inverted predicates accordingly: e.g. $\langle \textit{venue}, \leftarrow \textit{perform at}, \textit{jazz} \rangle$ will be displayed as *"jazz perform at *"*.

Relating M to H Directly

We first build property profiles for a given MH by taking the predicate and object from any tuple in $\mathcal{D}$ in which the subject is the head and the object is the modifier:

$$I_{head}(MH) = \{\langle \langle r, M \rangle, w \rangle \mid \langle H, r, M, w \rangle \in \mathcal{D}\} \tag{5.3}$$

In the above definition, we expand adjectival modifiers to encompass nominalized forms using a nominalization dictionary extracted from WordNet (Fellbaum (1998)). For example, if MH is *"American composer"* we would extract properties from tuples matching $\langle H, r,$ *"American"*$, w \rangle$ and well as from tuples matching $\langle H, r,$ *"America"*$, w \rangle$.

Relating M to an Instance of H

We also consider an extension of Equation 5.3 above, in which, rather than requiring the subject to be the class label H itself, we require the subject to be any instance of H.

$$I_{inst}(MH) = \{\langle\langle r, M\rangle, w\rangle \mid \langle e, H\rangle \in \mathcal{O} \wedge \langle e, r, M, w\rangle \in \mathcal{D}\} \tag{5.4}$$

For example, if MH is *"American composer"*, we could extract properties from any tuple in which the subject is an instance of *"composer"* according to $\mathcal{O}$: e.g. $\langle$ *"Mingus"*$, r, M, w\rangle$ and $\langle$ *"J.S. Bach"*$, r, M, w\rangle$. As above, we expand adjectival modifiers to encompass their nominalized forms.

Modifier Expansion

In practice, when building property profiles, we do not require that the object of the fact tuple match the modifier (or its nominalized form) exactly, as suggested in Equations 5.3 and 5.4. Instead, we follow Pasca (2015) and take advantage of facts involving distributionally similar modifiers. Specifically, rather than looking only at tuples in $\mathcal{D}$ in which the object matches M, we consider all tuples, but discount the weight proportionally to the similarity between M and the object of the tuple. Thus, in practice, I_{head} and I_{inst} are computed as below:

$$I_{head}(MH) = \{\langle\langle r, M\rangle, w \times sim(M, N)\rangle \mid \langle H, r, N, w\rangle \in \mathcal{D}\} \tag{5.5}$$

$$I_{inst}(MH) = \{\langle\langle r, M\rangle, w \times sim(M, N)\rangle \mid \langle e, H\rangle \in \mathcal{O} \wedge \langle e, r, N, w\rangle \in \mathcal{D}\} \tag{5.6}$$

where $sim(M, N)$ is the cosine similarity between M and N. We compute sim using a vector space built from Web documents following the algorithms described in Lin and Wu (2009) and Pantel et al. (2009). We retain the 100 most similar phrases for each of approximately 10 million phrases, and consider all other similarities to be 0.

5.2.4. Analysis of Learned Properties

In general, our method yields good results. That is, the assumption that frequently-discussed relations between M and H will capture relevant properties of MH is often accurate. Table 42 provides example property profiles returned for using several MHs using I_{head}. By and large, the predicate which most frequently relates H to M in $\mathcal{D}$ proves to be a good paraphrase of the meaning of MH: e.g. a *"still life painter"* is a *"painter who paints still life"* and a *"Led Zeppelin song"* is a *"song written by Led Zeppelin"*.

rice dish	French violinist	Led Zeppelin song	still life painter
* serve with rice	* live in France	Led Zeppelin write *	* paint still life
* come with rice	* born in France	Led Zeppelin play *	* create still life
* make with rice	* be in France	Led Zeppelin have *	* do still life
* have rice	* go to France	Led Zeppelin perform *	* make still life

Table 42: Example property profiles learned by observing predicates that relate the class H to modifier M (I_{head}). Results, among top-ranked properties, are similar when using I_{inst}.

Not every MH is handled well by our method, however. For example, the most frequently discussed relation between *"child"* and *"actor"* is that actors have children, but this property is not indicative of the meaning of *"child actor"*. Several examples of MHs for which the our method fails to learn good property profiles are shown in Table 43. We discuss means of addressing these types of noun phrases in our discussion in Section 5.5.

child actor	risk manager	machine gun
* have child	* take risk	machine have *
* expect child	* be at risk	machine equip with *
* play child	* be aware of risk	machine use *

Table 43: Examples of MHs for which our central assumption–that frequently-discussed relations between M and H capture relevant properties of MH–does not hold.

Importantly, we do see that the profiles capture the fact that the meaning of M is often dependent on the head H: for example, an *"American company"* is a company that is *"based in America"* while an *"American composer"* is a composer who is *"born in America"* (Table 44).

M	H	Top-ranked property in profile
American	company	* based in America
American	composer	* born in America
American	novel	* written in America
jazz	album	* features jazz
jazz	composer	* writes jazz
jazz	venue	jazz performed at *

Table 44: Head-specific property profiles learned by relating instances of H to the modifier M (I_{inst}). Results are similar using I_{head}.

Qualitatively, among the top-ranked interpretations, property profiles obtained using I_{head} (Equation 5.3) are nearly identical to those learned using I_{inst} (Equation 5.4). However, I_{inst} returns many more properties than does I_{head}. Specifically, I_{inst} returns 194M properties for a total of 19M MHs, or about 10 properties per MH on average, compared to an average of just over one property per MH returned by I_{head}. Anecdotally, we see that I_{inst} captures many more specific properties than does I_{head}. Many of these are properties which *entail* MH, but are not entailed *by* MH. For example, for *"jazz composers"*, both I_{head} and I_{inst} return the properties *"* write jazz"* and *"* compose jazz"*, but I_{inst} additionally returns properties like *"* create new blueprint for jazz"*. These more specific properties are effective for identifying instances of MH (Section 5.4), but are less accurate in capturing the intrinsic meaning of M. We compare I_{head} and I_{inst} quantitatively in Section 5.4.

rice dish	jazz composer	still life painter
* combine meat with rice	* match great verse with jazz	* find source material for
* be pork chop over rice	* create new blueprint for jazz	still life be signed work by *
rice be staple item in *	* surpass limit of jazz	* follow theory of still life

Table 45: Examples of properties learned by I_{inst} that are not learned by I_{head}. These are properties which entail MH, but are not necessarily entailed by MH.

5.3. Class-Instance Identification

In this section, we turn our attention to the task of class-instance identification. In general, this task is defined as: given the a class label C (e.g. *"American composer"*) and an entity e (e.g. *"Charles Mingus"*), determine whether or not $e \in C$. We frame the task in terms

of modifier-noun composition, by assuming the class label C is of the form MH. Then the definition becomes: given an entity $e \in H$, determine whether $e \in MH$. Thus, we aim to use the property profiles for M relative to H, as constructed in Section 5.2, in order to determine the set denoted by MH, as defined by Equation 5.2.

5.3.1. Class Membership as a Real-Valued rather than Binary Attribute

Let us re-frame Equation 5.2 so that the decision of whether "e is M" is made by calling a function ϕ_M, parameterized by the class H within which e is being considered:

$$[\![MH]\!] = [\![M]\!](H) = \{e \in H \mid \phi_M(H, e)\} \tag{5.7}$$

In theory, ϕ_M is a binary function which returns true if and only if e has the properties entailed by M in the context of H. In practice, we will instantiate the above equation using a real-valued function $\hat{\phi}_M$, which returns a score reflecting the likelihood that e has the properties necessary to be part of the class MH. For notational convenience, let $\mathcal{D}(\langle s, r, o\rangle) = w$, if $\langle s, r, o, w\rangle \in \mathcal{D}$ and 0 otherwise. We define $\hat{\phi}_M$ as follows:

$$\hat{\phi}_M(H, e) = \sum_{\langle\langle r,o\rangle,w\rangle \in I(MH)} w \times \mathcal{D}(\langle e, r, o\rangle) \tag{5.8}$$

where I is either I_{head} or I_{inst} as defined in Section 5.2. That is, for a given $e \in H$, $\hat{\phi}_M$ returns a score which is simply a weighted sum of all the properties in the property profile of M relative to H which hold for e according to the fact database $\mathcal{D}$.

Applying M to H, then, is as in Eq. 5.7 except that instead of a discrete set, it returns a scored list of candidate instances:

$$[\![M]\!](H) = \{\langle e, \hat{\phi}_M(H, e)\rangle \mid \langle e, H\rangle \in \mathcal{O}\} \tag{5.9}$$

Ultimately, we need to identify instances of arbitrary class labels, which may contain multiple modifiers. Given a class label $C = M_1 \ldots M_k H$ which contains a head H preceded by

modifiers $M_1 \dots M_k$, we generate a list of candidate instances by finding all instances of H which have some property to support every modifier:

$$\bigcap_{i=1}^{k} \{\langle e, score(e)\rangle \mid \langle e, w\rangle \in [\![M_i]\!](H) \wedge w > 0\} \tag{5.10}$$

where $[\![M_i]\!](H)$ is as in Equation 5.9 and $score(e)$ is simply the average score over all of the modifiers in C, as below:

$$score(e) = \frac{1}{k} \sum_{i=1}^{k} \hat{\phi}_{M_i}(H, e) \tag{5.11}$$

5.3.2. Weakly Supervised Scoring Model

In Equation 5.8, the confidence we assign for $e \in MH$ is based on a weighted sum which is simply product of the weight of a property in the property profile of M relative to H and the raw number of times that e has been observed as having that property, according to $\mathcal{D}$. This naive calculation has the weakness that instances of H with overall higher counts in $\mathcal{D}$ are scored highly by $\hat{\phi}_M$ regardless of MH. To remedy this, we train a simple logistic regression model to predict the likelihood that e belongs to MH.

As training data, we take a random sample of $\langle e, MH\rangle$ pairs from $\mathcal{O}$ and consider these to be positive training examples. We select another sample of $\langle e, MH\rangle$ pairs which do not appear in any Hearst pattern in our Web corpus and consider these to be negative training examples. That is, we have a stricter requirement for our negative training data than simply not appearing in $\mathcal{O}$: $\mathcal{O}$ includes all $\langle e, C\rangle$ tuples which are supported by at least 5 sentences in our corpus (Section 5.2.2), and our negative training data comes only from $\langle e, C\rangle$ tuples which were supported by 0 sentences in the corpus. The resulting training set contains 3M pairs of which 45% are positive and the remaining are negative.

We frame the task as a binary prediction of whether $e \in C$. We use a handful of features, including the total number of categories in $\mathcal{O}$ of which e is an instance and the total number of tuples in $\mathcal{D}$ in which e is the subject. Full feature templates are given in Appendix A.6.

We train a standard logistic regression model implemented in the scikit-learn Python toolkit (`http://scikit-learn.org`). We tune the regularization parameter using cross validation on the training data. On cross validation, the trained model achieves 65% accuracy over the 45% majority class baseline.

We use the trained model's predicted probability of the positive class (i.e. the probability that $e \in MH$) as the value of $\hat{\phi}_M$ in Equation 5.9, in place of the sum defined in Equation 5.8. Table 46 provides some comparisons of ranked lists of entities for a given class when using the naive computation of $\hat{\phi}_M$ (Equation 5.8) and when using the score produced by the trained logistic regression. As shown, the naive model has a tendency to inflate the scores for entities which appear as instances of the head (i.e. $\langle e, H \rangle \in \mathcal{O}$) and have overall high counts in $\mathcal{D}$, even if those entities have overall low evidence for the particular modifiers appearing in the class label. The trained model effectively down weights such entities.

electronic signature provider		Russian art critic	
Weighted Sum	Log. Reg.	Weighted Sum	Log. Reg.
UnitedHealth Group	DocuSing	Jon Stewart	Edmund Wilson
Ascertia	Crossgate AG	Ronald Reagan	Denis Diderot
US Dept of VA	HelloSign	Glenn Beck	Viktor Shklovsky
Pacific G&E Co.	DocuWare	Václav Havel	Walter Benjamin
Aetna	Softpro	Benjamin Netanyahu	Mikhail Bakhtin

Table 46: Top-ranked entities for a given class according to the naive score model (defined in Equation 5.8) and according to a weakly-supervised logistic regression model.

5.3.3. Summary of Proposed Methods and Variations

To summarize our proposed method for class-instance identification: given an entity e and a class label $C = M_1 \dots M_k H$ which consists of a head noun preceded by at least one and possibly many modifiers, we return a real-valued score w reflecting the likelihood that $e \in C$. We compute w using Equation 5.11. This equation depends on $\hat{\phi}_{M_i}(e, H)$, our confidence that "e is M_i" for each of the M_i in C, which in turn depends on the property profiles for each of the M_i relative to H. We may compute these property profiles either using facts which relate the head H to a given M directly (I_{head} defined by Equation 5.3) or using

	Final Score (for $e \in C$)	Property Profiles (for each M_iH)	Aggregation (for $\hat{\phi}_{M_i}(e, H)$)
Mods_H Raw	Eq. 5.11	I_{head} (Eq. 5.3)	Weighted sum (Eq. 5.8)
Mods_I Raw	Eq. 5.11	I_{inst} (Eq. 5.4)	Weighted sum (Eq. 5.8)
Mods_H Reranked	Eq. 5.11	I_{head} (Eq. 5.3)	Logistic Regression (§5.3.2)
Mods_I Reranked	Eq. 5.11	I_{inst} (Eq. 5.4)	Logistic Regression (§5.3.2)

Table 47: Summary of model variations proposed for the task of class-instance identification given a class label $C = M_1 \dots M_k H$ and an entity e.

facts which relate instances of H to M (I_{inst} defined by Equation 5.4). In addition, given e and a property profile for M_i relative to H, we may determine the final score for whether $e \in M_iH$ (that is $\hat{\phi}_{M_i}(e, H)$) using a naive weighted sum, as defined in Equation 5.8, or using a trained logistic regression, as described in Section 5.3.2. Thus, we propose the four variations shown in Table 47, which we evaluate against several baselines in Section 5.4.

