# Relation Acquisition over Compositional Phrases

Gregory Adam Marton

February 24, 2006

**Abstract**

Relations, after morphemes and words, are the next level of building blocks of language. To successfully employ relations in language applications like unrestricted question answering, we must be able to acquire them automatically.

I propose to take two new steps towards this goal: to combine existing relation learning algorithms in a single joint or simultaneous algorithm for higher accuracy, and to begin learning relations on compositional phrases.

I show first results for each goal: For simultaneous learning, in an idealized setting, meronymy, hypernymy, and phonetic similarity together can dramatically improve acquisition accuracy of the synonym relation. For compositional phrases, I show an automatic method, already implemented, with moderately good performance.

# 1 Introduction

The single most widespread use of language technology today is in information retrieval, most notably World Wide Web search. Web search has been so successful because words in a document are highly informative about its content [39]. Yet there is much that bags of words fail to capture, that users nonetheless want access to, in particular how the entities in the world described by those words relate to each other. When information is associated with more structure, as in a relational database, then it becomes much easier to ask and answer complex queries.

One way to associate plain text with more structure is by codifying knowledge of that structure in an ontology. Ontologies identify and describe important relationships. A linguistic ontology, WordNet [52], focuses on relations applicable to almost any word, like *synonym* relationships ("car" means the same as "automobile"), *hypernym* (*is-a*) relationships (a car *is-a* vehicle), *meronym* (*part-of*) relationships (a door is a *part-of* a car), and others. In contrast, Semantic Web ontology efforts [50] are more encyclopedic in nature, codifying relationships between creators and works, companies and their headquarters, genes and proteins, and so forth.

The technological goal of the proposed research is to make the ontology building process itself easier and more automatic by extending small existing ontologies automatically using large amounts of unlabeled text. The scientific goal is to better understand the kinds of relations that are important to humans (separating them from many that are not), and to discover regularities in the ways that humans express these relations in language.

In particular, I hope to improve the state of the art in automatic relation acquisition by combining existing single-relation acquisition methods into a multi-relation method that uses each relation to disambiguate the others. I also hope to pioneer relation acquisition of a higher order: relations on already-related groups, expressed in natural language as compositional phrases.

Section 2 reviews the relevant literature in greater detail, and Section 3 outlines the methods I hope to explore.

In Section 4, I describe two experiments: the first aims to show that acquisition of a single relation can benefit from information provided by other relations, and the second aims to show that compositional phrases, like individual words, are amenable to similar automatic acquisition methods.

Finally, Section 5 discusses future experiments, lays out the resources I plan to use, and proposes criteria for evaluation.

The rest of this introduction explains the background in greater detail: what role relations play in question answering, how one can learn relations automatically from text, and why relations on compositional phrases are interesting but difficult to acquire.

## 1.1 Relations in Question Answering

Unlike keyword search engines, which assume you want documents on a particular topic, natural language question answering starts with the assumption that you have a more specific information need, often one you can express as a relation between topics of interest. A clear example is the START natural language question answering system's object–property–value model (Figure 1) [32, 33, 36].

For many of the object–property–value examples, their keywords might be enough to find a small snippet that, for a human reader, answers the question. However, to illustrate that keyword search is inadequate, consider the following pair of questions: "Who was the president before Nixon?" and "When was he born?". Searching the Web for the first question today, one gets many hits about President Nixon,

| Question | Object | Property | Value |
|---|---|---|---|
| Who wrote the music for Star Wars? | Star Wars | composer | John Williams |
| Who invented dynamite? | dynamite | inventor | Alfred Nobel |
| How big is Costa Rica? | Costa Rica | area | 51,100 sq. km. |
| How many people live in Kiribati? | Kiribati | population | 94,149 |
| What languages are spoken in Guernsey? | Guernsey | languages | English, French |
| Show me paintings by Monet. | Monet | works | [images] |

Figure 1: START's object–property–value model gives [36] several examples of questions interpreted as relations on objects in the world. Note, one can also ask about the relation: "What did John Williams do for Star Wars?" and the relations are bidirectional: "What did Alfred Nobel invent?"

*Question:* Who was the first American to walk in space?
*Answer:* Ed White
*Support:* He was the first American astronaut to make a spacewalk in 1965, and was backup command pilot for Gemini 7.
*Source:* `http://en.wikipedia.org/wiki/Edward_Higgins_White`

Figure 2: An example natural language question, the exact answer, and support for that answer.

especially where that phrase is near the word "before", but comparatively few snippets mention Lyndon Johnson. It is clear that the search engine didn't *understand*. How does one even approach the followup question? A computer might substitute "When was the president before Nixon born?", but this is unlikely to yield the correct answer, until the answer to the first question, Lyndon Johnson, is substituted instead.

The opposite extreme to keyword search is full understanding, but full understanding of unrestricted natural language questions and text is currently well beyond the state of the art.

In between keyword matching and full understanding are textual entailment relations. The phrase "in Paris" is said to *entail* "in France" in that if something happened "in Paris", then we can conclude that it happened "in France".[1] [44, 25] For the exposition to follow, consider the example question and answer text in Figure 2.

A few specialized kinds of knowledge are especially helpful in enabling these entailments, and thus helping us select good answers, namely:

- **paraphrasing**: *perform a space walk* has a paraphrase relationship with the verb phrase *walk in space*—each entails the other because they mean the same thing. **Synonymy** is paraphrasing for individual words and non-compositional phrases: *space walk* and *spacewalk* would be called synonyms.

- **hyponymy**: or the **is-a-type-of** relation, captures the fact that something of one kind is-a something of another kind. For example all *African Americans* are *Americans*, so *African American* is a hyponym of *American*, and *American* is a hypernym of *African American*; they are in a hyponymy relationship. If a question asks for an African American, then any old American may not be a good answer, but if a question asks for an American, then an African American certainly will be. In the space walk example above, hyponymy interacts with the word "first" to restrict conditions on allowable entailment: Dr. Bernard Harris, Jr., was the first African-American to walk in space, but

[1] Or "in Texas", as someone from Paris, TX may note.

his name is a bad answer because we can't tell if he is also the first American of any kind to do so. Such restrictions are a key reason to reason to learn the hyponymy relation.

- **membership**: Knowing that *Ed White* **is-a-member-of-type** *astronaut*, of *American astronaut*, and of *male* may help a pronoun-resolution system recognize that *He* at the beginning of the supporting answer must refer to *Ed White* in this context. Some lexicons like WordNet [52] make no distinction between membership and hyponymy, but while members of the type *American* are good candidate answers to our question, hyponyms (subtypes) are terrible: *"*African American* was the first American to walk in space."[2]

- **meronymy**: A meronym is a **part-of** its holonym. An engine is a meronym of a car, and the car is a holonym of its engine. Louisiana is a part of the United States of America, so the two are in a meronymy relationship. Questions about meronyms can be explicit ("What peninsula is Spain part of?") or implicit (referring to an F-16, "Who manufactures the engines?"). Meronyms also enable textual entailments like the "in Paris"→"in France" example above: X-in-meronym entails X-in-holonym (and not vice versa).

- **domain attributes**: The relations described above are general to language—almost everything enters into those kinds of relations. Individual domains have more specialized relations that are important and often asked about in those domains. For example in the space exploration domain, missions like Gemini 7 have backup command pilots, in this case Ed White. More commonly, companies have executives and headquarters locations, books have authors and prices, and flights have pilots. These domain attributes are the focus of technologies in the field of Information Extraction [28, 19], and are commonly identified using patterns that they appear in. We find entailment when patterns (X headquartered in Y; Y-based X) are combined with related pairs: "Google, headquartered in Mountain View" is a paraphrase of "Mountain View-based Google". Of course these relations are commonly asked about as well, so it is separately useful to have compiled their answers.

The kinds of relations mentioned above are the most prominent in the literature, and certainly some of the most important. There are other helpful kinds of relations: possession (Kate has a car), changes ("President" becoming "former" and "President elect" becoming "President"), adjective–action ("fast cars" refers to moving, while "fast food" refers to preparation), substance ("The Statue of Liberty is made of *steel*", as opposed to meronyms "torch", "robe", "crown", etc.), and more. Particular kinds of relations are less central to this thesis than the fact that there are a number of useful lexical relations, and that it would therefore be valuable to have a large-scale lexicon of them.

