# Indirect Anaphora Resolution as Semantic Path Search

**James Fan, Ken Barker, and Bruce Porter**
Dept. of Computer Sciences, University of Texas at Austin
Austin, Texas, USA
jfan@cs.utexas.edu, kbarker@cs.utexas.edu, porter@cs.utexas.edu

## ABSTRACT

Anaphora occur commonly in natural language text, and resolving them is essential for capturing the knowledge encoded in text. Indirect anaphora are especially challenging to resolve because the referring expression and the antecedent are related by unstated background knowledge. Such anaphora need to be resolved properly in order to automatically capture the knowledge expressed in natural language. Resolving indirect anaphora has been treated as a unique problem that requires special-purpose methods, and these methods have had limited success in precision and recall. In this study, we used a generic tool for finding semantic paths between two concepts to resolve these anaphora, and it achieved approximately twice the recall of the best previous system without loss of precision. A series of ablation study showed that the biggest increase in recall came from an abductive stopping criterion of the search.

## Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: [natural language processing]

## General Terms

Languages, Experimentation

## Keywords

anaphora, indirect anaphora, knowledge-based systems, ontology

## 1. INTRODUCTION

Indirect anaphora is a type of anaphora that requires background knowledge in order to identify the referent. It may account for 15% of noun phrase anaphora [20], making it an important type. Resolving indirect anaphora is a necessary step for automatically capturing knowledge from text. However, resolving indirect anaphora is problematic for shallow processing systems because it usually requires common sense knowledge. Because most knowledge sources contain little common sense knowledge, achieving a high level of recall in resolving indirect anaphora is especially difficult.

| Link Type | Example |
|---|---|
| Set/element | a ***class*** ... the *student* ... |
| Whole/part (metonymy) | a ***room*** ... the *wall* ... |
| Hypernym/hyponym | an ***oak*** ... the *tree* ... |
| Event/role | a ***murder*** ... the *killer* ... |
| Cause/consequence | an ***earthquake*** ... the *debris* ... |

**Table 1: Some frequent types of indirect anaphora.**

In this paper, we describe the results of resolving indirect anaphora using an interpreter for finding short semantic paths between concepts. However, we do not claim that this tool can find any type of associations between any two concepts, e.g. analogical associations are well outside its scope. In this study, we found that when applied to indirect anaphora resolution, our interpreter achieved a significant increase in recall (with no drop in precision) compared to previous studies on the same data set using the same knowledge base.

## 2. INDIRECT ANAPHORA

Indirect anaphora, also known as *bridging reference* or *associative anaphora*, arises "when a reference becomes part of the hearer's or reader's knowledge indirectly rather than by direct mention" [16]. The object that is being referred to is called the *anchor* or the *antecedent*, the expression that refers to the antecedent is called the *referring expression*, and the association between the referring expression and the anchor is called the *link*. For example, the following sentence contains an instance of indirect anaphora.

> When the detective got back to ***the garage***, *the door* was unlocked.

The referring expression, *the door*, relates to the antecedent, ***the garage***, through a whole/part (metonymy) link.

Unlike other types of anaphora, which can often be resolved using syntactic features, the resolution of indirect anaphora requires semantic knowledge of the relationship between the referring expression and the antecedent. Because such knowledge was previously unavailable to computer programs, most of the early studies in indirect anaphora were theoretical [5, 9]. These studies identified a variety of types of indirect anaphora (see table 1).

Recently there has been more progress in experimental studies of indirect anaphora. These studies can be divided into WordNet-based systems and machine-learning systems. The WordNet-based systems [26] use WordNet as the knowledge base. They take in a referring expression and a list of nouns that appear earlier than the referring expression in the same text. The systems choose one noun as the most likely antecedent for the referring expression. They select the antecedent by first grouping the nouns based on sentence boundary, then using stack-based theory [23] to sort the candidate associations and select the most promising one. Specifically, the systems look back one sentence at a time and return a candidate as the antecedent as soon as the candidate satisfies one of the following conditions based on WordNet knowledge:

- The candidate is a synonym of the referring expression, such as *aviator* and *flyer*.
- The candidate is a hypernym (superclass) or hyponym (subclass) of the referring expression, such as *oak* and *tree*.
- The candidate is a coordinate sibling of the referring expression, such as *home* and *house*.
- The candidate has a meronymic (has-part) or holonymic (is-part-of) relation with the referring expression, such as *room* and *wall*.