5.4. Evaluation

In this section, we evaluate our proposed methods. Specifically, given an arbitrary class label C, which contains at least one and potentially many modifiers, and a large corpus of natural language text, we evaluate each method on its ability to return the list of entities which belong to the class C.

5.4.1. Evaluation Data Sets from Wikipedia

We derive our gold-standard evaluation data from Wikipedia category pages[12]. These are pages in which the title is the name of a category (e.g., *"Pakistani film actresses"*) and the body is a manually-curated list of links to other pages which fall under the category. We consider the title to be a class label and the list of links on the page to be the gold-standard list of entities belong to the class.

To build our evaluation sets, we begin with the set of titles of all Wikipedia category pages. We remove those in which the last word is capitalized, a heuristic intended to

[12] `http://en.wikipedia.org/wiki/Help:Category`

retain only class labels in which the head is a single common noun: e.g. *"South Korea"* is the title of a category page which links to pages such as *"South Korean culture"* and *"Images of South Korea"*. In addition, we remove titles containing fewer than three words as well as titles of pages which contain links to sub-categories. These filters are intended to favor compositional class labels containing multiple modifiers (*"Pakistani film actresses"*) as opposed to coarser-grained ones (*"film actresses"*).

Although all of the remaining class labels contain at least three words, they represent a mix of single modifier (*"Puerto Rican sculptors"*) and multiple modifier (*"Canadian business journalists"*) phrases. We therefore perform heuristic noun-phrase chunking as a preprocessing step. We use a constituency parser trained to parse queries (Petrov et al. (2010)), which gives good performance on short phrases. Given the parse tree, we group together any tokens which share a common parent other than the root node, with the exception of the rightmost token (the head), which we force to appear as a chunk by itself. This heuristic was chosen since, on manual inspection, it produced good chunks. We use these pre-chunked class labels as input to all of the systems, including baselines, in our evaluation.

From the resulting list of class labels, we draw two samples of 100 labels each, enforcing that no H appear as the head of more than three class labels per sample. The first sample is chosen uniformly at random (denoted UNIFORM). The second (WEIGHTED) is weighted so that the probability of drawing $M_1 \ldots M_k H$ is proportional to the total number of class labels in which H appears as the head. These different evaluation sets are intended to evaluate performance on the head versus the tail of class label distribution, since information retrieval methods often perform differently on different parts of the distribution. On average, there are 17 instances per class in UNIFORM and 19 in WEIGHTED. Tables 48 and 49 give example class labels from each dataset.

2008 california wildfires · australian army chaplains · australian boy bands · canadian business journalists · canadian military nurses · canberra urban places · cellular automaton rules · chinese rice dishes · coldplay concert tours · daniel libeskind designs · economic stimulus programs · german film critics · invasive amphibian species · log flume rides · malayalam short stories · pakistani film actresses · puerto rican sculptors · string theory books · tampa bay devil rays scouts

Table 48: Examples of class labels from UNIFORM.

2face idibia albums · ancient greek physicists · ancient spartan soldiers · art deco sculptors · crisis pregnancy centers · data modeling tools · east german sprinters · indian bass guitarists · indoor roller coasters · international water associations · iomega storage devices · jerusalem prize recipients · latin logical phrases · new urbanism communities · new zealand illustrators · newell rubbermaid brands · north american football league teams · pakistani cricket captains · scottish football referees · southern cross flags

Table 49: Examples of class labels from WEIGHTED.

5.4.2. Experimental Setup

We compare the following models on their ability to retrieve instances for a given class label ($C = M_1 \ldots M_k H$), using only the information available in a corpus of raw natural language text. For all of our experiments, we use a corpus of approximately 1 billion English Web documents.

Baseline Methods

We test three different baseline models for class-instance identification. Our simplest baseline (referred to simply as **Baseline**) ignores modifiers altogether, and simply assumes that any instance of H is an instance of MH, regardless of M. We implement Baseline using $\mathcal{O}$, and the confidence value for whether $e \in M_1 \ldots M_k H$ is equivalent to the confidence value assigned to $\langle e, H \rangle$ in $\mathcal{O}$.

Our second, stronger baseline (**Hearst**) uses the lexico-syntactic patterns used in the construction of $\mathcal{O}$ to directly identify instances of the . That is, in Hearst, the confidence value for whether $e \in C$ is equivalent to the confidence valued assigned to $\langle e, C \rangle$ in $\mathcal{O}$. Thus, for Hearst to assign any score to e as an instance of C, the entire class label $C = M_1 \ldots M_k H$

must have appeared in some sentence in the corpus matching some Hearst pattern.

Finally, we test a baseline compositional model (**Hearst∩**), in which we augment the Hearst baseline via set intersection. Specifically, for a class $C = M_1 \dots M_k H$, if each of the $M_i H$ appears in $\mathcal{O}$ independently, we take the instances of C to be the intersection of the instances of each of the $M_i H$. We assign the weight of an instance e to be the sum of the weights associated with each independent modifier. Note that while this method is compositional, it fails to meet the first of the desiderata outlined Section 5.1; i.e. Hearst∩ does not assign any intrinsic meaning to the modifiers.

Proposed Methods

We evaluate each of the four variations of our proposed methods as described in Table 47, namely: **Mods$_H$ Raw**, **Mods$_I$ Raw**, **Mods$_H$ RR**, and **Mods$_I$ RR**, where RR stands for "reranked".

Hybrid Methods

We experiment with using the proposed methods to extend rather than replace the Hearst baseline. We combine predictions of different models by merging the ranked lists produced by each system: i.e. the score of an instance is the inverse of the sum of its ranks in each of the input lists, i.e. $rank_{merged} = (rank_{list1} + rank_{list2})^{-1}$. If an instance does not appear at all in an input list, its rank in that list is set to a large constant value. We refer to these combination systems by concatenating the names of the input systems, e.g. **Hearst+Mods$_H$ RR** and **Hearst+Mods$_I$ RR**.

5.4.3. Results and Analysis

We now compare the described methods in terms of their precision and recall for returning a list of instances given a class label, using the datasets described in Section 5.4.1.

Reliability of Wikipedia as Gold Standard

While we use Wikipedia as a gold standard throughout our evaluations below, it possible that there are true instances of a class that are missing from our Wikipedia reference set and that our precision scores may underestimate the actual precision of the systems. In order to assess the extent to which this is the case, we manually verify the top 10 predictions of each of the systems for a random sample of 25 class labels. We choose class labels for which Hearst was able to return at least one instance, in order to ensure reliable precision estimates. For each of these labels, we manually check the top 10 instances proposed by each method to determine whether each belongs to the class. Table 50 shows the precision scores for each method computed against the original Wikipedia list of instances and against our manually-augmented list of gold instances. The overall ordering of the systems does not change, but the precision scores increase notably after re-annotation. We thus continue to evaluate against the Wikipedia evaluation sets as constructed in Section 5.4.1, but acknowledge that reported precision scores for each system are likely an underestimate of true precision.

	Wikipedia	Manual
Hearst	0.56	0.79
Hearst$\cap$	0.53	0.78
Mods$_H$ RR	0.23	0.39
Mods$_I$ RR	0.24	0.42
Hearst+Mods$_H$ RR	0.43	0.63
Hearst+Mods$_I$ RR	0.43	0.63

Table 50: Precision@10 for several methods computed using Wikipedia as the definitive gold standard and computed using manually-augmented gold standard reference sets.

Comparison of Methods

We first compare the methods in terms of their coverage, the number of class labels for which the method is able to find some instance, and their precision, to what extent the method is able to correctly rank true instances of the class above non-instances for both the UNIFORM and WEIGHTED evaluation sets. We report total coverage, the number of labels for which

the method returns any instance, and correct coverage, the number of labels for which the method returns a correct instance. For precision, we report mean average precision (MAP), which is the mean of the average precision (AP) scores across all the class labels. AP ranges from 0 to 1, where 1 indicates that all positive instances were ranked above all negative instances. Note that MAP is only computed over class labels for which the method returns something, meaning methods are not punished for returning empty lists.

Table 51 shows the precision and coverage for each of the methods. The proposed compositional models show consistently better coverage than the non-compositional baselines. This is expected, as the proposed models do not have the restrictive requirement that the entire class label appears verbatim in the text corpus within one of the specified lexico-syntactic patterns. As a result, the proposed models are able to make use out of a much larger set of sentences than can the Hearst baselines. However, this increase in coverage comes with a tradeoff in precision, with the proposed models exhibiting significantly lower MAP than the baseline methods.

	Uniform			Weighted		
	Total Cov.	Correct Cov.	MAP	Total Cov.	Correct Cov.	MAP
Baseline	95%	70%	0.01	98%	74%	0.01
Hearst	9%	9%	0.63	8%	8%	0.80
Hearst$\cap$	13%	12%	0.62	9%	9%	0.80
Mods_H raw	56%	32%	0.23	50%	30%	0.16
Mods_H RR	56%	32%	0.29	50%	30%	0.25
Mods_I raw	62%	36%	0.18	59%	38%	0.20
Mods_I RR	62%	36%	0.24	59%	38%	0.23

Table 51: Total coverage, correct coverage, and mean average precision for each method when identifying instances of arbitrary classes. Class labels are derived from titles of Wikipedia category pages.

Figure 19 illustrates how the single MAP score (as reported in Table 51) can misrepresent the relative precision of different methods. Specifically, since MAP is computed only over classes for which the method returns at least one instance, the baseline methods are not penalized for the many classes for which they are unable to return any instances. Thus,

overall, we see that the proposed methods extract instances about as well as the baseline, whenever the baseline can extract anything at all; i.e. the proposed method does not cause a precision drop on classes covered by the baseline. However, the proposed method is additionally able to identify instances (albeit at lower precision) for many classes for which the baseline returns nothing.

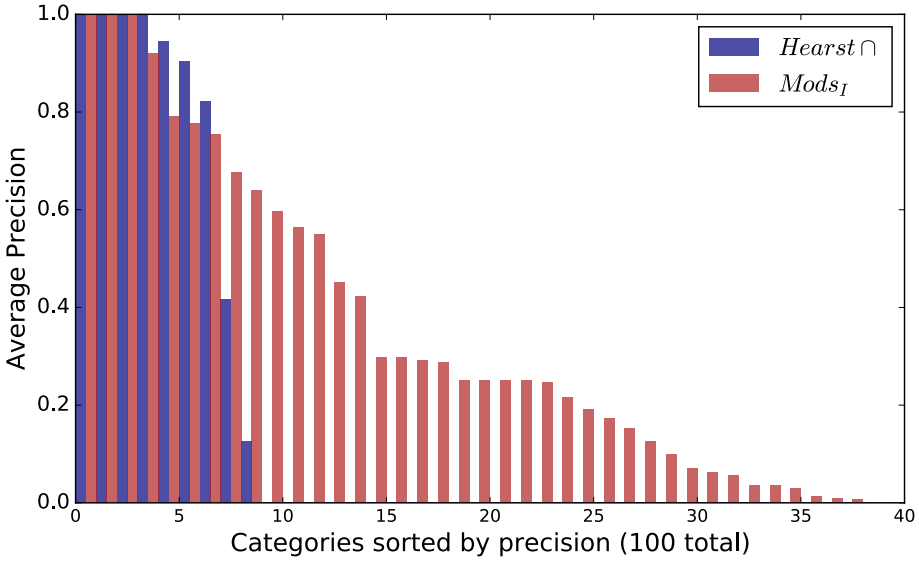


Figure 19: Distribution of AP over 100 class labels in Weighted.

Table 52 gives anecdotal examples of instances returned for several class labels by the proposed Mods$_I$ method. For the classes shown, Hearst∩ has high precision but very low coverage (i.e. only one or two instances per class) while the proposed method identifies a much larger set of instances, many of which are correct but some of which are not.

Finally, we look at the precision-recall tradeoff in terms of the area under the curve (AUC) achieved when each method attempts to rank the complete list of candidate instances. We take the union of all of the instances proposed by all of the methods (including the Baseline method which, recall, proposes every instance of the head H as an instance of the class $M_0 \ldots M_k H$). Then, for each method, we rank this full set of candidates such that any instance returned by the method is given the score the method assigns, and every other instance is scored as 0.