WordNet [52] is a freely available, manually constructed, machine-readable lexicon that captures synonymy, hyponymy, and meronymy relations, among others. While WordNet is an excellent resource, and is the cornerstone of many current natural language processing applications, it has several limitations. Because it is manually constructed, it grows slowly: creating a WordNet in another language or in a particular domain is difficult and expensive [74, 53]. Much knowledge can be expressed in terms of WordNet relations, but is not lexical, and thus more appropriate for an encyclopedia than for a lexicon. An example is the hypernymy between "Mercedes Benz" and "luxury car".

[2]The answer "a dog" is responsive to the question "What was the first animal to go to space?". I would analyze this one of two ways: The answer might be acceptable because the questioner intended to ask "What type of animal...", the way we might ask "What coin is worth 25 cents?". The answer might be conversationally acceptable if the respondent does not have an unambiguous identification, but can supply information about the type of that first animal. The first animals in space were fruit flies. The name of the first animal to orbit the Earth, a dog, Laika, is ambiguous with its breed.

The Semantic Web project also relies on manual annotation of relations [47], expressed in the Web Ontology Language (OWL) [50]. It has the potential to encode much more information, but still must be (painstakingly) manually constructed.

## 1.2 Automatic Acquisition

It is possible to acquire relations automatically from large corpora of text. The central idea comes from Hearst 1992 [30], in which she used a seed set of hypernyms and their hyponyms to find common patterns that expressed those pairs in a corpus, then searched for those patterns to find a much larger set of pairs. Her patterns included "such Y as A, B, and/or C", "A, B, C, (and/or) other Y", "Y, including A, B, and C", and "Y, especially A, B, (and/or) C", where Y denotes the hypernym of A, B, and C.

The automatic method she suggested was pioneered and extended by Yarowsky [77], Blum and Mitchell [10], Brin [11], and Collins and Singer [18], and was first mathematically analyzed by Abney [1]. The method of using seed pairs to find patterns in a corpus (training instances for a classifier), and then iteratively using those to find more seed pairs, is known as *bootstrapping*, or more specifically *self-training*. Another kind of bootstrapping extends this approach to two independent classifiers instead of one, and it is called *co-training*. Bootstrapping is not the only approach used for automatically acquiring such relations from corpora, but it is the most common. Bootstrapping is a form of *semi-supervised training*, and requires little initial training data (only the seed pairs). In contrast, *supervised training* approaches rely on their training data to be fully labeled. Supervised training approaches are simpler, but they often require much more labeled training data. An *unsupervised* approach would receive no training data and thus no labels. It might look for pairs of related terms, and clusters of those pairs, but would leave labeling to the user or to another method.

### 1.2.1 Paraphrasing and Synonymy

Synonyms can be gathered automatically from a corpus by characterizing each word by the context it appears in, and looking for other words in similar contexts [71, 20]. The most successful approaches use parse features (what arguments, modifiers, adjuncts, the words have), rather than linear context features (what words come before or after them), for the similarity measure [59].

Paraphrases can be discovered by looking for different patterns that connect related pairs of items in a comparable corpus [8]. For example, if one gathers many different news stories on the same topic from a small time window, then many pairs of salient words will have only one underlying connection: there are but few relationships between EgyptAir Flight 990 and the Atlantic Ocean in news from early November 1999; in most accounts it "crashed into", "plunged into", or "plummeted into" the Atlantic. From the statistics of these alternations, we can determine strengths of similarity between crashing, plunging, and plummeting into something. One can use similar techniques on multiple translations of a single original text. [5, 8]

### 1.2.2 Hyponymy

The method described above, of looking for contextual cues for hyponymy by using a seed set of examples, and iteratively gathering more examples and cues, has been the primary driver of automatic hypernym acquisition. Caraballo *et al.* [12] also added information from coordinations, observing that coordinate terms often share a hypernym. Because WordNet already provides a large set of hyponyms and hypernyms, supervised approaches have also been successful [68]. Unsupervised approaches can cluster groups of

semantically similar words (as for synonymy) and then label[3] the resulting semantic classes using features of the cluster [60].

### 1.2.3 Meronymy

The methods used for hyponymy transfer to meronymy [9, 56, 66]. Common patterns indicating a meronymy relation include explicit constructions ("consists of", "made of", "part of"), and more complex implicit constructions ("girl's mouth", "eyes of the baby", "door knob", "oxygen-rich water", "Kate has green eyes") [26]. Some of these are easily confused with other relations like possession ("Kate has a green Cadillac") or other attributes ("Kate's panache", "Kate has a headache") [26]. Clearly it would be wrong to answer that the Statue of Liberty is made of millions of visitors a year, simply because it *has* them.

Conversely, knowing that a "Press Secretary" is part of "the White House" may help us better understand the internal structure of "White House Press Secretary Scott McClellan" which is treated by many systems as a single opaque named entity, and which in turn may help us identify the structure in "White House Counsel Harriet Miers".

### 1.2.4 Membership

The methods for hyponymy and meronymy also apply for membership. [60] Aside from explicit cues like "member of", many of the cues are shared with hyponymy and meronymy. In some cases the relations are not mutually exclusive: Frank Sinatra is both a member and a part of the Rat Pack. I know of no work that has sought to distinguish the membership phenomenon.

### 1.2.5 Domain Attributes

Perhaps the greatest amount of work has been done on the subject of domain attributes, as the rich field of Information Extraction is focused on this topic. [19, 28] The earliest work was similar in spirit to the methods described above for hyponymy relations, but the seed pairs were books and authors [11]. That pioneering work by Sergey Brin was also one of the first to use the World Wide Web as its corpus. From just a few seed pairs of books and authors, Brin was able to find many books not available in Amazon.com's already extensive catalog. Recently, supervised [63], semi-supervised [2], and unsupervised [16] methods have been used to address this learning problem.

## 1.3 Simultaneous or Joint Acquisition

The ambiguity in each of the cues used in acquiring each relation limits precision: possession or "of" is often used to indicate meronymy and domain attributes; noun–noun compounds are notoriously ambiguous across domain relations, "Google CEO" vs. "Ford Escort" vs. "assistant director", and across many other kinds of relations.

Hyponyms vs. synonyms could also often appear ambiguous: the contexts that "terrier" occurs in may be similar to the contexts that "dog" appears in, e.g., "I walk my dog/terrier every morning", and especially in a domain corpus, there may not be enough evidence to make a hyponymy argument on the grounds that the hyponym "terrier" appears in a subset of the contexts that the hypernym "dog" appears in. However, if there is independent evidence, e.g., "terriers, pinchers, and other dog varieties", then the indication of hyponymy should be a strong *counter*-indication of synonymy.

[3] this step supervised

Within many domain corpora, recall will also be limited, and features may be sparse enough to make generalization difficult. In these cases making use of synonymy and paraphrasing when gathering relation cues is even more crucial than in the case with abundant data.

I propose to perform simultaneous acquisition by allowing iteration $t + 1$ of each classifier to use the iteration $t$ outputs of all other classifiers as features. Confidence that "Intel" and "Intel Corp." are similar, combined with confidence that "Intel Corp." is a "company", might lead an algorithm to be more confident that "Intel" is a "company" too, despite seeing only weak or indirect evidence in a membership classifier. A synonymy component by itself might consider "dog" and "terrier" to be synonyms due to sparseness in the distributional evidence (they happen to appear in the same contexts, as above), but evidence from a hypernym classifier that a "terrier" is a "dog" might account for the distributional similarity and make their synonymy less conclusive.

Thus by using the output of each classifier as a feature for the other classifiers, and learning all simultaneously, I hope to open the door to using these interactions as a helpful (rather than confounding) feature of language.

An alternative method for combining the acquisition results might be *joint*, rather than *simultaneous*, inference. Rather than using the decisions of each acquisition method as features to the others, a joint approach would make the decisions based on a single larger model. The advantage is that the model can find a global maximum-probability fit. The disadvantage is that the model would be more complex, more expensive to compute, and would possibly require more training data.

### 1.4 Compositional Phrases

The final challenge is to learn relations not just between individual objects or words, but over expressions that themselves include a compositional relation on their own components.