This approach not only returns the antecedent, but it can also reveal the type of association between each referring expression and its antecedent, which is a piece of information important for other parts of a full natural language processing system. However, many frequently used types of links, such as event/role or cause/consequence, cannot be discovered by these systems because WordNet does not contain such knowledge.

There have been many successful machine learning based-coreference resolution systems, such as [4, 25, 12], however most of them do not resolve indirect anaphora. The ones that do [15, 2] typically use the web as the corpus. Instead of searching through WordNet, they issue a series of web search queries made of the referring expression and each candidate antecedent. These systems use the number of web pages that contain both the referring expression and the candidate antecedent being queried as a measure of the strength of association. If the strength exceeds a threshold, then the systems consider the candidate the true antecedent of the referring expression. Machine learning techniques are used to

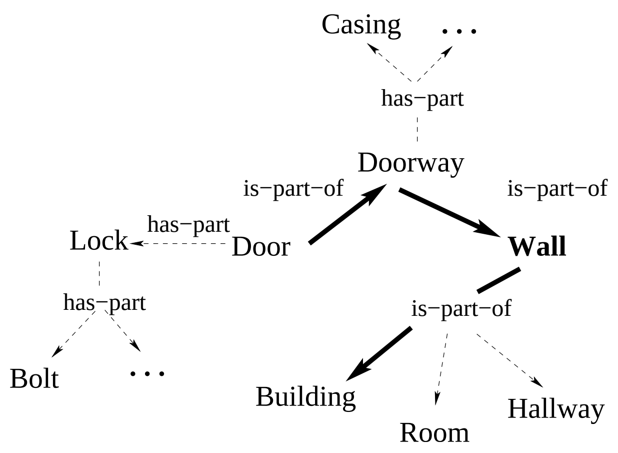


**Figure 1: An example of how the algorithm works. Given the nouns *garage* and *door*, the solid bold lines show the path found by breadth-first search starting at *door* and ending at *building*, which is a superclass of *garage*.**

determine the best threshold. The strength of this approach is that it provides a broad coverage of all types of links, and it has achieved precision and recall that are comparable to the WordNet-based systems. The weakness of this approach is that it does not determine the semantic nature of the relationship between the referring expression and the antecedent.

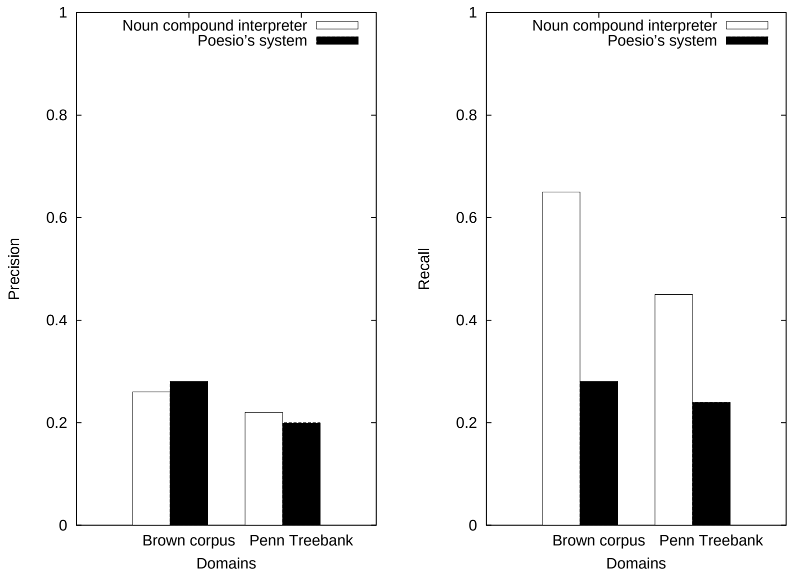
A more recent study of indirect anaphora has shown that precision for either approach can be significantly improved with a more sophisticated selection mechanism that learns by combining several features, such as salience (the contextual distance of a referring expression and its anchor) and lexical distance (the semantic distance between a referring expression and its anchor) [19]. Improving recall remains a challenge.

## 3. OUR SYSTEM

We approach the indirect anaphora resolution problem from a different perspective. Whereas previous researchers treated indirect anaphora resolution as a unique problem requiring special-purpose methods, we view it as an instance of a more general problem: how to use background knowledge to infer relations among linguistic constituents.