Flemish still life painters	Pakistani cricket captains	Thai Buddhist temples
Clara Peeters	Salman Butt	Wat Buddhapadipa†
Willem Kalf	Shahid Afridi	Wat Chayamangkalaram
Jan Davidsz de Heem	Javed Miandad	Wat Mongkolratanaram†
Pieter Claesz†	Azhar Ali	~~Angkor Wat~~
~~Peter Paul Rubens~~	~~Greg Chappell~~	~~Preah Vihear Temple~~
Frans Snyders	Younis Khan	Wat Phra Kaew
Jan Brueghel the Elder	Wasim Akram	Wat Rong Khun
~~Hans Memling~~	Imran Khan	Wat Mahathat Yuwaratrangsarit
~~Pieter Bruegel the Elder~~	Mohammad Hafeez	~~Vat Phou~~
~~Caravaggio~~	Rameez Raja	Tiger Temple
Abraham Brueghel	Abdul Hafeez Kardar	Sanctuary of Truth

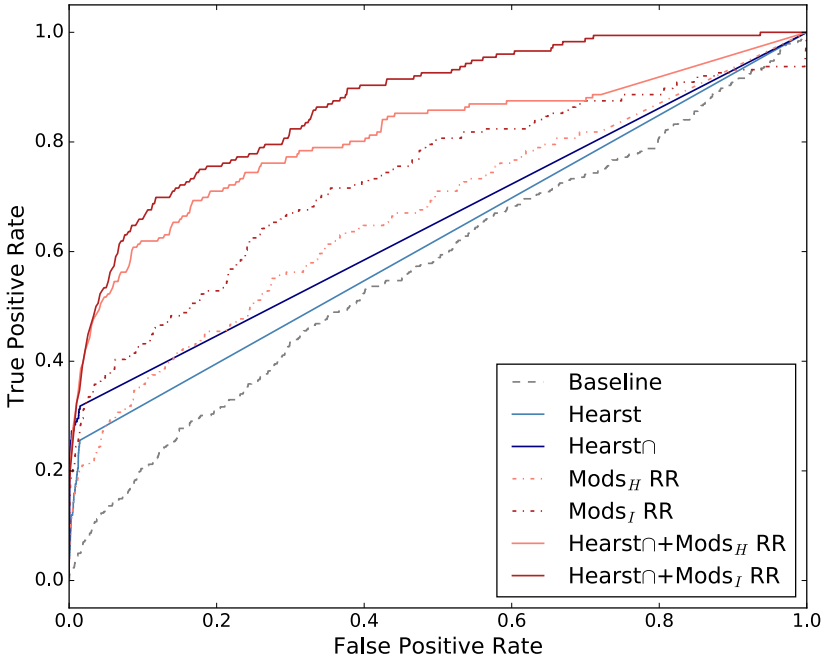
Table 52: Instances extracted for several fine-grained classes using Mods_I. † denotes the instance was also returned by Hearst∩. Strikethrough denotes the instance is incorrect.

	Uniform		Weighted	
	AUC	Recall	AUC	Recall
Baseline	0.55	0.23	0.53	0.28
Hearst	0.56	0.03	0.52	0.02
Hearst∩	0.57	0.04	0.53	0.02
Mods_H RR	0.68	0.08	0.60	0.06
Mods_I RR	0.71	0.09	0.65	0.09
Hearst∩+Mods_H RR	0.70	0.09	0.61	0.08
Hearst∩+Mods_I RR	0.73	0.10	0.66	0.10

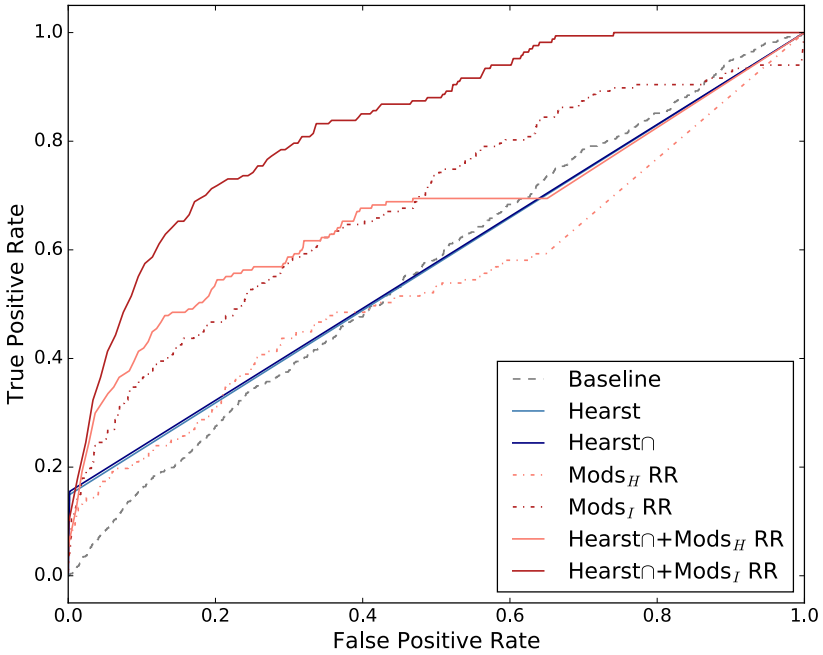
Table 53: Recall of instances on Wikipedia category pages, measured against the full set of instances from all pages in sample. AUC captures tradeoff between true and false positives.

Table 53 reports the AUC and recall and Figure 20 plots the full ROC curves. Given a ranked list of instances, ROC curves plot true positives vs. false positives retained by setting various cutoffs. Note that, in Figure 20, the curve becomes linear once all remaining instances have the same score (e.g., 0), as this makes it impossible to add true positives without also including all remaining false positives.

Recall that the Hearst baselines require that a class label appear in full in a single sentence in order to return a non-zero score for any instance of that class. As a result, the vast majority of candidate instances receive a score of 0 from these methods, which translates into very low AUC. By comparison, the proposed compositional methods can make use



(a) Uniform random sample (UNIFORM).



(b) Weighted random sample (WEIGHTED).

Figure 20: ROC curves of various methods for ranking instances based on likelihood of belonging to a specified class.

of a larger set of sentences, and can provide non-zero scores for many more candidates. This improved coverage results in a >10 point increase in AUC on both the UNIFORM and WEIGHTED evaluation sets.

5.5. Discussion

In this chapter, we proposed a concrete method for operationalizing the formal semantics definition of modifier-noun composition. We argued that this formulation has several practical and theoretical advantages for NLP systems. Specifically, it enables us to assign intrinsic semantics to the modifier itself and it enables us to make inferences in a truth-theoretic context. We applied our method to the task of class-instance identification for fine-grained classes for which the class labels consisted of at least one and possibly multiple modifiers. Our experimental results demonstrated that the proposed compositional method outperforms existing non-compositional ones.

The method we proposed here is arguably the simplest means of instantiating the formal semantics framework discussed. That is, we made many assumptions and implementation decisions which simplified our proposed method, but which could likely be revisited in order to improve performance. One such assumption was that frequently-expressed relations between M and H will capture the most salient properties of MH. This assumption enabled us to learn property profiles without any supervision or any known instances of MH, and, very often, this assumption led to good results. However, we saw examples (e.g. *"child actor"*, *"risk manager"*) for which the learned profiles were incorrect. A logical direction for improving the proposed method would be to look not at the frequency of a property, but rather at a property's discriminative power in separating instances of MH from instances of H. Such a model would require known instances of MH, but would more directly access what it is we expect the property profiles to capture.

In addition, our proposed model conflated two aspects of modifier meaning which, while related, should ideally be kept different. Namely, we did not differentiate between properties

that are entailed by the modifier and those which simply entail the modifier. That is, we did not differentiate between $\equiv$ and $\sqsupset$ relations as they pertain to modifiers and properties. For example, while *"jazz composer"*$\equiv$*"composer who writes jazz"*, *"jazz composer"*$\sqsupset$*"composer who creates a new blueprint for jazz"*. For the task of class-instance identification, this distinction is not particularly important. In fact, we could likely improve our recall further by specifically targeting $\sqsupset$ properties: e.g. that a *"composer born in New York"* is an *"American composer"*. For other tasks which might be more sensitive to the meaning of the modifier itself, for example in an RTE setting, differentiating between properties which are strictly entailed and those which are not will be important.

Finally, for simplicity, we focused on subsective modifiers and on class labels which clearly decomposed into modifiers and heads. This enabled us to, for example, consider only entities in the set of *"composers"* when searching for *"American composers"*. However, the framework described could be extended to operate much more generally. First, we could consider associating properties with common nouns directly, in order to enhance our models of lexical entailment, like those in Chapter 3, to capture semantics beyond set containment: e.g. what are the properties that differentiate the subclass *"composers"* from the superclass of *"people"* more generally? Going further, we could consider using the framework in order to associate properties with non-subsective modifiers, like those in Chapter 4: e.g. what are the properties that differentiate the class of *"former senators"* from the class of *"senators"*. The learned properties (for both subsectives and non-subsectives) could be applied in an RTE setting, like that discussed is the previous chapter.

Broadly speaking, the framework presented in this chapter provides a small first step toward building more robust models of the lexical entailments, which focus on the properties entailed by words rather than the set-theoretic relationships between words. Such models, properly implemented, could enable a much more principled treatment for the types of context-dependent inference problems presented in Chapter 4, for which simple set-containment inference is ill-suited.

CHAPTER 6 : Conclusion

Recent advances in natural language processing have been driven by a focus on statistical models of language which can be learned from large amounts of data. There is no denying that these shallow models have demonstrated great practical success on a variety of down-stream tasks, including information retrieval, machine translation, and speech recognition. However, as we have discussed throughout, many aspects of language require a more subtle understanding of semantic meaning than that which can be easily derived from word co-occurrence alone. As the field of NLP moves toward addressing increasingly complex tasks, requiring more nuanced inference and closer interaction with humans, it will become important that the models we build are both precise and transparent regarding the inferences they make. The goal of this thesis has been to use insights from formal and experimental linguistics to inform the models we build, so that we can continue to take advantage of the convenience and robustness of large-scale statistical models of language without compromising on depth or clarity in our semantic representations.

6.1. Summary of Contributions

We began with a discussion of lexical entailment. There exist a number of data-driven methods for inferring whether words or phrases have "similar" meanings. Among the most popular are those based on *distributional semantics*, in which words are considered to mean roughly the same thing if they tend to be used in similar contexts, and those based on *bilingual pivoting*, in which words are considered to mean roughly the same thing if they can be translated to the same word in a foreign language. These methods are effective at finding sets of related words, but the precise semantics of how the words are related is not clear. For example, if x and y are close in distributional space, we can conclude only that *x occurs in similar contexts to y*; if x and y are associated via bilingual pivoting, we can conclude only that *x shares at least one translation with y*. Neither of these relations is meaningful or useful for natural language understanding more generally.

In Chapter 3, we built a statistical classifier to differentiate between a handful of distinct semantic entailment relations. Our classifier assigned each pair of words to be related by either Equivalence (*"couch"*$\equiv$*"sofa"*), Entailment (*"couch"*$\sqsubset$*"furniture"*), Exclusion (*"couch"*$\dashv$*"table"*), or Independence (*"couch"*$\#$*"antique"*). We used this classifier to assign an interpretable entailment relation to each of the phrase pairs in the Paraphrase Database (PPDB), a large-scale paraphrase resource extracted automatically via bilingual pivoting. We demonstrated that the resulting resource–PPDB augmented with these automatically-assigned entailment relations–led to improved performance for an end-to-end system on the task of recognizing textual entailment, outperforming both the unannotated version of PPDB and WordNet, and manually-constructed lexical entailment resource.

We then turned our attention to compositional entailment, in particular, to modifier-noun composition. Existing work in NLP and in linguistics has largely focused on modeling the meaning of modifier-noun phrases in terms of the set theoretic denotations of the noun phrase. This focus on sets and denotations has lead to the conventional wisdom that inferences involving modifier-noun phrases is by-and-large governed by the class to which the modifier belongs: subsective modifiers like *"red"* generate Forward Entailment ($\sqsubset$), plain non-subsective modifiers like *"alleged"* generate Independence ($\#$), and privative modifiers like *"fake"* generate Exclusion ($\dashv$).

In Chapter 4, we showed that, in fact, inferences involving modifier-noun phrases are much more complex than previously assumed. We performed an in-depth study of human inferences surrounding modifier-noun composition and revealed that common sense assumptions and pragmatic reasoning, rather than strict logical inference, very often dictates the inferences that humans make. Through a systematic evaluation of a range of existing RTE systems, we illustrated the inability of current NLU technology to handle these types of common sense inferences. We concluded that, overall, automatic systems for natural language inference are unlikely to see a significant benefit to differentiating between modifiers based on the established formal semantics class. Rather, our observations suggest that real

improvements will likely involve explicit computational models of conversational implicature and of clear but cancelable common sense assumptions regarding the entities and events about which the systems are expected to reason.

We observed that, in order to improve systems' ability to reason about modifier-noun compositions, we require models that permit reasoning about entailment beyond simple semantic containment. Specifically, we need to explicitly model the intrinsic meaning and entailments of the modifiers themselves. Formal semantic theory provides a basic framework in which modifiers are functions which operate on the denotations of nouns. This framework is appealing in that it provides both an explicit representation of the meaning and entailments of the modifier as well as process for invoking that representation in order to make truth-theoretic inferences. Existing treatments of modifier-noun composition in NLP have not effectively addressed these two aspects simultaneously.

In Chapter 5, we proposed a data-driven method for operationalizing this formal semantics framework. This method first assigns explicit and interpretable semantic representations to individual modifiers and then uses these representations to infer whether or not a particular entity (e.g. *"Charles Mingus"*) belongs to a specific set (e.g. the set of *"1950s American jazz composers"*). We evaluated our proposed method by using it to find instances of fine-grained classes involving multiple modifiers (e.g. *"Pakistani film actresses"*, *"invasive amphibian species"*). We demonstrated that approaching the task compositionally–by considering the semantics of each modifier in the class label individually–leads to a significant improvement in recall over existing non-compositional approaches which treat the entire class label as though it is a single atomic symbol. Moreover, we argued that our proposed compositional model is better poised for use in more complex language understanding tasks which require general reasoning about modifiers for purposes other than class-instance identification.

6.2. Discussion and Future Directions

A central theme that has arisen throughout the experiments and discussions in this thesis that that of semantic versus pragmatic reasoning. This is a much larger question that sits at the heart of much of the work in linguistics and in philosophy of language generally. In natural language processing, it manifests very concretely as: how much of the required work of NLP systems can be pre-computed, and what must necessarily be handled at runtime? The fact that, in NLP, this can be seen as an engineering question rather than a philosophical one does not change that fact that, ultimately, it must be addressed. Determining where this line sits, i.e. determining which components of word meaning are task-independent and how those components are to be represented and stored, is one of the major open questions to be addressed by future work in computational semantics.