The category "American astronaut", alluded to earlier, is not a WordNet category, nor will the phrase be found in any lexicon of English because it is not a lexical item. It is a *compositional phrase*, which is to say that its meaning is specified by some composition of the meanings of "American" and "astronaut", in this case perhaps an astronaut who happens also to hail from the United States.

In most of the related literature, compositional phrases are explicitly avoided in relation learning—often only noun phrase heads (in this case "astronaut") are used. The philosophical reason is that if one knows which persons are "astronauts" and which are "American", then one should be able to take the intersection of those sets to get a meaning for "American astronaut". The practical reason is that, especially in smaller corpora, many of these longer phrases appear just once, or perhaps a few times, but usually too few times to overcome problems of data sparseness. The most frequent phrases often have added meaning beyond their compositional meaning, and become idiomatic.

Nevertheless, explicitly learning relational knowledge about compositional phrases is useful. From a philosophical perspective, ambiguous cases like "English teacher" make it impossible to tell from some texts whether the referent should be considered a member of the type "English" (person from England, as opposed to a teacher of the school subject, English). From a practical perspective, deciding *ab ovo* whether someone is an "American astronaut" is also more difficult than simply recording that the person has been described as such.

Relational knowledge about compositional phrases is vital in question answering. Many questions describe the type of the answer by a compositional phrase: "Which *American astronauts* died in the Apollo 1 fire?", or, "What *public middle school* was named in his honor?".

Acquiring relations on compositional phrases is still challenging, due to the aforementioned data sparse-

ness problem. Such sparseness problems have been overcome for other tasks by using the Web as a surrogate corpus [8], and new work suggests that sparseness over entire noun phrases can be overcome within a biomedical domain corpus as well [16].

Besides being useful in real-world applications, compositional phrase learning is interesting because it may reveal new aspects of noun-phrase-internal structure. Certainly some aspects are of the same kind as "American astronaut", combining two hypernyms. Others combine membership and domain knowledge ("White House spokesman"), hypernymy and meronymy ("car door"), spatial relations ("driver's side fender"), and possibly many other relations.

# 2 Related Work

Reasoning about answers allows programs to answer questions that must be broken down into a series of other questions to be answerable: "What is the population of the capital of the largest country in Africa?" is answerable by a reasoning system even if only the individual components of the knowledge ("sizes" of countries, their continents and capitals, and the populations of those cities) are available, and even if those knowledge components come from initially independent sources [35, 34]. The above are examples of encyclopedic knowledge, incidental knowledge about how the world happens to be in a particular domain at a particular point in time, and this sort of encyclopedic knowledge is what users are primarily interested in at the level of base facts.

A question answering system also relies on knowledge about language: syntactic regularities [37], meanings of words, types, categories and members, parts and wholes, names, events, turns of phrase, and other relationships between words. WordNet is the premier machine readable lexicon project in which experts catalogue many of these relations. [52] There have been many attempts to automate lexicographic tasks. An early and seminal attempt to automatically acquire category–member relations (hypernym–hyponym relations) to augment WordNet was by Marti Hearst [30], in which she proposed a bootstrapping algorithm—one she followed manually to look for pairs of words that fit the relation in a corpus, creating a *concordance* of them, finding common linguistic cues that indicated the relationship, and then acquiring more pairs by looking for those cues. In her example, she started with the cue "such as", and found hypernym–hyponym pairs such as "printer" *is-an* "input–output device".

In her introduction, Hearst pointed out that this method could be used not only for lexicon augmentation, but also information about noun phrase semantics (that for example a "broken bone" is a type of "injury") about which she asks us to "note also that a term like 'broken bone' is not likely to appear in a dictionary or lexicon, although it is a common locution". Though the work on automatically acquiring hypernym–hyponym pairs for lexical augmentation has received much attention, to my knowledge these noun phrase semantics have largely been unexplored. This is the topic of the proposed work.

## 2.1 Self-Training and Co-Training

In her conclusion, Hearst left automation of the process up for future work, and indeed others have considerably furthered that effort. In 1995, David Yarowsky [77] first solidified the algorithm as applied to word sense disambiguation, in which he achieved results comparable to supervised methods. In 1998 Sergey Brin [11] applied a similar method to learning books and authors from Web pages, calling the method Dual Iterative Pattern Relation Expansion (DIPRE). The paper points out several variables in the learning process, including 1. the set of patterns or features used to generate new rules from found items, 2. an item

evaluation metric, 3. a rule evaluation metric, 4. an item threshold, and 5. a rule threshold, that determine whether an item or rule is used for training in the next round, based on the metrics.

Collins and Singer [18] incorporate modifications to Yarowsky's algorithm from Blum and Mitchell [10], calling the new algorithm *co-training*. Assuming for simplicity a set of binary features, if one can partition them into two independent sets (of size $m$ and $n$), then one can reduce the size of the hypothesis space, and thus the need for training data, from $2^{m+n}$ to $2^m + 2^n$. Now instead of maximizing the number of correct predicted answers from a single classifier (*self-training*), we must maximize the size of the intersection of correct predicted answers from the two independent classifiers (*co-training*). In semi-supervised training, the available training data set is often small, so such reductions are critical.

Collins and Singer apply co-training to named-entity recognition, where their two independent feature sets, or "views" are 1. spelling features like capitalization, titles, the word "Corporation", etc., and 2. contextual features like preceding "said" or following "in". Collins and Singer use AdaBoost [24] to maximize agreement between these two classifiers on the unlabelled data. Steven Abney [1] clarifies the relevant notion of independence of the "views" and gives a theoretically better justified version of the co-training algorithm which performs equally well.

There have been a number of applications of co-training for finding relations. Riloff and Jones [65] apply self-training to finding companies, locations, and associated people on the web and locations and weapons in a terrorist domain, and they find the technique especially well suited to lexicon acquisition in a specific domain, discovering for example that in the terrorist domain, vehicles are often used as weapons. Yangarber *et al.* [75] show MUC-6 performance from self-training comparable to successful manually-tuned systems. They used two human inputs to self-training: high information-retrieval score of each candidate rule (using manual relevance judgements), and explicit human filtering of rules at each iteration. They are also the first to use parse features rather than linear ones. Two years later, [76] they applied a similar algorithm to finding "generalized names", classifying phrases like "mad-cow disease" into a fixed set of classes including diseases and symptoms, and leveraging a mutual exclusion assumption among the categories (that a disease won't also be a symptom) to incorporate negative evidence into the item evaluation metric. Pierce and Cardie [62] further explored the effects of seed size and human correction in the co-training process on an IOB[4] base noun phrase chunking task. They found that too few or too many initial training instances flatten the learning curve: given too few (0.1%), the classifiers lose accuracy quickly, whereas given too many (0.5%), the classifier accuracy has little room to improve before the data are contaminated. If one prevents labelled data degradation by introducing human correction, then the classifier accuracy does not degrade, and the accuracy of the fully supervised classifier is achieved using just 4% of the available training instances, including (simulated) human correction. Thus semisupervised methods can be as effective as supervised methods, and can use far less training data, especially if given human correction at each iteration.

## 2.2 Application to Question Answering

There is a great deal of work that uses self-training and co-training methods and I discuss only a few examples here.

Stevenson and Greenwood [69] used self-training for the MUC-6 Information Extraction task, Hasegawa *et al.* [29] clustered results of self-training over pairs of named entities to find the most significant relations

[4] inside/outside/begin (IOB) notation is often used for chunk labelling tasks, marking each token as inside a markable chunk, outside any markable chunk, or beginning a markable chunk. A named-entity example: "Honda/B Corporation/I 's/O 2006/B Acura/B sales/O"...

in a text, and Geffet and Dagan [25] used self-training on anchor words and words of interest in a single-corpus paraphrase alignment task in support of their work on lexical entailment. Each of these approaches addresses the acquisition of knowledge about the world or knowledge about language that can be directly used in question answering.

MUC-6 style information extraction is useful for answering commonly asked questions about a domain, both because it captures the answers and because it captures in the rules a wide variety of ways to express the relations in question. The ways to express the relation intuitively should be usable in matching an unseen question to a type of relation already extracted into a structured source. I know of no work directly addressing this challenge, and it will be among the first aims of my research.

If the set of interesting relations can also be found automatically, following Hasegawa, then more of the burden of guessing what questions will be asked, is alleviated.