### 3.1 Our interpreter

Our interpreter considers a pair of concepts to semantically related if it can find an interpretation for the pair. The interpretation task can be viewed as follows: given a knowledge base encoded as a semantic network, and a pair of nouns corresponding to two nodes in the network, find a path of semantic relations between them. First, the two nouns are mapped to nodes in the knowledge base, $C_1$ and $C_2$, by a process that is outside of the scope of this paper. Then, two breadth-first searches of the knowledge base along all binary semantic relations are conducted. The first search starts from $C_1$ and looks for $C_2$ or any superclass or subclass of it. The



(a) Precision. (b) Recall.

**Figure 2: Performance comparison of the two indirect anaphora resolution systems on the two data sets. As shown in the data, our interpreter is as precise as Poesio's system, but it approximately doubles the recall.**

second search starts from $C_2$ and looks for $C_1$, or any superclass or subclass of it. Two searches are needed because the semantic relations are directed. If any interpretations are found, they are passed to a sorting function, which ranks interpretations by ascending path length with preference given to paths containing essential relations as determined by valency theory [24]. During the evaluation, only the top interpretation from the sorting function is returned as the interpretation of the input.

Figure 1 is an example of how the algorithm works. Given the nouns *garage* and *door*, the first search begins at Door, then traverses all the semantic relations from Door, such as *has-part* and *is-part-of*. The search stops at Building, which is a superclass of Garage. The solid bold lines indicate the path found and returned. It describes that the door "is part of a doorway, which is part of a wall, which is part of a building" (a superclass of Garage). The second search begins at Garage, and looks for any superclass or subclass of Door. The sorting function ranks the paths returned from the two searches, and chooses the shortest path to return.

If multiple valid interpretations from different candidate antecedents are found, the same sorting function selects the interpretation whose path length is the shortest.[1]

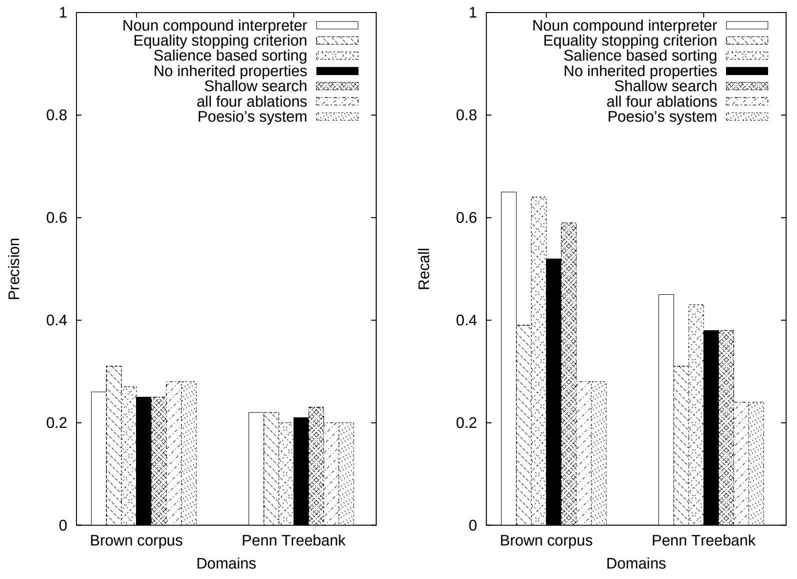
## 3.2 Comparison with previous studies

Our system is most similar to WordNet based-systems because they infer relations among candidate antecedents and referring expressions through a series of searches. However, there are some significant differences. First, unlike the WordNet-based systems [26], which only search direct properties of a class, our interpreter also searches inherited properties from the ancestors of a class. For example, in WordNet, the word *barrack* has a *squad room* part, and it inherits the part *foundation* from its ancestor *structure*. This difference originates from the fact that our interpreter was originally developed with a knowledge base of formal representations of actions, entities and modifiers. Such knowledge bases typically use subsumption to form a taxonomic graph, and they use inheritance to infer that properties of a general class apply to subclasses as well. In contrast, WordNet was developed as a lexical reference system, which does not use inheritance.