At the beginning of this thesis, in our discussion of lexical entailment, we took a simplistic but practical approach in which we assumed the entirety of lexical entailment could be pre-computed and stored in a database. As such, we ignored important issues like word sense and context when assigning semantic entailment relations. In Chapter 4, we moved toward the other extreme, arguing that in fact very little of the (semantic containment) properties associated with modifiers could be precomputed, and that the primary process governing whether or not a modifier can be inserted into a sentence is a pragmatic one. Finally, in Chapter 5, we moved back toward the semantics side, in which we sought to pre-compute and store the properties entailed by individual modifiers. However, in Sections 4.6 and 5.5 we outlined potential ways in which that these pre-computed representations could be applied in context-specific inferences, by comparing the entailed properties of modifiers to the (explicitly stated or pragmatically inferred) properties of the entity being discussed.

Thus far in NLP, pragmatic inference has received very little attention in comparison to syntax and semantics. However, as we have discussed throughout this thesis, pragmatic processes are likely to play a central role in natural language inference systems and should be a priority in future work. Even, and especially, if we treat NLP as primarily an engineering

endeavor, the goal of which is to build computational systems to understand language, then we care deeply about which mechanisms most accurately predict human inferences in practice. This could be dismissed as a question for linguistics, cognitive science, and philosophy. But it is also a engineering question. We care what these mechanisms are, whether they are computable, whether they can be learned efficiently, how they represent meaning in memory, and how they access memory at runtime. We will not be able to significantly advance the state of automatic natural language understanding beyond the tasks we currently perform until we begin to address these questions.

A focus on pragmatic processes may in fact simplify the computation we ask systems to perform, at cost of requiring greater investment in representation. It has been suggested that many complex pragmatic inferences such as metaphor can be explained by simple Bayesian processes (Goodman and Frank (2016)), and the results we presented in Chapter 4 suggest that inferences about literal language may be explained similarly. The computational implementations of such processes (Andreas and Klein (2016); Monroe et al. (2017)) are much simpler than the bottom-up logical approaches that have been explored previously in NLP (Bos and Markert (2006)), and an exciting direction of future work would be the application of these pragmatic processes to natural language inference more generally. However, such processes, while probabilistic, are still symbolic. The current state-of-the-art methods for representing sentence meaning for natural language inference (Bowman et al. (2015); Rocktäschel et al. (2016)) do not learn precise representations that can be applied symbolically. We saw from the results of our evaluation in Chapter 4 that these systems, even in the transfer learning setting, did not learn the required abstraction to support the type of precise probabilistic reasoning we want to perform. Specifically, as is, these systems did not learn to represent the concept of discrete nouns (entities) and their modifiers (properties). Nonetheless, these data-driven models are powerful and exciting. Thus, an important direction for future work is how to coach such data-driven models toward learning a more general representation that is consistent with the type of flexible yet still crisp representations that humans seem to use when performing natural language inference.

APPENDIX

A.1. Comparison of Lexical Entailment Annotation HIT Designs

In this Appendix, we compare three task designs for the annotation of basic lexical entailment relations. We show that there are no significant disadvantages to using our chosen design over the reasonable alternatives.

A.1.1. Designs

In these pilot experiments, we asked workers to classify a pair of lexical expressions according to one of the six relations in Table 54. Note that these are not as clean as the relations defined in Section 2.4 and used for the experiments in the main body of this thesis.

E	Equivalent: The words have the same meaning. Ex. car/automobile.
G	More General: The first word generalizes the second word. Ex. dog/dalmatian.
S	More Specific: The first word is a more specific form of the second word. Ex. dalmatian/dog.
X	Mutually Exclusive: The first word is the opposite of or is mutually exclusive with the second. Ex. relevance/irrelevance.
N	No Relation: The words are not related. Ex. car/dragon.
Q	I cannot tell: One or both of the words are in a language other than English, or are not understandable for some other reason.

Table 54: Six entailment relations used to classify word pairs during pilot study on task design.

We consider the below three HIT designs:

Isolation HIT. Annotators are shown two words side-by-side (Figure 21) and asked to label their relationship, choosing between the six relation types from Table 54.

somebody ____ girlfriend

Figure 21: Pair of words as presented to annotators in the Isolation HIT.

Context HIT. Annotators are shown two words side-by-side along with an example context and asked to make the same judgement (Figure 22). Specifically, they are shown one sentence of context which contains the first word of the pair being judged. Workers were asked to judge the relationship of the first word to the second word in the supplied context, and to consider only the meaning, not the grammaticality of the words in the context.

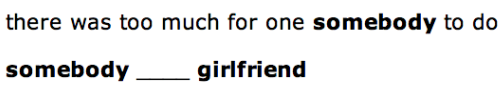


Figure 22: Pair of words as presented to annotators in the Context HIT and the Two Pass HIT.

Two Pass HIT. Annotators are shown two words side-by-side along with an example context and asked to make the same judgement (Figure 22), but the task is split into two steps. First, annotators are asked to make a coarse judgement on the relation, choosing between only three relation types. Based on the results of the first step, a second set of workers are asked to make a finer-grained judgement on the words' relation type. Specifically, in the first pass, annotators are asked to choose between semantic containment, semantic exclusion, and unrelated. Pairs for which the majority of first-pass annotators chose containment are shown to annotators in the second-pass. In the second step, annotators are asked to distinguish between equivalence, forward entailment, and reserve entailment. These options are shown in Table 55.

A.1.2. Setup

Task Parameters. Workers were shown 18 or 19 pairs per HIT. Workers were instructed to use choose "Equivalent" for words which differed only in spelling, and to choose "I cannot tell" if one of the words was an unfamiliar acronym, a foreign word, or if they felt they could not make a judgement for some other reason.

We posted 28 HITs in each of the isolation HIT, the context HIT, and the first pass of the two-pass HIT. In each design, we gathered labels for the same sample of 508 noun pairs with

Pass 1	
C	Containment: The words have equivalent or nearly equivalent meaning.
X	Mutually Exclusive: The first word is the opposite of or is mutually exclusive with the second.
N	No Relation: The words are not related. Ex. car/dragon.
Q	I cannot tell: One or both of the words are in a language other than English, or are not understandable for some other reason.
Pass 2	
E	Equivalent: The words have the same meaning. Ex. car/automobile.
G	More General: The first word generalizes the second word. Ex. dog/dalmatian.
S	More Specific: The first word is a more specific form of the second word. Ex. dalmatian/dog.

Table 55: Options for relations in first pass (top) and second pass (bottom) of the two-pass HIT. Pairs for which the majority label in the first pass was "equivalent (or nearly equivalent)" were shown to Turkers in the second pass in order to receive a more fine-grained label.

known labels (as described in the following section). Our chosen pairs were approximately evenly distributed between the five entailment relationships. For the two pass HIT, 306 noun pairs were judged to be "entailment" in the first pass and we gathered finer-grained labels for these in 17 second-pass HITs. We paid \$0.07 per HIT and had five Turkers label each noun pair. We made our HITs available only to US workers, and only to workers who had completed at least 50 HITs and had at least a 90% approval rate. Our assignments were completed by 108 unique workers, but 11 of these workers performed tasks in more than one of our designs. For simplicity, the work completed by these workers is removed from all of the analysis in this paper.

Gold Standard Data. We use our Wordnet as a source of gold-standard pairs belonging to the synonym, hypernym, hyponym, and antonym relations. We draw random pairs of nouns to compile a list of noun pairs for the independent relation. For the context HITs, we use Wordnet example sentences as our contexts. These example sentences are synset specific, which allows us to ensure that the context shown matches the sense of the word pair for which the gold standard label holds.

A.1.3. Findings

We calculate the accuracy of the majority label for each pair. We remove from the calculation responses of "I cannot tell", so that they do not count for or against the majority label. Table 56 shows that while the two-pass HIT has the highest overall accuracy, the results when broken down by relation type are mixed.

	Isolation	Context	Two Pass
Overall	0.52	0.48	**0.53**
Hyponym	**0.20**	0.16	0.15
Hypernym	**0.32**	0.25	0.29
Synonym	0.54	0.54	**0.67**
Antonym	**0.77**	0.72	0.70
Independent	0.96	0.96	**0.98**

Table 56: Accuracy of the majority label in each HIT design. Benefits of using the two-pass HIT design rather than the basic isolation design are not conclusive.

We also calculate the percent of annotators in agreement for relations of each type. Table 57 show the results.

	Isolation	Context	Two Pass
Hyponym	**32%**	25%	27%
Hypernym	**32%**	25%	24%
Synonym	56%	**62%**	58%
Antonym	69%	77%	**86%**
Independent	71%	64%	**77%**

Table 57: Percent of workers in agreement for word pairs of each relation type.

Overall, the HITs in which context is provided show lower accuracies (as measured against WordNet as a gold standard) but higher agreement. However, for the hypernym/hyponym relations, Turkers show the highest levels of agreement when given the word pairs in isolation. To satisfy curiosity, Table 58 shows several examples of WordNet hypernyms and hyponyms that humans considered to be synonyms in context.

Considering this analysis, we choose to use the Isolation HIT design, as it is simplest to implement and does not have substantial disadvantages over the other designs.

Most churches baptize infants but some insist on adult **baptism/christenings**
The **appointment/nomination** had to be approved by the committee
This storm is certainly an **alteration/transformation** for the worse
They collect the **waste/refuse** once a week
The **acquisition/takeover** of wealth

Table 58: Examples of hypernym/hyponym relationships labeled as synonyms. In the above examples, at least 3 out of 5 workers chose the label "synonym."

A.2. Instructions for Lexical Entailment HIT

Please choose the option which best describes the relation of the first word to the second word. Please focus **only on the meaning** of the phrases. **Do not** take the grammaticality into consideration.

Consider the words as they are generally used in language and use your best judgement. The relation you choose does not have to hold in every single context in which the words might be used, as long as you can imagine a reasonable context when the relationship would hold.

*** You should choose **more specific** if the first word *is a type of* the second. E.g.
a dog is more specific than/is a type of an **animal**
running is more specific than/is a type of **moving**

You should choose **more general** if the second word is a type of the first. E.g.
an animal is more general than/encompasses **a dog**
moving is more general than/encompasses **running**

The direction is important, so please be careful when choosing between these options.

*** For phrases involving groups or numbers, you should consider the phrase with the smaller number to be more specifc, unless the objects in the groups are not related. E.g.
a woman is more specific than/is a type of **a group of woman**
two boys is more general than/encompasses **a boy**

two boys is unrelated to **a dog**

*** You should choose **is the exact opposite of** if the first word means exactly not the second word. E.g.
reliable is the opposite of **unreliable**
someone is the opposite of **no one**

You should choose **is mutually exclusive with** if it is not possible for someone/something to be described by both words simultaneously. E.g.
mother is mutually exclusive with **child**
german is mutually exclusive with **french**

The difference between opposite and mutually exclusive is subtle. Please use your best judgement. If you are unsure, you should choose mutually exclusive.

A.3. Feature Templates for Lexical Entailment Classifier

All of the feature templates below assume as input a list of POS-tagged phrase pairs in the form $\langle (w_1, t_1), (w_2, t_2) \rangle$. Note that w may be a single word (*"boy"*) or a short phrase (*"little boy"*). Precise details on how the training data was obtained are given Section 3.1.3.

A.3.1. Lexical Features

We compute a variety of simple lexical features for each phrase pair, which I will refer to collectively as LEXICAL. Below, $set(w)$ refers to the set of words appearing in the phrase w, $set(t)$ refers to the set of tags appearing in the tag sequence t, and $charset(w)$ refers to the set of characters appearing in the phrase w. $len(x)$ returns the length of x in words if x is a string, and the number of elements in x if x is a set. The lexical features we extract are as follows:

- `lexical` = 1 if $len(w_1) == 1$ and $len(w_2) == 1$ else 0
- `phrase` = 1 if $len(w_1) > 1$ and $len(w_2) > 1$ else 0

- `xiny` = 1 if w_1 is a substring of w_2 else 0

- `yinx` = 1 if w_2 is a substring of w_1 else 0

- `poseq` = 1 if $t_1 == t_2$ else 0

- `courseposeq` = 1 if `lexical` == 1 and ($t_1[0] == t_2[0]$) else 0

  where $t_i[0]$ is the first character of the POS tag t_i. Since we use the Penn TreeBank tag set, this serves as proxy for a coarse-grained POS tag. Note that this feature is zero when either w_1 or w_2 is a multiword phrase.

- `words_`w_1 = 1 if $len(w_1) <= 3$ else `words_`$rarest(w_1) = 1$

  `words_`w_2 = 1 if $len(w_2) <= 3$ else `words_`$rarest(w_2) = 1$

  where $rarest(w)$ returns the lowest-frequency single word appearing in w, according to the Google 1-grams corpus (Brants and Franz (2009)). I.e. this feature is a high-dimensional sparse feature such that `words_`w is 0 for every $w \in V$ for which it is not explicitly set to 1, where V is set of all the w_i in the training data.

- `words_x_`w_1 = 1 if $len(w_1) <= 3$ else `words_`$rarest(w_1) = 1$

  `words_y_`w_2 = 1 if $len(w_2) <= 3$ else `words_`$rarest(w_2) = 1$

  I.e. this feature is like the `words_`w feature except it differentiates between words appear in w_1 and those appearing in w_2.

- `tags_x_`t_1 = 1; `tags_y_`t_2 = 1

  I.e. like the `words_`w feature, this feature is a high-dimensional sparse features such that `tags_`t is 0 for every $t \in V_t$ for which it is not explicitly set to 1, where V_t is the set of all t_i in the training data.