Glickman and Dagan introduce textual entailment in the context of question answering, casting a question as a request for an entailing phrase. They also cast paraphrasing as a special case of textual entailment in which the paraphrases mutually entail one another, and suggest that for question answering only entailment, not mutual entailment, is required. While their present work only addresses lexical entailment (single words), their methodology may prove useful in the broader task.

Another resource for question answering that may benefit from self-training and co-training approaches is the Automatic Content Extraction (ACE) task [64], which involves entity detection and tracking (EDT) as well as relation extraction among the entities. The ACE inventory of relations is fixed and perhaps domain-specific, but may prove a useful testbed. Yangarber *et al.* [76] present the NOMEN self-training algorithm for locating "generalized names". Generalized names include non-compositional phrases that are used in relevant relations in the text, even if they are not capitalized, such as "mad cow disease". Mueller *et al.* [57] report mostly negative results on co-training for reference resolution (entity tracking), but interestingly showed a significant improvement over baseline in one of the more difficult resolution cases: when the anaphor is a definite noun phrase. Markert and Nissim take this observation further [48] to show that automatic web-based mining of WordNet-like knowledge (but knowledge which goes beyond what is available in WordNet) significantly improves performance on the definite NP coreference task.

Finally, Chklovski and Pantel [14, 13] explore fine-grained verb semantics, showing that not only verb similarity, but also strength, antonymy, enablement, and temporal relations, can be acquired using semi-supervised methods, and suggesting ways that these could be used in question answering and natural language inference.

Thus a number of important problems for question answering have been addressed by semi-supervised learning methods, showing promise. As yet there is significant room for improvement in each of these tasks, and little work in applying the resulting classifiers to question answering and evaluating their contribution in the end-to-end task.

## 2.3 Related Problems and Methods

Semi-supervised learning as highlighted above is clearly not the only approach to problems in knowledge acquisition and question answering.

Caraballo [12] clustered nouns using cosine similarity, found candidate hypernyms in much the same way Hearst did, and had each cluster vote on its parent. More recently, Snow [68] learned characteristic paths in Minipar dependency parse trees, based on a seed set of known pairs, instead of using fixed syntactic patterns, and achieved both higher precision and higher recall than WordNet. Both used similarity, as given by words appearing in coordinate conjunctions, to propose hyponym–hypernym pairs that weren't

seen in any explicit pattern.

Some lexical relations that have been addressed in the literature include synonymy [27, 42, 45], hypernymy and meronymy extending WordNet [30, 68, 54, 70, 15], adjectival modification [40], noun-noun compounding [67], nominalization [41], sense disambiguation [51], paraphrases [8, 58, 5], and entailment [44, 21]. Other kinds of relations have also been sought: authors and titles [11], corporate headquarters [3], corporate acquisition and corporate production [55], and others.

Noun-phrase-internal structure learning is both a potential informant and a potential application of the proposed research. Lapata finds a correlation between verb–argument statistics and related adjective–noun modifications, for example noting that cars often "speed" down a roadway and that there are correspondingly "fast cars" [40]. She finds similar relationships for noun–noun modifications, for example that "vegetable soup" expresses a "have" relation, "peanut butter" expresses a "from" relation, "sound synthesizer" expresses a *verb–object* relation, and so on. Rosario and Hearst use similar methods to characterize noun–noun modification in the biomedical domain [67]. The semantic relations we seek may be closely related to those expressed in these noun-phrase-internal modifications.

More general approaches to paraphrase acquisition rely on comparable corpora: texts that talk about the same thing, but with different wording. Sources of comparable corpora include multiple translations of the same text, where presumably different translators express the same underlying ideas in different ways [7, 31, 58]; multiple versions of the same news story, as from different news agencies [6]; and surprisingly, multiple translations in parallel corpora of the same phrase in *different* contexts, rather than in a single context [5]. That the last approach is effective is surprising because one has to be careful to ensure that the same word or phrase is being used in the same sense in the two unrelated contexts, despite having different translations.

Thus there are many interesting and effective approaches towards the same two goals: learning semantic relations between words, and finding different ways of expressing the same idea. The methods described in this section are ripe for systematic combination into a larger system, because they could each benefit from a richer feature space expressing guesses at the others' outputs. Moreover, each approach is applicable to the complex compositions of words I seek to characterize.

# 3 Approach

The first task is to allow several single-relation acquisition processes to gain information from each other's progress. The second is to modify single-relation acquisition to be able to learn relations on compositional phrases.

## 3.1 Single-Relation Classifiers

The traditional component in each single-relation classifier must extract relevant candidates from each sentence. In addition to linear patterns, I plan to use paths in parses of sentences, where available. I will adopt Snow's method of learning paths in parse trees [68], applying it to Minipar [43], Stanford Parser [38], and Collins [17] parses. The three give different views of the sentence (dependency with empty categories, binary dependency, and constituency parses, respectively) and may yield different kinds of results.

The non-traditional component in the single-relation learning task will be use of features from other single-relation learners. We can use, from every other classifier, its score for its relation between any two words globally, or we can ask that classifier whether a particular instance matches well.

For a phrase "driver's side fender", for example, the meronymy learner might globally have lower probability of a driver having a fender or a side fender than of having a side, but it might locally assign high probability that the possessive applies syntactically between driver and fender. Another single-relation learner might be interested separately in each of the probabilities, of a driver having a side or fender in general, and of this driver having the side vs. the fender in this particular instance. Of course to get the relation right, a classifier would have to see the unseen word "car" and note that it has sides, a driver, and a fender (globally, even though there is no car here).

Combining parse features with external features for a candidate relation to yield a "yes" or "no" for the current single relation is a standard classification problem, with many existing tools.

### 3.2 Many-Relation Classifiers

One approach to simultaneous acquisition is to use co-training. A minor extension to the co-boost algorithm from [18] (reproduced and extended in Figure 3) allows us to run $J$ learners, one at a time, feeding each one the results of the others among its feature set for each candidate instance. This is the most general method for simultaneous acquisition, and introduces no constraints of its own: it simply allows each learner to make decisions informed by the others.

Using multiple relations as features is strictly weaker than classifying into one of several relations, because it does not rule out the possibility of a pair having more than one relation. In addition, the sum used to combine the verdicts of the classifiers (in the Update phase) is a good place to start, but adding a separate classifier, or at least learning weights for the sum using additional training data, may improve performance.

As an alternative to co-training, Ando and Zhang [4] present *structural learning*, a joint inference method over many auxiliary classifiers. The single-relation classifiers in this proposal correspond to Ando and Zhang's partially-supervised strategy for generating auxiliary problems. Ando and Zhang point out that many auxiliary problems can be generated in an unsupervised way by ablating individual words from the unlabeled data and creating an auxiliary classifier that tries to predict them. One could, for example, try to predict either end of a candidate relation pair based on the other end and the path. While Ando and Zhang were interested in named entity classification and syntactic chunking, their method appears directly applicable to this task.

### 3.3 Compositional Phrases

The final challenge is to learn relations not just between individual objects or words, but over expressions that themselves include a compositional relation on their own components. Most current methods for relation acquisition go to some trouble to find the head of the phrase at each end of a candidate relation, and only gather statistics on that head word. The reason is twofold: relevant data for phrases are sparser for phrases than for heads, and the decomposition into sub-phrases is not always clear.

I plan to attack the data sparseness issue by looking outside the project's primary corpora for instances: there may be only a few mentions of a "driver's side fender" in one corpus, but Google reports some 5,000 hits on the phrase, and Yahoo over 15,000. There has been some debate about the accuracy of search engine counts and results for linguistic analysis, but even a few extra uses give valuable evidence, and the Web has been successfully used for similar statistics-gathering purposes by others [48].