Second, our interpreter uses a more relaxed stopping criterion (*super_or_subclass*) which stops the search when a superclass or subclass of the goal is found. Previous systems [21, 27, 6, 11, 22, 18, 3] used a more restricted stopping criterion (*equality*), in which the search stops when a class identical to the goal is found. The more relaxed stopping criterion allows both inductive reasoning and abductive reasoning when deciding if the search should terminate. When a subclass of the goal is found, by inductive reasoning we infer there is a path between the starting concept and the goal concept; any instance of the subclass is an instance of the class itself. For example, given ***a barrack*** ... *the room* ..., the interpreter will start a search from Barrack and will stop the search at the Squad-Room part of the Barrack because a Squad-Room is a kind of Room, which is the goal concept.

When a superclass of the goal is found, by abductive reasoning we infer that there may be a path between the starting concept and the goal concept: an instance of the superclass may be an instance of the class itself. For example, given ***a***

[1]This is typically referred to as lexical distance.



(a) Precision. (b) Recall.

**Figure 3: Performance of our interpreter after a series of ablations. The effect of these ablations on precision is limited, but recall suffered. Of all the ablations, a more restricted stopping criterion has the biggest impact on recall. If all the ablations take place at the same time, then our interpreter behaves essentially like Poesio's system.**

***garage*** ... *the door* ..., the interpreter will conduct a search starting from Door as illustrated in figure 1. The search stops at Building, a superclass of the goal, Garage, because if a door is part of a building, which can be a garage, then the door **may** be part of a garage to which the expression, *garage*, refers.

In contrast, when *equality* is used in this example, the search will not stop at Building because it is not equal to Garage. Previous studies [14, 7] have shown that a combination of inductive and abductive reasoning is effective for finding semantic paths between two given concepts.

The third difference with WordNet-based algorithms is that our interpreter conducts deeper searches. The search depth limit of previous systems was set to one for computational efficiency. Our interpreter can use a deeper limit because its more relaxed stopping criterion stops most searches at a shallow depth. It can afford to have deeper searches occasionally.

Finally, our interpreter's sorting and selecting function depends on lexical distance rather than salience. Salience is not used because it utilizes the sequential order of sentences in a discourse; our interpreter is a generic solution to the semantic search problem, whose input is an unordered pair of concepts. If experiments show that salience is key to the interpreter's performance for indirect anaphora resolution, the interpreter can be tailored to indirect anaphora resolution problems by replacing the lexical distance based sorting function with a salience based sorting function.

## 4. EXPERIMENT 1

We first evaluated the performance of our interpreter applied to indirect anaphora by comparing it to our reimplementation of a WordNet-based system from previous studies [26]. Poesio's system was chosen for comparison for two reasons. First, it was one of the best WordNet-based systems. Second, part of the data sets used in its evaluation is available for easy comparison.

We used WordNet 2.0 as the knowledge base in our experiments for both systems. In order to apply our interpreter, we had to convert WordNet databases into our knowledge base representation. First, we mapped each WordNet noun or verb synset into a class concept. Then, we mapped WordNet hyponymy and meronymy relations into *subclass*, *has-part*, *element* and *material* relations accordingly (meronymy relation can be mapped to three relations: has-part, element or material). Because it has been shown that the taxonomy derived from WordNet in this way is not of particularly high quality [17] (sometimes a taxonomic sibling should actually be a descendant or an ancestor), we changed the stopping criterion of the interpreter to be *super_or_subclass_***or_sibling**.

We used two data sets for the evaluation. The goal of the evaluation is to measure the performance of the interpreter for indirect anaphora *resolution*, not *detection*. The first data set is a subset of the Brown corpus containing 32 articles on various topics. This data set was annotated for indirect anaphora and used in a previous study [2]. The annotation marks all referring expressions and their antecedents. We used six articles from the data set for debugging, and the rest for testing. The training data were used to set the threshold for the number of sentences in the window for candidate antecedents. We used a 6 sentence window for our experiment. In the data set, we excluded instances of direct anaphora to isolate the effects on indirect anaphora. We also excluded named entities because WordNet contains little knowledge about them, and recognizing named entities is not central to our task of indirect anaphora resolution. There are a total of 196 instances of indirect anaphora in the test set.

The second data set consists of 32 Wall Street Journal articles from the Penn Treebank I corpus containing 82 instances of indirect anaphora after removing all named entities and direct anaphora. It was used previously [26], and it was partially annotated. We completed the annotations for our experiments.