- `len_x` = $len(w_1)$; `len_y` = $len(w_2)$

- `posoverlap` = $len(set(t_1) \cap set(t_2))$

- `levenstein` = the edit distance between w_1 and w_2 as computed by NLTK's distance

package: `http://www.nltk.org/_modules/nltk/metrics/distance.html`

- `jaccard` $= len(charset(w_1) \cap charset(w_2))/len(charset(w_1) \cup charset(w_2))$
- `hamming` = number of positions i at which the i^{th} character in w_1 does not equal the i^{th} character in w_2. Note that the maximum value of `hamming` is the length, in characters, of the longer of w_1 and w_2. Before computing, the end of shorter string is padded with blanks until it is equal in length to the longer string.

A.3.2. WordNet Features

For each pair $\langle (w_1, t_1), (w_2, t_2) \rangle$, we include indicator features to capture the relation or relations (as defined below) to which the pair $\langle w_1, w_2 \rangle$ can be assigned according to WordNet. This feature group is referred to as WordNet.

To determine which basic relation(s) hold according to WordNet, we use the definitions of $\equiv$, $\sqsubset$, $\sqsupset$, $\dashv_{alt}$, $\dashv_{opp}$, and $\not\sim$ defined in terms of the WordNet hierarchy, as described below. If w_1 and w_2 can be assigned to the $\sim$ relation, we indicate which of the WordNet pointers (e.g. attribute, derivationally related, meronym, etc) exists between w_1 and w_2. This results in a total of 14 unique relations that a pair can be assigned via WordNet.

We include indicators for every possible relation between w_1 and w_2, even those for which the relation in WordNet is defined for a different part of speech than that indicated by t_1 and t_2. Note that WordNet only covers four coarse-grained parts of speech: noun, verb, adjective, and adverb. Specifically, for each relation r and each WordNet part of speech tag $pos \in \{n, v, a, r\}$, we include a binary feature r_{pos} which is 1 if WordNet contains any senses for w_1 and w_2 with POS *pos* such that that r holds, and 0 otherwise. We use special OOV_{pos} features to signify that either w_1 or w_2 did not appear in WordNet with the given POS tag *pos*.

In the definitions below, lowercase letters (x and y) refer to lexical expressions (i.e. natural language strings) and capital letters (X and Y) refer to the synsets containing x and y

respectively. We define the following relations.

Equivalence ($x \equiv y$)

We say $x \equiv y$ if any sense of x shares a synset with any sense of y.

$$\texttt{synsets}(x) \cap \texttt{synsets}(y) \neq \emptyset \tag{A.1}$$

Forward and Reverse Entailment ($x \sqsubset y$ and $x \sqsupset y$)

We say $x \sqsupset y$ if some synset of x is the root of a subtree containing some synset of y.

$$\texttt{synsets}(y) \cap \Big(\bigcup_{X \in \texttt{synsets}(x)} \overline{\texttt{children}(X)} \Big) \neq \emptyset \tag{A.2}$$

where $\overline{\texttt{children}(X)}$ is the transitive closure of the children of X. Forward Entailment ($x \sqsubset y$) is defined symmetrically.

Alternatives ($x \dashv_{alt} y$)

We say $x \dashv_{alt} y$ if x and y belong to disjoint subtrees rooted at a shared hypernym.

1. None of the conditions for $\equiv$, $\sqsupset$, $\sqsubset$, or $\dashv_{opp}$ are met.

2. x and y share some hypernym:

$$\texttt{hypernyms}(x) \cap \texttt{hypernyms}(y) \neq \emptyset \tag{A.3}$$

where $\texttt{hypernyms}(x)$ is the set of all hypernyms, including indirect hypernyms, of all synsets to which x belongs.

Opposites ($x \dashv_{opp} y$)

We say $x \dashv_{opp} y$ if x and y are associated by antonymy in WordNet. WordNet defines y as an antonym of x or as an antonym of any synonym of x. Note that antonym is defined as a pointer in WordNet, not as part of the hierarchical structure.

$$y \in \bigcup_{X \in \texttt{synsets}(x)} \bigcup_{w_x \in X} \texttt{WNantonyms}(w_x) \tag{A.4}$$

Otherwise Related ($x \sim y$)

We assign a pair of words x and y to the "Otherwise Related" ($\sim$) relation if x and y are in different synsets and are connected by any of the pointers in WordNet other than the hyponym/hypernym or antonym pointers, as described below.

Semantic Pointers. We consider seven pointers which WordNet models as "semantic relations": Also See, Attribute, Causes, Entailment, Holonym, Meronym, and Similar To. For semantic relations, our definition is analogous to our definitions of Forward and Reverse Entailment (i.e. those which followed WordNet's hypernym pointers). Given a pair of lexical expressions, x and y, we say that x and y are related by the semantic relation R if:

$$\texttt{synsets}(y) \cap \bigcup_{X \in \texttt{synsets}(x)} \overline{R(X)} \neq \emptyset \tag{A.5}$$

where $\overline{R(X)}$ is the transitive closure of the synsets related to X via an R pointer in WordNet.

Lexical Pointers. We consider two pointers which WordNet models as "lexical relations": Derivationally Related and Pertainym. For lexical relations, our definition is analogous to our definition of Opposites (i.e. those which followed WordNet's antonym pointers). Given a pair of lexical expressions, x and y, we say that x and y are related by the lexical relation R if any lexical item in any of the synsets of x is connected to any of the lexical items in

any of the synsets of y via an R pointer in WordNet. Specifically, we first find the set of all lexical items related to x via R as follows:

$$R(x) = \bigcup_{X \in \texttt{synsets}(x)} \bigcup_{w_x \in X} R(w_x). \tag{A.6}$$

We define all the the forms of y as all the lexical items in all of the synsets of y as follows:

$$\texttt{forms}(y) = \bigcup_{Y \in \texttt{synsets}(y)} \bigcup_{w_y \in Y} w_y. \tag{A.7}$$

We then say x and y are related by the lexical relation R if:

$$\texttt{forms}(y) \cap R(x) \neq \emptyset \tag{A.8}$$

In the case of one lexical relation, derivational relatedness, we make slight modification to enforce transitivity. I.e. x is derivationally related to y if $\texttt{forms}(y) \cap \texttt{deriv}(x) \neq \emptyset$ or $\texttt{deriv}(x) \cap \texttt{deriv}(y) \neq \emptyset$. This is to deal with a small number of cases in which two nodes share a derivational stem but lack an explicit pointer between them, e.g. WordNet relates vaccine/vaccinate and vaccination/vaccinate, but not vaccine/vaccination.

Unrelated ($x \not\sim y$)

None of the above. I.e. x and y do not belong to any of the entailment relations nor to the $\sim$ relation.

A.3.3. Distributional Features

We follow Lin and Pantel (2001) in building distributional context vectors from dependency-parsed corpora, following the intuition that good paraphrases should tend to modify and be modified by the same words. We refer to the features in this group collectively as

DISTRIBUTIONAL.

All of the DISTRIBUTIONAL features are built using the Annotated Gigaword corpus (Napoles et al. (2012)). For a single word w, we compute the dependency context vector by considering every dependency relation in which the w participates. When w is the governor of a relation r of which the word v is the dependent, we record the context as r:`gov`:v; when w is the dependent of a relation r of which v is the governor, we record the relation as r:`dep`:v. For multiword phrases $p = w_1 \dots w_k$, we consider the dependency context of p to be the combined dependency contexts of the words $w_1 \dots w_k$, excluding dependency contexts in which v is one of $w_1 \dots w_k$. As with the lexical features, we represent phrases longer than three words with their single least-frequent word, according the the Google 1-gram corpus (Brants and Franz (2009)). We ignore contexts for a word which are observed fewer than 3 times. We do not consider POS tags when building these dependency contexts. E.g. *"play"* the verb and *"play"* the noun will be represented by the same set of contexts.

Given the phrase pair $\langle (w_1, t_1), (w_2, t_2) \rangle$, let W_1 be the set of contexts of w_1 and W_2 the set of contexts of w_2. Let $|W|$ be the number of contexts in the set W. Let $W_1(c)$ be the number of times w_1 was observed in context c, and $W_2(c)$ be the number of times w_2 was observed in context c. We compute the following basic similarity features:

- `numx` $= |W_1|$; `numy` $= |W_2|$
- `diff` $= |W_1| - |W_2|$
- `intersection` $= |W_1 \cap W_2|$
- `jaccard` $= |W_1 \cap W_2| / |W_1 \cup W_2|$

We also compute various symmetric and asymmetric similarity measures defined in future work. While these metrics are original proposed separately, all of our reimplementations follow the definitions given in Kotlerman et al. (2010)'s review of distributional similarity metrics.

- `cosine_similarity`: Standard cosine similarity measure using implementation from SciKit Learn's metrics package.[13] To compute `cosine_similarity`, W_1 and W_2 are first converted into sparse real-valued vectors where the k^{th} entry of the vector for W_i is equal to $W_i(c_k)$, i.e. the number of times w_i was observed with context c_k. These vectors are l1-normalized using SciKit Learn's normalization package[14] before computing `cosine_similarity`.

- `lin_similarity`: Symmetric similarity measure proposed by Lin (1998):

$$\frac{\sum_{c \in W_1 \cap W_2} W_1(c) + W_2(c)}{\sum_{c \in W_1} W_1(c) + \sum_{c \in W_2} W_2(c)} \tag{A.9}$$

- `weeds_similarity`: Asymmetric similarity measure proposed by Weeds et al. (2004):

$$\frac{\sum_{c \in W_1 \cap W_2} W_1(c)}{\sum_{c \in W_1} W_1(c)} \tag{A.10}$$

- `clark_similarity`: Asymmetric similarity measure proposed by Clarke (2009):

$$\frac{\sum_{c \in W_1 \cap W_2} \min(W_1(c), W_2(c))}{\sum_{c \in W_1} W_1(c)} \tag{A.11}$$

- `balprec`: Asymmetric similarity measure proposed by Szpektor and Dagan (2008):

$$\sqrt{\texttt{lin\_similarity} \times \texttt{weeds\_similarity}} \tag{A.12}$$

[13] `http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html`

[14] `http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html`

A.3.4. Lexico-Syntactic Pattern Features

Hearst (1992) made the observation that certain textual patterns are strong indicators of specific semantic relationships, e.g. *X and other Y* strongly indicates hypernymy. Snow et al. (2004) used dependency parsed corpora to automatically recognize these "lexico-syntactic patterns" and use them to infer taxonomic relationships between words. Our features follow the work of Snow et al. (2004), and extend it to include all of our basic relations, rather than just hypernymy. We refer to the features in this group collectively as PATTERN.

Given a phrase pair $\langle (w_1, t_1), (w_2, t_2) \rangle$, we find all sentences in the Annotated Gigaword corpus in which the w_1 and w_2 co-occur. We then enumerate all paths through the dependency tree which connect the w_1 to w_2. We do not consider paths longer than 5 nodes. If w_1 or w_2 is a multiword phrase, we collapse the entire phrase into a single node, so that we consider all paths which originate from any word in w_1 and end at any word in w_2, subject to the constraint that none of the intermediate nodes on the path belong to w_1 or w_2.

We build a path lexicon consisting of all paths which occurred between at least 5 unique pairs in our training data. Then, the feature vector for $\langle w_1, w_2 \rangle$ is a binary vector indicating whether or not the pair was observed with each path in our path lexicon. We add three additional bits to this feature vector to indicate special cases in which:

1. w_1 was not observed anywhere in the Annotated Gigaword.

2. w_2 was not observed anywhere in Annotated Gigaword.

3. w_1 and w_2 both appear in Annotated Gigaword separately, but they never co-occur in a sentence.

A.3.5. Paraphrase Features

There are a variety of features distributed with PPDB, which we include in our classifier. These include 33 different measures used to sort the goodness of the paraphrases, including distributional similarity, bilingual alignment probabilities, and lexical similarity. A complete list is given below. These features combined are referred to as PARAPHRASE features.

- Abstract – a binary feature that indicates whether the rule is composed exclusively of nonterminal symbols.
- Adjacent – a binary feature that indicates whether rule contains adjacent nonterminal symbols.
- AGigaSim – the distributional similarity of e_1 and e_2, computed according to contexts observed in the Annotated Gigaword corpus Napoles et al. (2012).
- CharCountDiff – a feature that calculates the difference in the number of characters between the phrase and the paraphrase. This feature is used for the sentence compression experiments described in **?**.
- CharLogCR – the log-compression ratio in characters, $\log \frac{chars(f_2)}{chars(f_1)}$, another feature used in sentence compression.
- ContainsX – a binary feature that indicates whether the nonterminal symbol X is used in this rule. X is the symbol used in Hiero grammars **?**, and is sometimes used by our syntactic SCFGs when we are unable to assign a linguistically motivated nonterminal.
- GlueRule – a binary feature that indicates whether this is a glue rule. Glue rules are treated specially by the Joshua decoder **?**. They are used when the decoder cannot produce a complete parse using the other grammar rules.
- GoogleNgramSim – the distributional similarity of e_1 and e_2, computed according to

contexts observed in the Google Ngram corpus **?**.

- Identity – a binary feature that indicates whether the phrase is identical to the paraphrase.
- Lex($e_2|e_1$) – the "lexical translation" probability of the paraphrase given the original phrase. This feature is estimated as defined by Koehn et al. (2003)
- Lex($e_1|e_2$) – the lexical translation probability of phrase given the paraphrase.
- Lexical – a binary feature that says whether this is a single word paraphrase.
- LogCount – the log of the frequency estimate for this paraphrase pair.
- Monotonic – a binary feature that indicates whether multiple nonterminal symbols occur in the same order (are monotonic) or if they are re-ordered.
- PhrasePenalty – this feature is used by the decoder to count how many rules it uses in a derivation. Turning helps it to learn to prefer fewer longer phrases, or more shorter phrases. The value of this feature is always 1.
- RarityPenalty – this feature marks rules that have only been seen a handful of times. It is calculated as $\exp(1 - c(e_1, e_2))$, where $c(e_1, e_2)$ is the estimate of the frequency of this paraphrase pair.
- SourceTerminalsButNoTarget – a binary feature that fires when the phrase contains terminal symbols, but the paraphrase contains no terminal symbols.
- SourceWords – the number of words in the original phrase.
- TargetTerminalsButNoSource – a binary feature that fires when the paraphrase contains terminal symbols but the original phrase only contains nonterminal symbols.
- TargetWords – the number of words in the paraphrase.