An additional tempting avenue for overcoming data sparseness in compositional phrases is to discover compositional paraphrase rules based on regularities in the data. If "Bill Gates" is both a "Microsoft CEO" and a "chief executive of Microsoft", and we see similar patterns among other chief executives, e.g.,

For $J$ independent classifiers over $m$ labelled and $n$ unlabelled examples,
Input: $\{(x_{1,i}, x_{2,i}, \ldots, x_{J,i})\}_{i=1}^{n}, \{y_i\}_{i=1}^{m}$
Initialize the predictions $g_j^t(e)$ for each instance $e$ and learner $j$ at time $t$: $\forall i, j : g_j^0(x_{j,i}) = 0$.
For $t = 1, ..., T$ and for $j = 1, ..., J$:

- Set pseudo-labels:

$$\tilde{y}_i = \begin{cases} y_i & 1 \leq i \leq m \\ \text{sign}(\sum_{j' \neq j} g_{j'}^{t-1}(x_{j',i})) & m < i \leq n \end{cases}$$

- Set virtual distribution:

$$D_t^j(i) = \frac{\exp\left(-\tilde{y}_i g_j^{t-1}(x_{j,i})\right)}{\sum_{i'=1}^{n} \exp\left(-\tilde{y}_{i'} g_j^{t-1}(x_{j,i'})\right)}$$

- Get a weak hypothesis $h_t^j : 2^{X_j} \to \mathbb{R}$ by training weak learner $j$ using distribution $D_t^j$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update:
  $\forall i : g_j^t(x_{j,i}) = g_j^{t-1}(x_{j,i}) + \alpha_t h_t^j(x_{j,i})$.

Output final hypothesis:

$$f(x) = sign\left(\sum_{j=1}^{J} g_j^T(x_j)\right)$$

Figure 3: The generalized CoBoost algorithm for binary classification.

of Merck and of Nokia, then we might have enough data to hypothesize that "chief executive of X" is a paraphrase of "X CEO". Though this idea is intuitively appealing, I am not yet in a position to test whether the data are amenable to such analysis.

The decomposition problem is much more difficult. Being conservative, one can gather statistics only on the entire phrase, or the entire phrase and the head. In a preliminary experiment, in joint work with Fernandes and Tellex at the Infolab group, we used a much more liberal method of removing modifiers one at a time, and counting each one separately (Section 4). Of course each parser will make some guess as to the structure, as would a method like Yuret's lexical attraction model [78], and we can use each guess about the structure as a decomposition, yielding different results. This problem may become fertile ground for experimentation.

# 4 First Results

I describe two experiments: the first aims to show that acquisition of a single relation can benefit from information provided by other relations, and the second aims to show that compositional phrases, like individual words, are amenable to similar automatic acquisition methods.

## 4.1 Simultaneous Acquisition

Upper bounds for the impact of simultaneous acquisition on relation learning can be found by looking at the amount of confusion in automatically acquired relations with other relations. One such relation is synonymy, where previous results of automatic acquisition are available [44]. The results are in the form of pairs of words, each associated with a similarity score. There is a global lower threshold of 0.04 similarity in the data provided, so pairs less similar have effectively unknown similarity. If the most similar words are correctly assigned the greatest similarity score, then words with the same meaning—words in the same WordNet synset for example—ought to have the highest scores.

I selected those 2142056 pairs from Lin's data set of 7144700 pairs where both words had a sense in WordNet. I found shortest paths in WordNet for these words, interpreting polysemous words with the sense closest to any sense of the the paired word.

For a baseline, we can measure precision and recall of same-synset pairs at each rank, and determine a rank with a maximum F-measure.[5] That rank is 15078th with F-measure=0.08.

Performance is so low because related non-synonymous terms are not well separated from synonymous terms. To fix this, we must find other information that can help discriminate between same-synset and merely related pairs. Along with similarity scores, Lin provides frequency information for each word. We might also use information about how similar the pair of words sounds (thus "nonproliferation" and "non-proliferation" would be phonetically close, as would "Muammar Gadaffi" and "Moamur Quadhafy") [61]. We might use hypernym information: are they related within 8 steps in the hierarchy, is one a hypernym of the other, if they are sisters or cousins how many hops to the lowest common parent, do they share a hyponym? We can ask similar questions from the meronymy hierarchy.

In the experiment presented in Figure 4, I use each of the features described above (Lin's similarity scores, phonetic similarity, hypernym path, meronym path) to train a classifier (Weka J48 decision list, 10-fold cross-validation) to yield yes (same-synset) or no (not same-synset). I chose a decision list because it transparently shows which features are most informative about the problem.

[5] F-measure is the harmonic mean of precision and recall, tending to balance the two.

| **features and method** | classification accuracy | yes F-measure |
|---|---|---|
| sim+freq+mero+phon+hypo DL | 99% | 89% |
| sim+freq+mero+phon DL | 99% | 88% |
| sim+freq+mero+hypo DL | 96% | 57% |
| sim+freq+mero DL | 96% | 57% |
| sim+freq+phon+hypo DL | 96% | 41% |
| sim+freq+phon DL | 95% | 19% |
| sim+freq+hype DL=no | 95% | 0% |
| sim+freq DL=no | 95% | 0% |
| sim threshold | 87% | 8% |

Figure 4: Classification accuracy and accuracy of same-synset (yes) judgements with using different features and classification methods.

With only the similarity and frequency information, the decision list learner was unable to separate the two classes, and always answered no. Always saying no yields a classification performance of 95% (because 95% of the pairs are not same-synset) but an F-measure for yes-classification of 0 (because yes-recall is zero). If our primary goal is classifying same vs. different synset pairs, then this is better than the single-threshold baseline (95% vs. 87%). But if our primary criterion is how well it classifies same-syset pairs, then not classifying any at all is clearly worse than the baseline 8% correct.

Can we do better by adding hypernym information? No. Most of the pairs in this dataset are closely related in the hypernym hierarchy. Thus if they are only distantly related, then we can clearly say no, but close relation doesn't help us in saying yes.

We can do better by adding phonetic and meronym information. Pronetic information alone brings yes F-measure up to 19%, and meronym information alone brings it up to 57%. Combining all three additional information sources achieves 89% accuracy in same-synset judgements, clearly an improvement over the single-threshold baseline. This is one dramatic example where lexical relations can help in the classification of what was previously thought to be an independent relation. I surmise (but have yet to show) that this is generally the case: that relations among words will help disambiguate each other.

### 4.2 Compositional Phrases Acquisition

The second experiment is a preliminary implementation of an automatic single-relation classifier for the hypernym relation, that also captures hypernyms on compositional phrases [49]. Hypernyms are classified using regular expression patterns (Figure 5), following [30, 23, 22], then ranked by their frequency in the document collection, and discriminated using a cutoff on estimated precision.

We collected compositional phrase candidates as proposed in the Approach section above: when a participant in a candidate relation was a phrase rather than a single word or noun–noun compound, then rather than counting only the head, we counted: 1. the whole phrase, 2. variants of the phrase with postmodifiers successively removed, 3. variants of the phrase-sans-postmodifiers, with premodifiers successively removed, and finally 4. the remaining head (Figure 6).

About one million candidate pairs were gathered, and human evaluation was performed on a sample of more than 600 of these, and precision was measured at close to 50%. To measure recall, we looked at the question focuses for TREC factoid and list questions from several years [72, 73, 72], and looked for

| Pattern cue | $p(z)$ | Frequency | Example |
|---|---|---|---|
| common noun then name | 0.75 | 2,125,812 | *President* Clinton |
| apposition marked by commas | 0.89 | 625,962 | Noemi Sanin, *a former foreign minister*, |
| plural then like | 0.42 | 158,167 | *immunisable diseases* like polio. |
| such as or such *X* as | 0.47 | 118,684 | *stinging insects*, such as bees, wasps, hornets and red ants, |
| called or also called | 0.70 | 14,066 | a *game* called Tightrope walker |
| named then proper name | 0.64 | 8,992 | an *Armenian* named Wilhelm Vigen |
| known as, also known as | 0.68 | 8,199 | *low-tariff trade rights* known as most-favored-nation status |

Figure 5: Examples of the seven patterns expressing hypernym relations. Precision ($p(z)$) and Frequency of each pattern are shown for a million-article body of newspaper text, along with an example *hypernym* and hyponym in context.

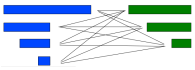
| | | |
|---|---|---|
| original pair: | *stinging insects* + red ants | *a new dedicated consultant for the CommunicAsia Conference* + Kim |
| remove articles: | | *new dedicated consultant for the CommunicAsia Conference* + Kim |
| remove postmodifiers: | | *new dedicated consultant* + Kim |
| remove premodifiers:  | *stinging insects* + ants<br>*insects* + red ants<br>*insects* + ants | *dedicated consultant* + Kim<br>*consultant* + Kim |

Figure 6: Modifier removal steps: first, articles are stripped, then postmodifiers and premodifiers are removed, one at a time, from each end of the pair, and all combinations are considered.

correct answers in the matching categories: recall was estimated at close to 30%.