Because each word in the data sets may be mapped to multiple WordNet synsets, we needed a mechanism of word sense disambiguation. As a simple approach we took the cross product of all the possible word senses of each referring expression and each candidate antecedent to form pairs of synsets. This yields a set of candidate solutions, and the the sorting and selecting function chooses among them. In the previous example, the referring expression, *door*, has five senses, and one candidate antecedent, *garage*, has two senses. The cross product yields ten pairs of synsets (<*door#1, garage#1*>, <*door#1, garage#2*>, <*door#2, garage#1*>, ...). The interpreter then interprets the pairs, and the sorting and selecting function chooses the best among all the solutions found as the interpretation for the candidate, *garage*.

The metrics we used to evaluate performance are precision and recall, which are defined as follows [13]:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

For our experiment, $tp$ (true positive) is the number of indirect anaphora whose selected antecedents match the annotated ones, $fp$ (false positive) is the number of indirect anaphora whose selected antecedents do not match the annotated ones, and $fn$ (false negative) is the number of indirect anaphora for which the system cannot find an antecedent. Precision estimates the likelihood of a correct resolution when an antecedent is found; recall is a measurement of coverage.

## 5. RESULTS

Figure 2 shows the performance of our system and Poesio's system on the two test sets. The left bar is the performance of our interpreter applied to the task of resolving indirect anaphora, and the right bar is the performance of our implementation of Poesio's system. [2] Our interpreter achieved about the same precision as Poesio's system with approximately twice the recall.

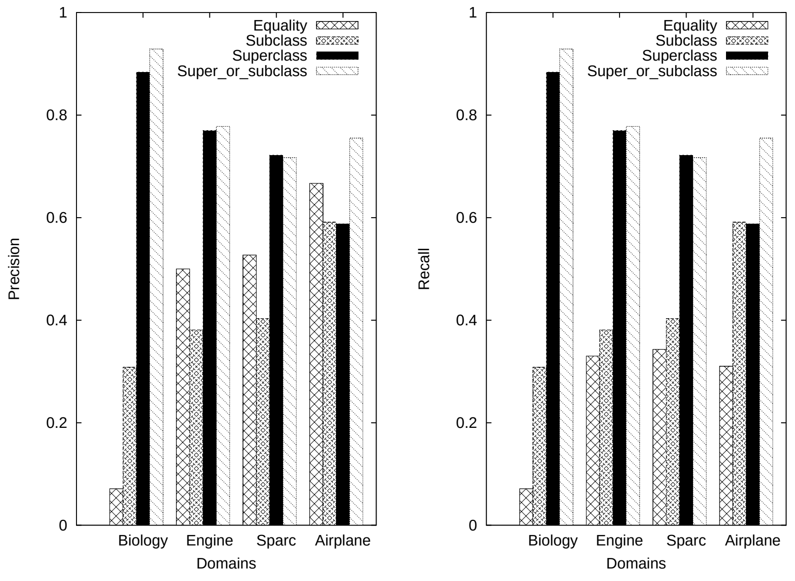
An analysis of the failed cases for both systems revealed that nearly half of them (45% for the Brown corpus data and 49% for the Penn Treebank data) are caused by insufficient knowledge. Because WordNet only provides formal encodings of mereological knowledge (hypernym, hyponym, meronym and holonym), several frequently used type of indirect anaphora, such as Event/role and Cause/consequence, cannot be resolved. For example, given the referring expression *the judge* and candidate antecedent *trial*, WordNet does not contain formally encoded knowledge that would associate "judge" with "trial". In order to improve the performance, other knowledge sources, such as FrameNet [8], the Component Library [1] or formal encodings of WordNet glosses [10], are needed.

### 5.1 Ablation study

Although the analysis of the failed cases reveals that additional knowledge would improve both systems, it does not explain why our system's recall is significantly better than that of Poesio's. As described in an earlier section, there are four main differences between our interpreter and Poesio's system (use of inherited properties, stopping criterion, search depth bounds and use of lexical distance instead of salience), and some combination of them must account for the improvement in recall. We analyzed their relative contributions through a series of ablations. Each ablation removes one difference while leaving other aspects of the original interpreter intact. We also removed all four differences at once. This version of the system is functionally identical to Poesio's system, and it should have identical performance as our reimplementation of Poesio's system.