- UnalignedSource – a binary feature that fires if there are any words in the original phrase that are not aligned to any words in the paraphrase.

- UnalignedTarget – a binary feature that fires if there are any words in the paraphrase that are not aligned to any words in the original phrase.

- WordCountDiff – the difference in the number of words in the original phrase and the paraphrase. This feature is used for our sentence compression experiments.

- WordLenDiff – the difference in average word length between the original phrase and the paraphrase. This feature is useful for text compression and simplification experiments.

- WordLogCR – the log-compression ratio in words, estimated as log words(e) words(f). This feature is used for our sentence compression experiments.

- $p(LHS|e_2)$ – the (negative log) probability of the lefthand side nonterminal symbol given the paraphrase.

- $p(LHS|e_1)$ – the (negative log) probability of the lefthand side nonterminal symbol given the original phrase.

- $p(e_2|LHS)$ – the (negative log) probability of the paraphrase given the lefthand side nonterminal symbol (this is typically a very low probability).

- $p(e_2|e_1)$ – the paraphrase probability of the paraphrase given the original phrase, as defined by Bannard and Callison-Burch (2005). This is given as a negative log value.

- $p(e_2|e_1, LHS)$ – the (negative log) probability of paraphrase given the the lefthand side nonterminal symbol and the original phrase.

- $p(e_1|LHS)$ – the (negative log) probability of original phrase given the the lefthand side nonterminal (this is typically a very low probability).

- $p(e_1|e_2)$ – the paraphrase probability of the original phrase given the paraphrase, as defined by Bannard and Callison-Burch (2005). This is given as a negative log value.

- $p(e_1|e_2, LHS)$ – the (negative log) probability of original phrase given the the lefthand side nonterminal symbol and the paraphrase.

A.3.6. Translation Features

PPDB is based on the "bilingual pivoting"' method, in which two phrases are considered paraphrases if they share a foreign translation. The English PPDB was built by pivoting through 24 foreign languages. We use the pivot words from all of these languages to derive a set of features, which we refer to as TRANSLATION features. Let $\tau_l(w)$ be the set of observed translations of the phrase w in language l. Let $\tau_*(w) = \bigcup_l \tau_l(w)$, the pooled set of observed translations of the phrase w across all languages. Given a pair of phrases $\langle (w_1, t_1), (w_2, t_2) \rangle$, for each language l, we compute the following features:

- $\texttt{trans}_l\texttt{\_x}$: The the number of shared translations as a fraction of the total translations of w_1 for the language l:

$$\frac{\mid \tau_l(w_1) \cap \tau_l(w_2) \mid}{\mid \tau_l(w_1) \mid} \tag{A.13}$$

- $\texttt{trans}_l\texttt{\_y}$: The the number of shared translations as a fraction of the total translations of w_2 for the language l:

$$\frac{\mid \tau_l(w_1) \cap \tau_l(w_2) \mid}{\mid \tau_l(w_2) \mid} \tag{A.14}$$

- $\texttt{xmin} = \min_l \{\texttt{trans}_l\texttt{\_x}\}$; $\texttt{ymin} = \min_l \{\texttt{trans}_l\texttt{\_y}\}$

- $\texttt{xmax} = \max_l \{\texttt{trans}_l\texttt{\_x}\}$; $\texttt{ymax} = \max_l \{\texttt{trans}_l\texttt{\_y}\}$

- $\texttt{xmean} = \frac{1}{\#\text{ languages}} \sum_l \texttt{trans}_l\texttt{\_x}$; $\texttt{ymean} = \frac{1}{\#\text{ languages}} \sum_l \texttt{trans}_l\texttt{\_y}$

- $\texttt{trans}_*\texttt{\_x}$: The the number of shared translations as a fraction of the total translations

of w_1 across all languages:

$$\frac{\mid \tau_*(w_1) \cap \tau_*(w_2) \mid}{\mid \tau_*(w_1) \mid} \tag{A.15}$$

- `trans`$_*$`_y`: The the number of shared translations as a fraction of the total translations of w_2 across all languages:

$$\frac{\mid \tau_*(w_1) \cap \tau_*(w_2) \mid}{\mid \tau_*(w_2) \mid} \tag{A.16}$$

A.4. Selecting MH Pairs and Contexts for Simplified RTE Annotation

In this appendix, we describe our process for constructing p/h pairs as required by simplified RTE task described in Section 4.1.2. Specifically, we need to 1) choose a sample of MH phrases to study, and 2) identify good contexts in which to instantiate[15] each MH. The point of the experiments here is to motivate our decisions for how to perform these two steps, before proceeding to collect a large number of annotations on which to perform our analysis. For all of the experiments in this section, we use the Annotated Gigaword corpus Napoles et al. (2012).

A.4.1. Choosing MHs

Sampling

We are interested in how the denotation of a noun is affected when it undergoes modification. We therefore take a noun-centric approach to sampling. That is, we want to first choose at a set of relevant nouns and second choose a set of relevant modifiers for those nouns. The primary concern in sampling is that the frequency with which an MH pair occurs will effect the tendency of MH to entail H. To test whether the frequency of occurrence of MH has any effect on perceived entailment, we take a stratified sample of MHs.

[15] We will refer to the act of displaying an MH phrase in the context of a p/h pair like this as "instantiating" the MH. For example, the p/h pair *"A dog is sleeping"/"A brown dog is sleeping"* instantiates the MH *"brown dog'.*

First, we count the number of occurrences of each MH in the corpus. We only consider MH instances in which the noun is immediately preceded by the adjective. We filter out cases in which, according to the dependency parse, the adjective is a not direct modifier of the noun or the noun is itself a modifier of another noun. Additionally, we remove sentences which do not begin with capital letters and end with periods. This heuristic is meant to exclude bad parses that split sentences incorrectly.

We define 2 parameters: `num_bins`, the number of bins to stratify over, and `K`, the number of instances to sample from each bin. We set `num_bins` = 4, and `K` = 5. We measure the "frequency" of a noun by the number of unique adjectives with which is was observed. To create our sample, we first sort all of the nouns in our list by frequency. We divide this list into `num_bins` bins, in which the first bin consists of the `K` most frequent nouns, and the remainder of the list is divided into `num_bins`-1 equally-sized bins. We then select `K` nouns from each bin. These `K` nouns constitute our `noun_list`. For each noun H_i in `noun_list`, we collect all MH_i phrases, and sort these phrases by frequency. Again, we divide each of these lists into bins such that the first bin consists of the `K` most frequent MH_is, and the remaining pairs are divided into `n_bins` equally sized bins. Finally, we sample `K` MH_is from each of these bins.

This procedure results in `n_bins`×`n_bins`+1 bins of MHs, and we sample `K` MHs from each. Our final sample as a total of (`n_bins` × `K` nouns) × (`n_bins`+1 × `K` MH_is per H_i) = (4×5)×(5×5) = 500 MHs in our sample.

Preprocessing

Before running the above sampling algorithm, we perform a preprocessing step on the full list of MHs, to ensure that our sampled nouns are frequent enough in the data to meet the constraints of our experimental design. Specifically, we only include MHs that meet the following conditions: 1) the noun H occurs with at least `K`×`n_bins` unique adjectives M, 2) the MH appears at least 3 times in the corpus, and 3) the modifier M occurs with at least

five unique nouns H. Condition (1) ensures that each chosen H_i is frequent enough that we can take a sample of `K` from each of `n_bins` bins of MH_is. Condition (2) ensures that each of the chosen MHs can be displayed in 3 unique instantiating contexts during the annotation. Condition (3) simply ensures that the chosen modifiers are relevant modifiers to study, as opposed to obscure or incorrectly-tagged adjectives. After applying these criteria, we have a list of 1,007,939 candidate MHs from which to sample.

A.4.2. Choosing Contexts

Native vs. Artificial Contexts

We want to select 3 contexts in which to instantiate each MH in order to collect entailment judgements. As described in Section 4.1, we want to collect judgements on the entailment implication modifying a noun H with a modifier M. There are two strategies for creating such contexts:

1. Chose sentences from the corpus in which the MH appears. Delete M from this context to create p, and leave the sentence as-is to create h. We refer to this strategy as using "native" contexts.

2. Chose sentences from the corpus in which the noun H appears unmodified. Leave the sentence as-is to create the p and insert the modifier M into the sentence in order to create the h. We refer to this strategy as using "artificial" contexts.

We would like to test the effect of using native vs. artificial contexts on the entailment judgements we receive from humans. Therefore, for each of the 500 MHs chosen by the method described in Section , we attempt to construct 3 p/h pairs using native contexts, and 3 p/h pairs using artificial contexts. This will allow us to make a side-by-side comparison of the two methods for creating contexts.

Sampling

Before sampling sentences to use as contexts, we omit sentences containing obvious negations (i.e. *no*, *not*, *n't*), since negations invert the true entailment associated with an atomic edit. To simplify the annotation, we try to prefer short sentences. Specifically, we first try to sample sentences that are less than 15 words long. If there are not enough sentences to fill our sample, we try to sample from sentences less than 20 words long. We continue raising the upper limit until the sample has been filled.

We are able to select exactly 3 foreign contexts for each of our 500 MHs, for a total of 1,500 foreign-contexts. After eliminating negations, we cannot select a full 3 native contexts for every MH in our sample, since many of the MHs from the infrequent bins only occur a handful of times. As a result, we have a total of 1,355 native contexts covering the 500 MHs in our sample. Specifically, 396 pairs have 3 contexts, 70 have 2, and 27 have 1 context, and 7 do not appear in any acceptable contexts, and so are removed from our sample.

A.4.3. Annotation

We collect entailment judgements on Amazon Mechanical Turk. We present each worker with the premise p (which contains only the noun) followed by the hypothesis h (which contains the noun modified by the adjective) and ask them to determine, on a likest-style scale from 1 to 5, how likely it is that h is true given that p is true, or that h and p describe the same scenario. We provide several examples of p/h pairs and the expected annotations. Our exact guidelines and examples are shown in Appendix A.5.

We post two batches of HITs: one consisting of the native contexts, and one consisting of the foreign contexts. Note that the foreign batch was posted after the native batch, and therefore had slightly improved instructions and QC questions. This does not appear to have a measurable effect on the workers' accuracy (Section A.4.3) or level of agreement (Section A.4.3), although it did result in happier workers.

Quality Control

We embed quality control question as described in Section 4.2.2. For this initial pilot study, we made decisions on which workers to accept and which to reject by manually inspecting workers who fell below 50% accuracy on our controls and determining whether their mistakes seemed like legitimate differences in opinion or like spam work. We settle on more principled quality control criteria for our experiments in Section 4.2.2.

Overall, 173 workers participated in this set of tasks: 121 in the native tasks and 66 in the foreign tasks. 14 workers participated in both batches. On average, workers achieved 77% accuracy on our controls. 68% of our annotations come from workers with greater than 80% accuracy. We rejected only 3 workers, contributing a total of 61 assignments. The rejected workers' annotations are not included in our analysis, and their assignments were reposted to be completed by another worker.

Inter-Annotator Agreement

We measure inter-annotator agreement using the quadratic-weighted κ, which is a variant of Fleiss's κ that accounts for ordinal annotations by punishing a large disagreement (e.g. 5 vs. 2) more than a small one (e.g. 5 vs 4). Our κ computations reflect the agreement between one annotator ("Annotator 1") and the mean of the other two annotators, rounded to the nearest integer ("Annotator 2"). We compute κ two ways: over the 5-way classification originally used by the workers, and over a 3-way classification that results from collapsing the 5-point range in the following way such that a score < 2.5 is considered CONTRADICTION, a score between 2.5 and 3.5 is considered UNKNOWN, and a score > 3.5 is considered ENTAILMENT. Note that fractional scores can result when we take the average of annotators' integer scores.

We compute each κ metric 1,000 times, each time randomly choosing which annotator to use as Annotator 1, and report the mean and 95% confidence interval (Table 59). We also

report the percentage of p/h pairs on which all three annotators agreed on the score (from 1 to 5), and the percentage for which two out of three agreed.

	3-way κ	5-way κ	3 out of 3	2 out of 3
Native	0.29±0.3	0.35±0.3	23%	61%
Artificial	0.35±0.3	0.32±0.3	25%	57%

Table 59: Inter-annotator agreement in native and in artificial contexts.

A.4.4. Analysis

Preprocessing

Before performing any calculations or analysis on our data, we remove sentences for which at least one worker gave the response "the sentence does not make sense". This results in removing 9% of sentences (3% of adjective/noun pairs) from the native setting and 17% of sentences (4% of adjective/noun pairs) from the foreign setting. The number of instances removed is roughly uniform across frequency bins (described in Section A.4.2). In the end, we have 44–66 sentences and 22-25 adjective/noun pairs per bin in the native setting, and 42–65 sentences and 21-25 adjective/noun pairs per bin in the foreign setting. Before removing any instances, ideally, we would have had 75 sentences and 25 adjective/noun pairs per bin in each setting (see Section A.4.2 for other reasons why the number of sentences per bin might fall below 75). On inspection, many of the nonsense sentences are due to part-of-speech tagging errors.