In analyzing each question focus, we created variants of the focus using the same modifier removal algorithm as that used for generating partial phrases for relation acquisition. With the learned hypernym–hyponym pairs, we were able to match 91% of question focuses, up from 75% using a non-learned ontology. More importantly, we matched 21% of multiword question focuses (e.g., "Hezbollah members", rather than "members"), up from 7%, indicating that relevant phrases were, in fact, being captured.

Among the highest confidence compositional hypernyms[6] captured: "defending champion", "police spokesman", "news agency", "opposition leader", "former President", "committee member", "civil rights activist", "gold medalist", "rebel group", "electronics giant", "oil company", "country singer". This technology will enable us to discriminate much more specific descriptions than noun phrase heads alone ("champion", "spokesman", "agency", "leader", "President", "member", "activist", "medalist", "group", "giant", "company", "singer") would provide.

[6]In the present terminology these would be categories with *members* rather than hyponyms, but much of the literature does not distinguish between the two relations.

# 5 Experimental Framework

In this section, I describe the questions I plan to answer, the corpora I plan to employ in answering them, and the particular forms of evaluation I plan to use throughout the process.

## 5.1 Goals

Experiments indicated by the proposed approach include:

- Implementing semi-supervised learning approaches for each parser, obtaining single-relation results on local corpora with various parameter settings of confidence and iterations, and comparing the results.
- Composing single-relation results using the many-relation methods described, testing the hypothesis that the composition will yield better accuracy than each single-relation method on its own, and documenting the greatest influences among single-relation classifiers. Those classifiers that influence each other most will be examined, seeking corresponding regularities in the world.
- Implementing and comparing methods for breaking down candidate pairs into their compositional components.
- Ameliorating data sparseness in single-relation candidates by developing a robust strategy for using data from the World Wide Web to augment low-frequency information in the corpus, especially for compositional phrases.
- Ameliorating data sparseness in compositional phrase candidates by discovering compositional structural paraphrases, or at first, by simply implementing a few key structural paraphrase rules.

## 5.2 Corpora

Prior work on automatically learning relations from text emphasizes the value of more data over cleaner data. At the same time, the questions to be answered for a standard extrinsic evaluation in question answering (to be described) pertain to a specific domain: news between 1998 and 2000. The best systems will make use of rich annotation on the small corpus in the target domain, as well as poor annotation on a large corpus in the open domain.

### 5.2.1 TREC and AQUAINT

The TREC and AQUAINT corpora of English newspaper text have sets of questions and answers associated with them, thanks to the annual TREC Question Answering evaluation program. Thus these corpora will serve as the small, richly analyzed, domain-specific texts.

The TREC corpus is part of the LDC Tipster collection (LDC93T3A) including news from the Associated Press (1988-1990), Wall Street Journal (1987-1992), San Jose Mercury news (1991), Ziff/Davis technical news (1989-1992), and documents from the Department of Energy, and the Federal Register (1988-1989). The AQUAINT corpus includes news articles from The New York Times (1998-2000), Associated Press (1998-2000), and Xinhua English News (1996-2000). Together these collections contain over 8Gb of edited English text.

The Infolab group has annotated the AQUAINT corpus with named entity information, as well as Minipar parse information, and are on our way to annotating it using Collins and Stanford parsers. I plan to use these annotations to learn syntactic patterns as opposed to purely textual ones.

### 5.2.2 The World Wide Web

Much automatic knowledge acquisition has been performed using the World Wide Web as a source of text, either through cached pages (from a spider or web crawler) or on the fly. For a broad knowledge development task, we of course would want access to a large number of web pages on all of the topics covered. Collecting web pages relevant to the domain and to the specific questions is also possible and perhaps even desirable for the question answering task. The Infolab has done so for several years as part of the Aranea project [46] (now open source), and could use cached pages and live links instead of only the available snippets for larger-scale text analysis and on-the-fly knowledge building.

On the World Wide Web, I plan to apply existing textual patterns [22] as well as learning new ones.

### 5.2.3 The Wikipedia

Though comparatively small, the Wikipedia has three main advantages: it is easily available, it is unencumbered by intellectual property concerns, and it is topical. The last point is especially important, because indeed, the text is *designed* as a resource for those wanting to learn about the topic, as well as people from a wide variety of backgrounds. Thus the language tends to be accessible, more correct than not, and many of the topics of interest to the TREC and AQUAINT news corpora are covered, ensuring some semantic overlap.

The Wikipedia, in its link and disambiguation structure, also encodes much of the underlying meaning structure of the domain. Each article represents a synset; separate identifiers are used besides the headword, to identify senses (e.g., "Nirvana", "Nirvana (band)", "Nirvana (album)", "Nirvana (1960s band)", "Nirvana (leafhopper)", and "Nirvana (movie)" are currently encoded senses of "Nirvana", made explicit, complete with glosses, on a special page "Nirvana (disambiguation)"); subtitles and redirects from other titles are used to indicate synonymy (e.g., "Nirvana (movie)" from the disambiguation page currently redirects to "Nirvana (film)"). Categories, lists, and list pages are used to indicate hypernymy (e.g., "Nirvana (film)" has categories "1997 films", "Science fiction films", and "Italian films").

I anticipate that many such "wikis" will be created for those humans learning about a new domain of discourse, and there will be increasing incentive to codify information in this form, if automatic language learning algorithms can make use of it. Learning from the other contexts may also help populate and link portions of the Wikipedia, and thus be subject to review (evaluation) by a large number of enthusiastic human annotators.

## 5.3 Evaluation

Evaluating a large scale ontology with compositional elements is not straightforward, because there is—almost by definition—no ground truth of the appropriate scale. Intrinsic evaluation is always possible for a relation by sampling the related items, providing examples in context, and asking a human to evaluate them. Along these lines, I will specify a method of presentation in context for each kind of knowledge I wish to test.

One can also evaluate the knowledge bases extrinsically, by their contribution to the performance of a larger system, for example as components in a question answering system. Here, the match between the

strengths of the knowledge base and the way that the larger system uses it can account for a large variation in performance between different systems using the same knowledge base. It is therefore important not only to test actual performance, but to evaluate limits on possible performance given oracular sources.

### 5.3.1 Hypernymy and Other Directed Relations

The most important relation to focus on with respect to evaluation is hypernymy, both because it is the area with the most present work with semi-supervised and other approaches well represented, and because it has the most direct measurable impact on question answering performance.

Evaluation for hypernym acquisition will concern:

- How many question focuses can be matched to an acquired relation (category recall)
- How many correct answers can be found in any acquired relation (member recall)
- How often a hypernym matched to a question focus has a correct answer as a hyponym (answer recall)
- For a random sample of hypernyms, how many of their hyponyms are correct (hyponym precision)
- For a random sample of hyponyms, how many of their hypernyms are correct (hypernym precision)
- How many of the available hyponyms for matched focuses are correct (matched hyponym precision)[7]
- Given the set of hypernym relations and a fixed question answering system, how much improvement in question answering performance is derived on standard data sets.

I anticipate evaluation by comparison with WordNet for those relations where WordNet has pertinent information, and for that subset of acquired relations that are in WordNet. Manually constructed hypernym (or often instance) relations also exist in Wikipedia in the form of list pages, lists within pages, and categories.

Similar evaluation can also be performed for many other target relations.

### 5.3.2 Synonymy

One relation, synonymy or paraphrasing, is symmetric and reflexive and thus better viewed as a clustering problem. In this case standard clustering metrics will apply. Given a reference clustering (like WordNet or a new one we construct):

- How many pairs of items in the same category in the reference are also in the same category in the observed clustering?
- How many pairs of items in different categories in the reference are also in different categories in the observed clustering?

[7] It is hard to evaluate the corresponding answer hypernym precision because we do not have ground-truth, so it would essentially be the same as hypernym precision on a random selection.

These can be performed for the subset of synonyms also in WordNet, and again Wikipedia offers a large set of synonym links via its redirect structure and subtitles. It may be difficult to find reference clusterings for compositional phrases, however. Wikipedia contains many phrases, and though these may not all be compositional, they can still serve as a manual benchmark for evaluation. I can finally resort to human evaluation of a random sample of clusters, using standard paraphrase evaluation procedures, presenting the items in context and measuring human ratings of interchangeability in context. The more kinds of contexts allow either item without changing meaning, the more synonymous the two items are.