Figure 3 shows the results of the ablation study on the two data sets. Replacing lexical distance-based sorting and selecting by salience did not have a large impact on precision or recall on the two data sets we tested. Shallow searches had a small influence on recall (7 percentage points on the Brown corpus data set and 7 percentage points on the Penn Treebank data set). We have experimented with various search depths, and we found that increasing the search depth beyond two has little effect on performance. As predicted, removing inherited properties lowers recall (13 percentage points on the Brown corpus data set and 7 percentage points on the Penn Treebank data set) with little effect on precision. Replacing the stopping criterion has the largest impact on recall. It lowered recall by 26 percentage points on the Brown corpus data set, and 14 percentage points on the Penn Treebank data set. In addition, as predicted, by removing all four differences, our interpreter performs just like Poesio's system.

[2]We believe our implementation is faithful because on average it performed within 4% of published results [26]. The difference is due to different annotations in the corpora.



(a) Precision. (b) Recall.

**Figure 4: Performance of the noun compound interpreter using four different stopping criteria on four different data sets. The four bars for each domain, from left to right, represent the interpreter's performance in that domain using *equality*, *subclass*, *superclass* and *super_or_subclass* stopping criterion respectively. The interpreter that uses the most common and restricted stopping criterion, *equality*, had the worst recall and second best precision. The interpreter that uses the least restricted stopping criterion *super_or_subclass* had the best precision and recall.**

## 6. EXPERIMENT 2

The ablation study showed that the biggest increase in recall comes from an abductive stopping criterion. One may wonder whether the effect is restricted to the data sets used in this study, so we evaluated the influence of stopping criteria on larger data sets and with a different application of semantic path search using the same tool.

### 6.1 Setup

We evaluated the impact of four different stopping criteria on semantic path search in this study. In addition to *super_or_subclass* and *equality*, we used *superclass* and *subclass*. These are less restrictive than *equality* but more restrictive than *super_or_subclass*. The *superclass* criterion stops the search when the system visits a class whose superclass has already been visited. The *subclass* criterion stops the process when the system visits a class whose subclass has already been visited.

When a more restrictive stopping criterion is used, false positives are expected to decrease because fewer answers will be found by the system. Consequently precision should increase. When a less restrictive stopping criterion is used, false negatives are expected to decrease because fewer inputs will have no interpretation, and recall should increase.

We evaluated the impact of different stopping criteria on the task of noun compound interpretation. Here, we define a noun compound to be a pair of nouns composed of a head noun and a modifier. The head noun determines the type of the whole compound, and the modifier specializes the type from the head noun. Longer sequences of nouns can be bracketed into pairs of nouns (with few exceptions). To interpret a noun compound is to find the semantic relation between the head noun and its modifier.

It is worth noting that this task has a slightly different emphasis than indirect anaphora. For indirect anaphora resolution systems, the goal is to see **if** a candidate antecedent is related to the referring expression. For noun compound interpretation systems, the head noun is assumed to be related to the modifier for each input, and the goal is to determine **how** they are related.

### 6.2 Test data

We used four sets of data drawn from different domains to avoid getting results that are skewed to a particular domain, knowledge base or data set. The first three data sets consist of a total of 742 pairs of nouns extracted from three different domains (a biology text book, a small engine repair manual, and a Sparcstation owners manual). Noun compounds are closely related to indirect anaphora, and they can be converted into pairs of antecedent and referring expressions. For example, the noun compound *engine manufacturer* can be converted to an indirect anaphora instance "***... an engine*** *... the manufacturer ...*". The knowledge bases we used to interpret noun compounds are based on the Component Library [1], an upper ontology composed of a set of general concepts (entities, events and roles) and their associated axioms. For each domain, we extended the Component Library in the following ways. First, the biology knowledge base was extended to have an 18-level deep taxonomy containing 1937 concepts. On average, each concept is directly connected to 9 other concepts. Then, the knowledge bases for the other two data sets (the small engine repair manual and the Sparcstation manual) were augmented with WordNet knowledge plus about ten concepts that are important to each of the two domains (whose partonomies are not complete in WordNet). The engine knowledge base contains an 18-level deep taxonomy and 2159 concepts. On average each concept is directly connected to 11 other concepts. The Sparcstation knowledge base contains a 16-level deep taxonomy and 2105 concepts. Each concept is directly connected to 10 other concepts on average.

The fourth data set consists of 55 noun compounds in the domain of airplane parts. These data are encodings of airplane-related sentences found in various online sources. The knowledge base used to interpret the airplane data is a custom-built knowledge base about airplanes. The taxonomy is 15 levels deep, and it has more than 8,000 concepts. Each concept is directly connected to 70 other concepts on average.