Comparing Native and Artificial Contexts

The first question we ask is: are contexts in which an MH naturally occurs more likely to produce forward-entailing judgements (e.g. $H \sqsubset MH$) than contexts in which the noun H appears unmodified? Roughly, we can use the entailment judgements from the artificial contexts as a means for assessing the extent to which the noun itself, on average, entails the modifier, and e can use the entailment judgements from native contexts as a means for

assessing the degree to which sentential context, on average, entails the modifier.

Figure 23 shows how the entailment properties associated with inserting adjectives are distributed in the native and the artificial setting. We can see that the distribution of judgements taken from the foreign setting falls to the left of that taken from the native setting. That is, in general, when a human writer includes a modifier (i.e. the native setting), that modifier is does not actually contribute new information, but rather is sufficiently entailed by the surrounding context such that, had it not been included, readers could nonetheless have inferred that the modifier applies. In contrast, when humans do not include an adjective (the foreign setting), it is often *not* the case that that adjective is implied or can be inserted without changing the meaning of the sentence.

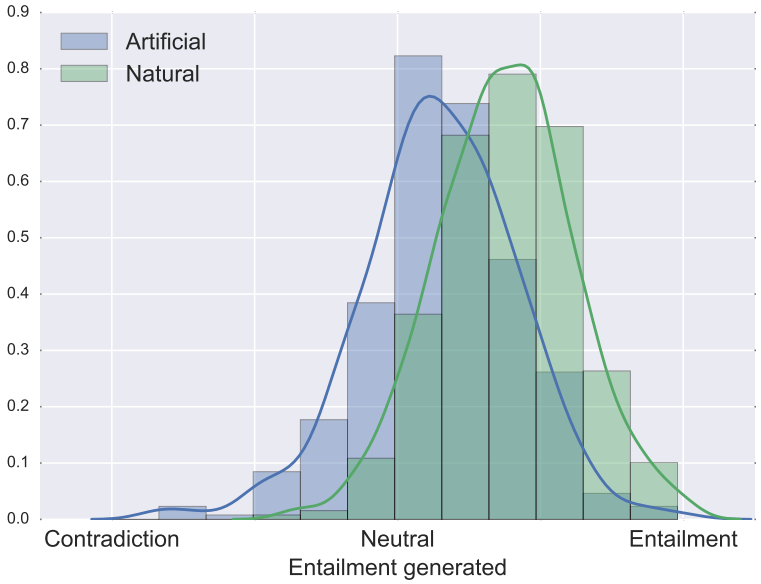


Figure 23: Entailment judgements for whether $H \sqsubset MH$ in native and in artificial contexts.

Effect of MH Frequency on Entailment Judgements

The second question we are interested in answering is whether the frequency with which an MH occurs in language is related to the tendency of the noun to entail the modified noun. In other words, if a noun H is frequently/infrequently modified by a specific adjective

M, is it more/less likely that humans will tend to infer $H \sqsubset MH$? Figure 24 shows the relationship between the log frequency of the MH in the corpus, and the mean entailment score (on our 5 point scale). We can see that while there does appear to to be a slight negative correlation– i.e. the more frequently MH appears, the less likely it is that M is entailed by H– this difference does not appear to be significant at ($p < 0.05$).

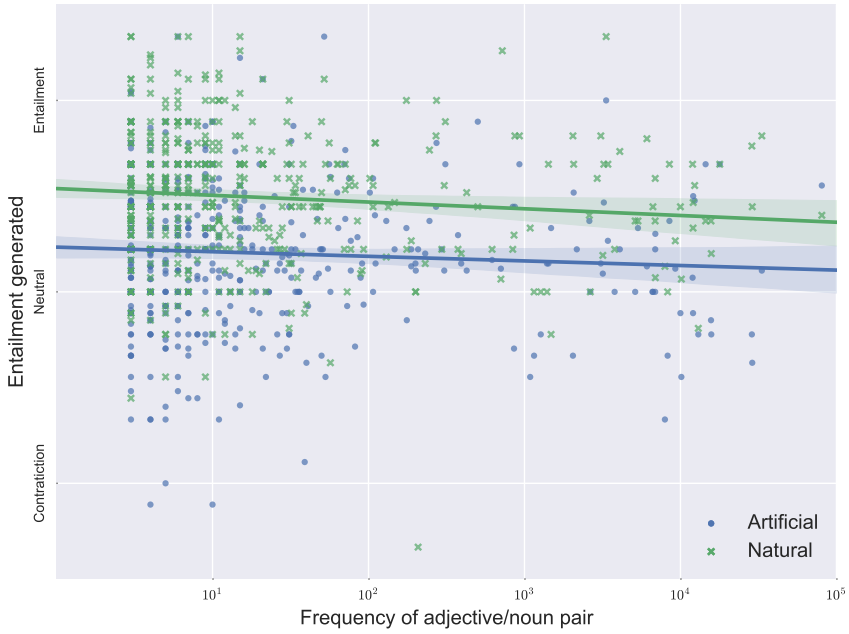


Figure 24: Relationship between human judgements of whether $H \sqsubset MH$ and frequency of MH in corpus.

A.5. Instructions for MH Composition HIT

A.5.1. Instructions

For each pair of sentences, assume that the first sentence is true, or describes a real scenerio. Using your best judgement, indicate how likely it is, on a scale of 1 to 5, that the sentence in the shaded box is also true, or describes the same scenerio. If either of the sentences do not make sense, indicate so by selecting "The sentence does not make sense."

Your answers should be based only on information which is stated or implied by the first

sentence. For many sentences, there is not a clear correct answer, and several of the sentences are fragments that may be hard to evaluate in terms of clear true/false. In these cases, we ask you to rely on your common sense and knowledge of how people typically use language in order to provide your best answer. When the sentence is truly too fragmented to be interpreted, you can choose "The sentence does not make sense." Keep in mind, **we are predominantly interested in understanding whether the second sentence communicates the same information as the first, or if it adds or removes important information**.

See the below examples for guidance.

Example 1

A dog is playing in the dirt.

A black dog is playing in the dirt.

You should answer 3 (neither true nor false) for this example, since we cannot infer the color of the dog from the first sentence.

Example 2

A man is in a car talking on a phone.

A man is in a car talking on a cell phone.

You should answer 4 (probably true) for this example, since it is reasonable to assume that if the man is in his car, he is most likely on a cell phone as opposed to a landline phone.

Example 3

A dog is playing in the snow.

A dog is playing in the white snow.

You should answer 4 (probably true) for this example, since it is reasonable to assume that the snow is white. 5 (definitely true) would also be acceptable here, but 4 would be a better choice, since it is possible for the snow not to be white.

Example 4

The policies are aimed at eliminating tax loopholes.

The tax policies are aimed at eliminating tax loopholes.

You should answer 4 (probably true) or 5 (definitely true) for this example, since it is reasonable to assume that the policies are tax policies, given that then are aimed at eliminating tax loopholes.

Example 5

A boy is holding a gun.

A boy is holding a fake gun.

You should answer 1 (false) or 2 (probably false) for this example, since the word "fake" in the sentence implies that the boy is not actually holding a gun.

Example 6

Barack Obama is the president of the United States.

Barack Obama is the former president of the United States.

You should answer 1 (false) or 2 (probably false) for this example, since the word "former" in the sentence implies that Barack Obama is no longer the president.

Example 7

My favorite move is the Butterfly Effect.

My favorite move is the Butterfly large effect.

You should answer the "sentence does not make sense" for this example, since it does not make sense to break up the phrase "Butterfly Effect" which is the name of a movie. Many of these sentences are generated automatically, and so there will be mistakes that may prevent you from providing a meaningful answer. Use your best judgement in determining whether or not an error prevents you from providing an answer. Some errors (e.g. poor grammar or missing spaces) can be overlooked and you should still attempt to answer the question.

For a small number of cases, there is only one correct answer. These cases will be quality

controlled, and we will not approve your work if you fail to answer these quality control question correctly. We try very hard to ensure that the quality control questions are unambiguous. If you read the examples carefully, you should not have trouble answering the quality control questions correctly.

Please read the sentences and think about each one carefully before making a selection. We rely on your careful judgements in our research, and very much appreciate your time and effort.

A.5.2. Quality Control Questions

Contradiction (1 or 2)

He radioed back several times but failed to get/got a response.
I refuse to conclude/conclude from all this that I have been unknowingly married to a rock star for nigh on 18 years.
Ellis refused to say/said Monday whether he had asked Bradley to call Shapiro.
A fourth sale was scheduled to be advertised but the newspaper failed to print/printed it.
Nynex, citing the pending litigation, refused to allow/allowed Burke to comment.
This year, Denver went 40-42 and failed to make/made the playoffs.
He failed to win/won over the Republicans he courted for his economic plan.
When Clinton refused to intercede,/interceded, Weirton voters took out their anger on Gore.
Pro-democracy legislators failed to postpone/postponsed deliberation of a the court bill until after legislative council elections in September.
Failing to reach/reaching top five was her only regret in Beijng.
A boy is holding a fake gun/gun.
A boy is holding a fake snake/snake.
A boy is pretending to bake/baking a cake.
A boy is pretending to paint/painting a house.

A girl is playing with a fake sowrd.

A little girl is playing with a fake spider/spider.

A girl is pretending to grill/grilling a burger.

A girl is pretending to chop/chopping down a tree.

Unknown (3)

Child, man, and asian woman/woman walking near the water.

A small baby in red winter clothing/winter clothing is being held upright to take a picture.

Three people are pulling a red rope/rope on a hillside.

Bright blue lake with an inflatable boat/a boat in the distance.

A brown dog wearing a black collar/collar is running through some water.

Woman jogging beside the muddy road/road.

The woman is dancing on the shore of a foggy lake/lake at dusk.

A man sits in his yellow tent/tent on a mountain.

Two young boys hold an asian man/man's hands.

A black dog retrieves a gray bird/bird.

Child, man, and blond woman/woman walking near the water.

A women sitting on a blue sofa/sofa while a small Jack Russell walks towards the camera.

A man with multi colored curly hair/hair wearing a weird outfit.

Close up of a gray sheep/sheep.

The man is riding a black motorcycle/motorcycle down the road.

Three people are pulling a white rope/rope on a hillside.

Crowd waiting on Main Street subway platform for a black train/train.

Man with children near a red bench/bench in a suburban area.

A girl is outside a gray house/house with a pink hula hoop spinning around her upper leg.

Two girls swing with a blond boy/boy, all three are wearing blue shirts of the same shade and blue jeans.

A small child wearing a brown hat/hat is playing on rocks at the edge of a body of water.

A runner on a red track/track.

Two dogs sit on a brown couch/couch with many stuffed animals.

A small child kissing a siamese cat/cat on the kitchen counter.

A black dog retrieves a white bird/bird.

A young girl wearing a green dress/dress and sandals runs in the grass.

A baby swings in a red swing/swing near a wooden fence.

A woman carries a sleeping new baby/new baby on her back.

A girl is climbing a brown rock/rock while someone is filming her.

A rollerblader skating inside a red tube/tube.

A man is riding his small bicycle/bicycle down the sidewalk.

A white dog with a green sweater/sweater on and a black and brown dog touching noses.

Child, man, and young woman/woman walking near the water.

An airliner on the snowy ground/ground is being loaded for flight.

Two skiing people jumping a yellow ramp/ramp and a man looking straight ahead.

A brown dog wearing a pink collar/collar is running through some water.

A woman assists a marathon runner by holding a gray umbrella/umbrella over her and giving her water.

A middle eastern couple sitting on a pink couch/couch holding their baby and displaying a gift.

Empty barge floating down a small river/river.

A girl is outside a blue house/house with a pink hula hoop spinning around her upper leg.

Entailment (4 or 5)

High school dropouts had the most diverse group of discussion-mates, while college graduates managed to shelter/sheltered themselves from uncomfortable perspectives.

Six regions managed to make/made their ends meet by securing food supplies from elsewhere.

Gennady Zyuganov was the leader of the Communist Party which managed to keep/kept

its power.

Clinton, by contrast, always manages to alter/alters his stump speech to appeal to the audience at hand.

Police say about 25 passengers managed to escape/escaped.

One managed to get/got into a polling center, killing two people.

He managed to win/won one service game but the set was long lost.

A.6. Feature Templates for Weakly-Supervised Reranking Model

Given an instance e and a class label MH, let

$$\texttt{props} = [w_0 \times \mathcal{D}(\langle e, p_0, o_0 \rangle) \ldots w_k \times \mathcal{D}(\langle e, p_k, o_k \rangle)]$$

be the list of count $\times$ weight scores for all of the properties in $I(MH)$ (i.e. the list which is summed over in Equation 5.8), sorted in decreasing order. We then extract the following features:

- For K in {1, 10, 100, 1000, len(props)}
  - sum(`props`[:K])
  - arithmetic_mean(`props`[:K])
  - geometric_mean(`props`[:K])
- `headconf`: Confidence score for $\langle e, MH \rangle$ according to $\mathcal{O}$
- `catcount`: Total number of categories in $\mathcal{O}$ of which e is an instance
- `factcount`: Total number of tuples in $\mathcal{D}$ in which e is the subject
- sum(`props`) / `catcount`

- arithmetic_mean(`props`) / `catcount`
- geometric_mean(`props`)/`catcount`
- sum(`props`) / `factcount`
- arithmetic_mean(`props`) / `factcount`
- geometric_mean(`props`)/`factcount`

All of the features are binarized. We use the log of the value, rounded to the nearest integer, in order to assign values to a bin for all features except for `headconf`, for which the bin is simply the value rounded to two decimal places. For features parameterized by K, the features are only defined for length(`props`)≥K. Otherwise, an indicator feature is set to designate that the length of the list was less than K.