Presentation of two items in an artificial coordination may be another way to measure interchangeability. Intuitively, if two items are close paraphrases (e.g., Microsoft and MSFT) then it is infelicitous to conjoin[8] them: "He worked for Microsoft and MSFT." If two items are related but distinct, then it is felicitous to coordinate them: Is your phone manufactured by Nokia or Motorola? Measuring anomalousness of such automatically generated test phrases can thus inform effectiveness of clustering. We will also be looking for these constructions as negative cues in the text during learning.

Synonymy can also be evaluated in the context of question answering. Synonymy and paraphrases augment a lexicon by allowing relations learned for one term to apply to relations learned for synonymous terms. Thus if the knowledge base contains a list of "cellular phone manufacturers", and we can determine that these are synonymous with "cell phone makers", then we can potentially increase the upper bounds on recall in the other relations. As such, the same extrinsic evaluation measures outlined above are also useful for synonymy.

The TREC QA relationship task in 2005 was intended to push the limits of semantics by asking long, complex questions that required knowledge from multiple documents to answer. The Infolab system did well based on a precision and recall-style approach to passage retrieval. Our attempts at using synonymy, presented in Figure 7, paint a surprising picture. Though individual synonymy approaches did yield improvements in precision and recall, the difference between using all methods ("everything") vs. using none ("nothing") in the region of highest F-measure is close to zero. It seems that the different synonymy modules cancel out each other's performance gains. High precision of the "everything" method at either end is encouraging: at top-1 answer, the program seems to know when it is right, and at many answers, it seems that matching more terms, even approximately, yields an improvement. High precision of the "nothing" method at around 30% recall is unsurprising: exactly matching terms are expected to yield higher precision than approximately matching terms. Because these data represent averages over only 25 data points (the relationship task had 25 quetsions), none of the differences is statistically significant.

### 5.3.3 Compositionality

Finally, because I focus on acquisition of compositional phrases in these relations, it may be possible to acquire and evaluate the Restrictivity and Monotonicity properties of various compositional textual extensions:

- Prediction of Restrictivity: If one hypernym is a textual extension of another, and its hyponyms are a subset of the other's, then the extension is restrictive.
- Prediction of Monotonicity: If hypernyms A and B are similar textual extensions of hypernyms a and b respectively, and they share the same subset relationship with their respective hyponyms, then the extension is monotonic.

[8] Disjunction is less reliable because "or" can also present another name: "the Department of Justice, or DOJ,..."

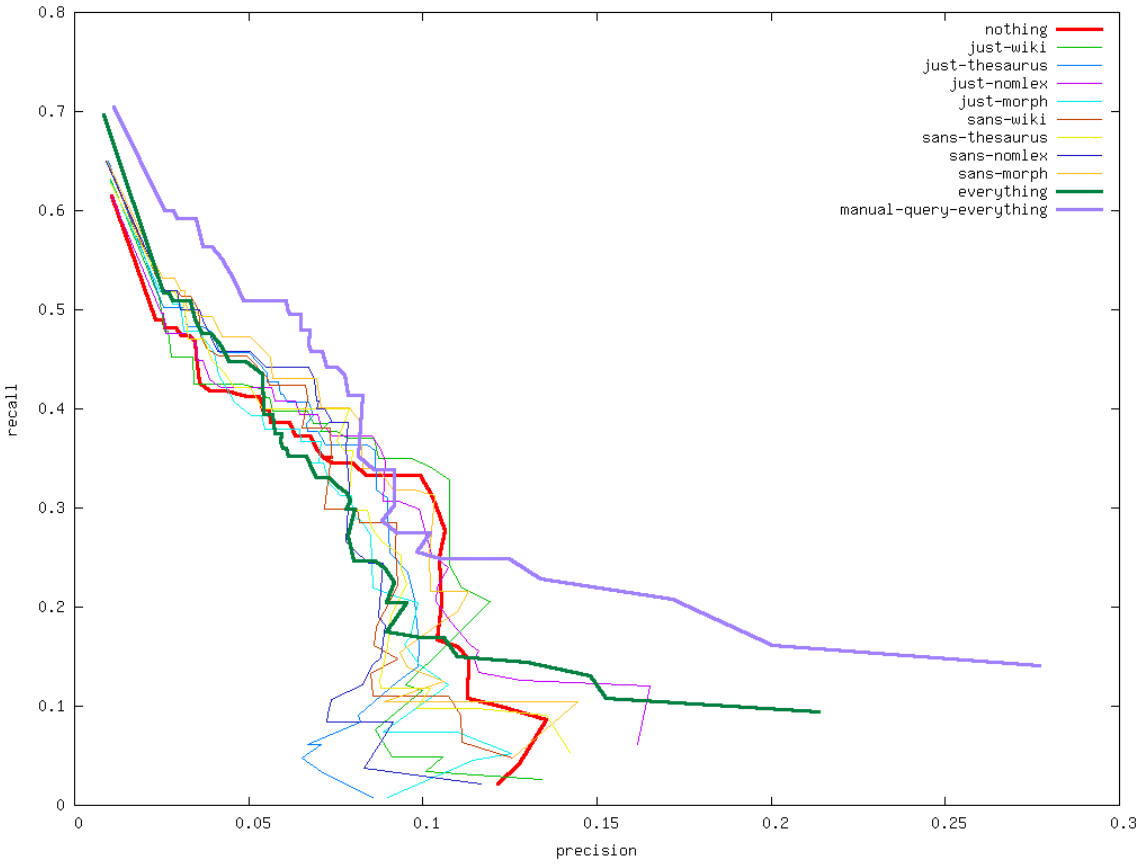



Figure 7: Word variant sources for relationships: bolder lines show exact match only ("nothing"), all synonym sources used ("everything") and for reference, the performance with manual query processing and all sources used ("manual-query-everything"). Results are visualized in terms of precision vs. recall, where each line proceeds from top-1 output cutoff (lower right) to unlimited output.

Compositionality can serve as a unidirectional analog to synonymy for question answering. If an extension is restrictive, then for many questions we may be able to use that extension to boost the upper bound of recall. For example, the extension "French" as a premodifying adjective is usually restrictive,[9] so when asked "Which cities...", we can augment the answer type "cities" with any restrictive modifier to include other categories, like "French cities", among the set of answer candidates to search for. Thus compositionality, like synonymy, can be evaluated extrinsically: if the compositionally expanded candidate set performs better than the exactly-matching candidate set, then compositionality is helping.

## 6 Contributions

I have proposed an effort to build a large-scale knowledge base for question answering that encodes many of the kinds of relations useful in common natural language tasks, and particularly for question answering. This knowledge base will effectively automatically extend manually built resources like WordNet.

In building this knowledge base, I will explore simultaneous learning methods, and phenomena in compositionality. I will evaluate components of the resulting knowledge base intrinsically, and also via their effect on question answering performance.

## References


[1] Steven Abney. Bootstrapping. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL2002)*, 2002.

[2] Eugene Agichtein. *Extracting Relations from Large Text Collections*. PhD thesis, Columbia University, 2005.

[3] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *The proceedings of the 5th ACM International Conference on Digital Libraries (DL)*, 2000.

[4] Rie Kubota Ando and Tong Zhang. A high-performance semi-supervised learning method for text chunking. In *The proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL2005)*, 2005.

[5] Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *The proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL2005)*, 2005.

[6] Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, pages 16–23, 2003.

[7] Regina Barzilay and Kathleen R. McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL/EACL 2001)*, 2001.

[8] Regina Barzilay and Kathleen R. McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–327, 2005.


[9] Except arguably in some collocations like "French fries" where the "base" form is unlikely to appear with any other related restrictive modifier like "German fries".

[9] Matthew Berland and Eugene Charniak. Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL1999)*, 1999.

[10] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1998.

[11] Sergey Brin. Extracting patterns and relations from the World Wide Web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, 1998.

[12] Sharon Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL1999)*, 1999.

[13] Timothy Chklovski and Patrick Pantel. Large-scale extraction of fine-grained semantic relations between verbs. In *The proceedings of the KDD workshop on mining for and from the Semantic Web (MSW2004)*, 2004.