BIBLIOGRAPHY

N. Abdullah and R. A. Frost. Adjectives: A uniform semantic approach. In *Conference of the Canadian Society for Computational Studies of Intelligence*, 2005.

E. Akhmatova. Textual entailment resolution via atomic propositions. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 150. Citeseer, 2005.

M. Amoia and C. Gardent. Adjective based inference. In *Proceedings of the Workshop KRAQ'06 on Knowledge and Reasoning for Language Processing*, pages 20–27. Association for Computational Linguistics, 2006.

M. Amoia and C. Gardent. A first order semantic approach to adjectival inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 185–192, Prague, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W07/W07-1430`.

J. Andreas and D. Klein. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1125`.

G. Angeli and C. D. Manning. NaturalLI: Natural logic inference for common sense reasoning. In *Empirical Methods in Natural Language Processing (EMNLP)*, October 2014. URL `http://nlp.stanford.edu/pubs/angeli2014-emnlp-naturalli.pdf`.

G. Angeli, M. J. Johnson Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1034`.

K. Bach. Context dependence (such as it is). *The Continuum Companion to the Philosophy of Language*, 2012.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

C. Bannard and C. Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604, 2005.

R. Bar Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The second pascal recognising textual entailment challenge. 2006.

R. Bar-Haim, I. Dagan, I. Greental, and E. Shnarch. Semantic inference at the lexical-syntactic level. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 871. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

M. Baroni and R. Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, October 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D10-1115`.

S. Bayer, J. Burger, L. Ferro, J. Henderson, and A. Yeh. Mitres submissions to the eu pascal rte challenge. In *Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment*. Citeseer, 2005.

L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The sixth pascal recognizing textual entailment challenge. In *TAC*, 2010.

L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The seventh pascal recognizing textual entailment challenge. In *TAC*, 2011.

J. Berant, I. Dagan, and J. Goldberger. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 610–619, 2011. ISBN 978-1-932432-87-9. URL `http://dl.acm.org/citation.cfm?id=2002472.2002550`.

R. Bhagat, P. Pantel, E. H. Hovy, and M. Rey. Ledir: An unsupervised algorithm for learning directionality of inference rules. In *EMNLP-CoNLL*, pages 161–170. Citeseer, 2007.

J. Bjerva, J. Bos, R. van der Goot, and M. Nissim. The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. *SemEval 2014*, page 642, 2014.

G. Boleda, E. M. Vecchi, M. Cornudella, and L. McNally. First order vs. higher order modification in distributional semantics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1223–1233, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D12-1112`.

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

J. Bos and K. Markert. Recognising textual entailment with robust logical inference. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 404–426. Springer, 2006.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1075`.

T. Brants and A. Franz. Web 1t 5-gram, 10 european languages version 1. *Linguistic Data Consortium, Philadelphia*, 2009.

C. J. Brockett, S. Kok, and D. Zhou. Locating paraphrases through utilization of a multipartite graph, July 9 2013. US Patent 8,484,016.

J. Brooke, A. Hammond, and G. Hirst. GutenTag: an NLP-driven tool for digital humanities research in the Project Gutenberg corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 42–47, Denver, Colorado, USA, June 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W15-0705`.

N. Chambers, D. Cer, T. Grenager, D. Hall, C. Kiddon, B. MacCartney, M.-C. de Marneffe, D. Ramage, E. Yeh, and C. D. Manning. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170. Association for Computational Linguistics, 2007.

T. Chklovski and P. Pantel. VerbOcean: Mining the web for fine-grained semantic verb relations. In *EMNLP*, volume 2004, pages 33–40, 2004.

K. Claessen and N. Sorensson. New techniques that improve mace-style model finding. In *Proc. of Workshop on Model Computation (MODEL)*, 2003.

D. Clarke. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119, 2009.

R. Cooper, D. Crouch, J. Van Eijck, C. Fox, J. Van Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, et al. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium, 1996.

I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190. Springer, 2006.

D. Das and N. A. Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and*

the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1, pages 468–476. Association for Computational Linguistics, 2009.

G. Del Pinal. Dual content semantics, privative adjectives and dynamic compositionality. *Semantics and Pragmatics*, 5, 2015.

A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, pages 1535–1545, Edinburgh, Scotland, 2011.

C. Fellbaum. *WordNet*. Wiley Online Library, 1998.

J. L. Fleiss, B. Levin, and M. C. Paik. *Statistical methods for rates and proportions*. John Wiley & Sons, 2013.

J. Fodor. There are no recognitional concepts; not even red. *Philosophical issues*, pages 1–14, 1998.

A. Fowler, B. Hauser, D. Hodges, I. Niles, A. Novischi, and J. Stephan. Applying cogex to recognize textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 69–72. Citeseer, 2005.

J. Ganitkevitch and C. Callison-Burch. The multilingual paraphrase database. In *The 9th edition of the Language Resources and Evaluation Conference*, Reykjavik, Iceland, May 2014. European Language Resources Association. URL `http://cis.upenn.edu/~ccb/publications/ppdb-multilingual.pdf`.

J. Ganitkevitch, B. Van Durme, and C. Callison-Burch. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `http://cs.jhu.edu/~ccb/publications/ppdb.pdf`.

D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W07/W07-1401`.

N. D. Goodman and M. C. Frank. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11):818–829, 2016.

A. D. Haghighi, A. Y. Ng, and C. D. Manning. Robust textual inference via graph matching. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 387–394. Association for Computational Linguistics, 2005.

S. Harabagiu and A. Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 905–912. Association for Computational Linguistics, 2006.

S. M. Harabagiu, M. A. Paşca, and S. J. Maiorano. Experiments with open-domain textual question answering. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 292–298. Association for Computational Linguistics, 2000.

C. Hashimoto, K. Torisawa, K. Kuroda, S. De Saeger, M. Murata, and J. Kazama. Large-scale verb entailment acquisition from the web. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1172–1181. Association for Computational Linguistics, 2009.

M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING '92, pages 539–545, 1992. doi: 10.3115/992133.992154. URL `http://dx.doi.org/10.3115/992133.992154`.

M. Heilman and N. A. Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics, 2010.

I. Heim and A. Kratzer. *Semantics in generative grammar*, volume 13. Blackwell Oxford, 1998.

T. M. Janssen. Montague semantics. *The Stanford Encyclopedia of Philosophy, winter*, 2012.

N. Kadmon. *Formal Pragmatics: Semantics, Pragmatics, Presupposition, and Focus. Wiley*. Blackwell. Oxford, 2001.

H. Kamp and B. Partee. Prototype theory and compositionality. *Cognition*, 57(2):129–191, 1995.

L. Karttunen. Presupposition: What went wrong? In *Semantics and Linguistic Theory*, volume 26, pages 705–731, 2016.

P. Kingsbury and M. Palmer. From treebank to propbank. In *LREC*, pages 1989–1993. Citeseer, 2002.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54, 2003.

L. Kotlerman, I. Dagan, I. Szpektor, and M. Zhitomirsky-Geffet. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389, 2010.

M. Kouylekov and B. Magnini. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the First Challenge Workshop Recognising Textual Entailment*, pages 17–20, 2005.

R. Lahav. Against compositionality: the case of adjectives. *Philosophical studies*, 57(3): 261–279, 1989.

G. Lakoff. *Linguistics and natural logic*. Springer, 1972.

N. Lazic, A. Subramanya, M. Ringgaard, and F. Pereira. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, 3: 503–515, 2015. ISSN 2307-387X.

M. Lewis and M. Steedman. Combining distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192, 2013.

D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774, 1998.

D. Lin and P. Pantel. DIRT – Discovery of Inference Rules from Text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM, 2001.

D. Lin and X. Wu. Phrase clustering for discriminative learning. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP-09)*, pages 1030–1038, Singapore, 2009.

E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics, 2002.

B. MacCartney. *Natural language inference*. PhD thesis, Citeseer, 2009.

B. MacCartney and C. D. Manning. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 521–528. Association for Computational Linguistics, 2008.

B. Magnini, R. Zanoli, I. Dagan, K. Eichler, G. Neumann, T.-G. Noh, S. Padó, A. Stern, and O. Levy. The Excitement Open Platform for textual inferences. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 43–48, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-5008`.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL `http://www.aclweb.org/anthology/P/P14/P14-5010`.

M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli. Semeval-

2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *SemEval-2014*, 2014a.

M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May 2014b. ISBN 978-2-9517408-8-4. URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf`. ACL Anthology Identifier: L14-1314.

Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-12)*, pages 523–534, Jeju Island, Korea, 2012.

J. P. McCrae, F. Quattri, C. Unger, and P. Cimiano. Modelling the semantics of adjectives in the ontology-lexicon interface. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)*, pages 198–209, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL `http://www.aclweb.org/anthology/W14-4724`.

L. McNally and G. Boleda. Relational adjectives as properties of kinds. *Empirical issues in formal syntax and semantics*, 8:179–196, 2004.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

K. Mineshima, P. Martínez-Gómez, Y. Miyao, and D. Bekki. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1244`.

W. Monroe, R. X. D. Hawkins, N. D. Goodman, and C. Potts. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 2017.

L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://anthology.aclweb.org/P16-2022`.

P. Nakov and M. Hearst. Semantic interpretation of noun compounds using verbal and other paraphrases. *ACM Transactions on Speech and Language Processing (TSLP)*, 10 (3):13, 2013.

C. Napoles, M. Gormley, and B. Van Durme. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100, 2012.

N. Nayak, M. Kowarsky, G. Angeli, and C. D. Manning. A dictionary of nonsubsective adjectives. Technical Report CSTR 2014-04, Department of Computer Science, Stanford University, October 2014.

P. Pantel, E. Crestan, A. Borkovsky, A. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 938–947, Singapore, 2009.

A. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1244`.

B. Partee. Compositionality and coercion in semantics: The dynamics of adjective meaning. *Cognitive foundations of interpretation*, pages 145–161, 2007.

B. H. Partee. Are there privative adjectives. In *Conference on the Philosophy of Terry Parsons, University of Massachusetts, Amherst*, 2003.

M. Pasca. Interpreting compound noun phrases using web search queries. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 335–344, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

E. Pavlick and C. Callison-Burch. Most babies are little and most problems are huge: Compositional entailment in adjective nouns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany, 2016a. Association for Computational Linguistics.

E. Pavlick and C. Callison-Burch. So-called non-subsective adjectives. August 2016b.

E. Pavlick and M. Pasca. Identifying 1950s american jazz musicians: Fine-grained isa extraction via modifier composition. August 2017.

E. Pavlick, J. Bos, M. Nissim, C. Beller, B. V. Durme, and C. Callison-Burch. Adding semantics to data-driven paraphrasing. In *Association for Computational Linguistics*, Beijing, China, July 2015a. Association for Computational Linguistics.

E. Pavlick, P. Rastogi, J. Ganitkevitch, B. Van Durme, and C. Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Pro-*

cessing (Volume 2: Short Papers), pages 425–430, Beijing, China, July 2015b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-2070.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

S. Petrov, P. Chang, M. Ringgaard, and H. Alshawi. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP-10)*, pages 705–713, Cambridge, Massachusetts, 2010.

J. Pustejovsky. Inference patterns with intensional adjectives. In *Proceedings of the 9th Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation*, pages 85–89, Potsdam, Germany, March 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-0509.

R. Raina, A. Y. Ng, and C. D. Manning. Robust textual inference via learning and abductive reasoning. In *AAAI*, pages 1099–1105, 2005.

M. Reimer. Do adjectives conform to compositionality? *Nous*, 36(s 16):183–198, 2002.

A. Riazanov and A. Voronkov. The design and implementation of vampire. *AI communications*, 15(2, 3):91–110, 2002.

T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kocisky, and P. Blunsom. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*, 2016.

D. Rothschild and G. Segal. Indexical predicates. *Mind & language*, 24(4):467–493, 2009.

V. M. Sánchez Valencia. *Studies on natural logic and categorial grammar*. VM Sanchez Valencia, 1991.

L. K. Schubert, B. D. Van Durme, and M. Bazrafshan. Entailment inference in a natural logic-like general reasoner. In *AAAI Fall Symposium: Commonsense Knowledge*, 2010.

K. K. Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon. 2005.

V. Shwartz, Y. Goldberg, and I. Dagan. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1226.

R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*, volume 17, pages 1297–1304, 2004.

R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational*

Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL-44, pages 801–808, 2006. doi: 10.3115/1220175.1220276. URL http://dx.doi.org/10.3115/1220175.1220276.

R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

A. Stern and I. Dagan. BIUTEE: A modular open-source system for recognizing textual entailment. In *Proceedings of the ACL 2012 System Demonstrations*, pages 73–78, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P12-3013.

I. Szpektor and I. Dagan. Learning entailment rules for unary templates. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 849–856, 2008. ISBN 978-1-905593-44-6. URL http://dl.acm.org/citation.cfm?id=1599081.1599188.

I. Szpektor, H. Tanev, D. Dagan, B. Coppola, et al. Scaling web-based acquisition of entailment relations. *Proceedings of Empirical Methods in Natural Language Processing*, 2004.

M. A. Walker, J. E. F. Tree, P. Anand, R. Abbott, and J. King. A corpus for research on deliberation and debate. In *LREC*, pages 812–817, 2012.

M. Wang and C. D. Manning. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1164–1172, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873912.

S. Wang and J. Jiang. Learning natural language inference with lstm. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1170.

J. Weeds, D. Weir, and D. McCarthy. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, 2004. doi: 10.3115/1220355.1220501. URL http://dx.doi.org/10.3115/1220355.1220501.

D. Weiskopf. Compound nominals, context, and compositionality. *Synthese*, 156(1):161–204, 2007.

P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual de-

notations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, 2(Feb):67–78, 2014.