[14] Timothy Chklovski and Patrick Pantel. Verb ocean: Mining the Web for fine-grained semantic verb relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2004)*, pages 33–40, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[15] Massimiliano Ciaramita. *Automatic Acquisition of Lexical Semantic Information*. PhD thesis, Brown University, 2002.

[16] Massimiliano Ciaramita, Aldo Gangemi, Esther Ratch, Jasmin Saric, and Isabel Rojas. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *The proceedings of the International Joint Conference on Artificial Intelligence (IJCAI2005)*, August 2005.

[17] Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

[18] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing (EMNLP/VLC-99)*, 1999.

[19] Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, 1996.

[20] Carolyn J. Crouch and Bokyung Yang. Experiments in automatic statistical thesaurus construction. In *Proceedings of the 15th annual meeting of the ACM Special Interest Group in Information Retrieval (SIGIR1992)*, June 1992.

[21] Ido Dagan and Oren Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. In *Learning Methods for Text Understanding and Mining*, 2004.

[22] Aaron Fernandes. Answering definitional questions before they are asked. Master's thesis, Massachusetts Institute of Technology, 2004.

[23] Michael Fleischman, Edouard Hovy, and Abdessamed Echihabi. Offline strategies for online question answering: Answering questions before they are asked. In *ACL2003*, 2003.

[24] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *The proceedings of the Thirteenth International Conference on Machine Learning*, 1996.

[25] Maayan Geffet and Ido Dagan. The distributional inclusion hypothesis. In *The proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL2005)*, 2005.

[26] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the North American Association for Computational Linguistics and the Human Language Technologies Conferences (NAACL/HLT2003)*, 2003.

[27] Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[28] Ralph Grishman. Information extraction: Techniques and challenges. In *SCIE*, pages 10–27, 1997.

[29] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd annual meeting of the Association for Computational Linguistics (ACL2004)*, 2004.

[30] Marti Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th international conference on computational linguistics (COLING1992)*, July 1992.

[31] Ali Ibrahim. Extracting paraphrases from aligned corpora. Master's thesis, Massachusetts Institute of Technology, 2002.

[32] Boris Katz. Using English for indexing and retrieving. In *Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88)*, 1988.

[33] Boris Katz. Annotating the World Wide Web using natural language. In *Proceedings of the Conference on the Computer-Assisted Searching on the Internet, RIAO97*, 1997.

[34] Boris Katz, Gary Borchardt, and Sue Felshin. Syntactic and semantic decomposition strategies for question answering from multiple resources. In *Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering*, pages 35–41, July 2005.

[35] Boris Katz, Gary Borchardt, and Sue Felshin. Natural language annotations for question answering. In *Proceedings of the 19th International FLAIRS conference*, May 2006.

[36] Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*, June 2002.

[37] Boris Katz and Beth Levin. Exploiting lexical regularities in designing natural language systems. In *In the Proceedings of the 12th International Conference on Computational Linguistics (COLING1988)*, 1988.

[38] Dan Klein and Christopher Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics (ACL2003)*, 2003.

[39] Robert Krovetz. Homonymy and polysemy in information retrieval. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 72–79, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[40] Maria Lapata. A corpus-based account of regular polysemy: The case for context-sensitive adjectives. In *Proceedings of the North American Association for Computational Linguistics Conference (NAACL2001)*, 2001.

[41] Maria Lapata. The disambiguation of nominalizations. *Computational Linguistics*, 28(3):357–388, 2002.

[42] Dekang Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774, 1998.

[43] Dekang Lin. LaTaT: Language and text analysis tools. In *Proceedings of the Human Language Technologies Conference (HLT2001)*, 2001.

[44] Dekang Lin and Patrick Pantel. DIRT @SBT@discovery of inference rules from text. In *Knowledge Discovery and Data Mining*, pages 323–328, 2001.

[45] Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *IJCAI*, pages 1492–1493, 2003.

[46] Jimmy Lin and Boris Katz. Question answering from the Web using knowledge annotation and knowledge mining techniques. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM 2003)*, November 2003.

[47] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE intelligent systems and their applications*, 16(2):72–80, March 2001.

[48] Katja Markert and Malvina Nissim. Comparing knowledge sources for nominal anaphora resolution. *Computational Linguistics*, 31(3):367–401, 2005.

[49] Gregory A. Marton, Stefanie Tellex, Aaron Fernandes, and Boris Katz. Hypernyms as answer types. In *Proceedings of the 1st annual CSAIL Student Workshop (CSW05)*, September 2005.

[50] Deborah L. McGuinness and Frank van Harmelen. OWL Web ontology language overview. Technical report, The World Wide Web Consortium (W3C), February 2004.

[51] Rada Mihalcea. Co-training and self-training for word sense disambiguation. In *The Conference on Natural Language Learning (coNLL2004)*, 2004.

[52] George Miller. Wordnet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–312, 1990.

[53] Michele Missikoff, Roberto Navigli, and Paola Velardi. The usable ontology: An environment for building and assessing a domain ontology. In *The proceedings of the first international Semantic Web conference (ISWC 2002)*, June 2002.

[54] Dan I. Moldovan, Roxana Girju, and Vasile Rus. Domain-specific knowledge acquisition from text. In *ANLP*, pages 268–275, 2000.

[55] Emmanuel Morin and Jacquemin Christian. Automatic acquisition and expansion of hypernym links. *Computers and the Humanities*, 38(4):363–396, 2004.

[56] Emmanuel Morin and Christian Jacquemin. Projecting corpus-based semantic links on a thesaurus. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL1999)*, 1999.

[57] Christoph Mueller, Stephan Rapp, and Michael Strube. Applying co-training to reference resolution. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL2002)*, 2002.

[58] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the North American Association for Computational Linguistics and the Human Language Technologies Conferences (NAACL/HLT2003)*, 2003.

[59] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Knowledge Discovery and Data Mining*, 2002.

[60] Patrick Pantel and Deepak Ravichandran. Automatically labeling semantic classes. In *Proceedings of the North American Association for Computational Linguistics and the Human Language Technologies Conferences (NAACL/HLT2004)*, 2004.

[61] Lawrence Philips. The double-metaphone search algorithm. *C/C++ User's Journal*, 18(6), June 2000.

[62] David Pierce and Claire Cardie. Limitations of co-training for natural language learning from large datasets. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP2001)*, 2001.

[63] Massimo Poesio and Abdulrahman Almuhareb. Identifying concept attributes using a classifier. In *The proceedings of the ACL SIGLEX workshop on deep lexical acquisition*, pages 18–27, June 2005.

[64] Mark Przybocki, Donna Harman, and George Doddington. Automatic content extraction (ACE) (http://www.itl.nist.gov/iad/89 4.01/tests/ace/), 2003.

[65] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479, 1999.

[66] Angus Roberts. Learning meronyms from biomedical text. In *The proceedings of the ACL Student Research Workshop (ACL2005/SRW)*, 2005.

[67] Barbara Rosario and Marti Hearst. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP2001)*, 2001.

[68] Rion Snow. Learning syntactic patterns for automatic hypernym discovery. In *The Conference on Neural Information Processing Systems (NIPS2004)*, 2004.

[69] Mark Stevenson and Mark A. Greenwood. A semantic approach to ie pattern induction. In *The proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL2005)*, 2005.

[70] Hakan Sundblad. Automatic acquisition of hyponyms and meronyms from question corpora. In *The proceedings of the Workshop on Natural Language Processing and Machine Learning for Ontology Engineering at ECAI2002*, 2002.

[71] Peter D. Turney. Mining the Web for synonyms: PMI–IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–??, 2001.

[72] Ellen Voorhees. Overview of the TREC 2003 question answering track., 2003.

[73] Ellen Voorhees. Overview of the TREC 2004 question answering track., 2004.

[74] Piek Vossen. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic, 1998.

[75] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 17th international conference on computational linguistics (COLING2000)*, 2002.

[76] Roman Yangarber, Winston Lin, and Ralph Grishman. Unsupervised learning of generalized names. In *Proceedings of the 19th international conference on computational linguistics (COLING2002)*, 2002.

[77] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting of the Association for Computational Linguistics (ACL1995)*, pages 189–196, 1995.

[78] Deniz Yuret. *Discovery of Linguistic Relations using Lexical Attraction*. PhD thesis, Massachusetts Institute of Technology, 1998.