## 7. RESULTS

Figure 4 shows the impact of different stopping criteria. Similar to the previous experiment, recall is improved significantly when *super_or_subclass* is used. The interpreter that used the most restricted stopping criterion, *equality*, always had the worst recall in all four data sets. The interpreter that used the least restricted stopping criterion, *super_or_subclass*, always had the best recall. In addition to the increased recall, the *super_or_subclass* based interpreter also had the best precision.

The results appear counterintuitive at first because the least restrictive stopping criterion produces in unsound reasoning and it may cause search to stop prematurely. Such stoppage may induce many false positives and reduce the precision significantly. In practice we found this was rarely the case. In fact, most of the interpretations found by the *super_or_subclass* stopping criterion are true positives, not false positives, so precision was not compromised. In addition, because *super_or_subclass* is a more relaxed stopping criterion, it can find more interpretations and it will have lower false negatives and better recall. This explains the performance difference between various stopping criteria. The precision improvement does not extend to the indirect-anaphora task because, unlike noun compounds, a candidate antecedent may be completely unrelated to the referring expression, and a relaxed stopping criterion may find interpretations when there are none, causing false positives to increase and precision to reduce.

## 8. DISCUSSION AND SUMMARY

In this paper, we studied the application of a general tool for finding semantic paths between concepts to the task of indirect anaphora resolution. Compared with a state of the art system designed specifically for this task, our system achieved twice the recall with no drop in precision. An analysis of the failed cases suggested that more knowledge sources could further improve the performance. Further, an ablation study revealed that the biggest increase in recall comes from a more relaxed stopping criterion, and an experiment conducted on four noun compound data sets showed that the most relaxed stopping criterion had the best performance.

The results have some important implications. First, they suggest that a single method that searches a knowledge base for semantic paths between pairs of concepts can achieve good results on two different linguistic tasks: resolving indirect anaphora and interpreting noun compounds. In future work, we will assess its effectiveness on other natural language processing tasks. Second, although the *super_or_subclass* stopping criterion is not knowledge base dependent or domain dependent, it may be affected by the taxonomy it uses. It will be interesting to see if the interpreter's performance can be further enhanced if methods such as OntoClean [17] are used to realign the WordNet taxonomy. Third, because *super_or_subclass* is a relaxed stopping criterion, it causes search to stop early. This makes spreading activation feasible even on very large knowledge bases, such as the searches required by many Semantic WEB tasks.

## 9. ACKNOWLEDGMENTS

Support for this research was provided by SRI, International as part of DARPA's PAL project.

This research was supported in part by the National Science Foundation under CISE Research Infrastructure Grant EIA-0303609.

## 10. REFERENCES

[1] K. Barker, B. Porter, and P. Clark. A library of generic concepts for composing knowledge bases. In *Proceedings of First International Conference on Knowledge Capture*, 2001.

[2] R. Bunescu. Associative anaphora: a web-based approach. In *Proceedings of the EACL Workshop on the Computational Treatment of Anaphora*, 2003.

[3] R. D. Burke, K. J. Hammond, V. Kulyukin, N. T. Steven L. Lytinen, and S. Schoenberg. Question answering from frequently asked question files: experiences with the FAQ FINDER system. *AI Magazine*, 18(2):57–65, 1997.

[4] C. Cardie and K. Wagstaff. Noun phrase coreference as clustering. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Scale Corpora*, 1999.

[5] H. H. Clark. Bridging. In *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*, pages 167–174. Association for Computational Linguistics, 1975.

[6] P. Cohen and R. Kjeldsen. Information retrieval by constrained spread activation on semantic networks. *Information processing & management*, 23(4):255–268, 1987.

[7] J. Fan, K. Barker, and B. Porter. The knowledge required to interpret noun compounds. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence*, 2003.

[8] C. J. Fillmore. FrameNet, 2004. http://www.icsi.berkeley.edu/ framenet/.

[9] C. Gardent, H. Manuelian, and E. Kow. Which bridges for bridging definite descriptions? In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*, 2003.

[10] S. Harabagiu, G. A. Miller, and D. I. Moldovan. WordNet 2 – a morphologically and semantically enhanced resource. In *Proceedings of ACL-SIGLEX99: Standardizing Lexical Resources*, 1999.

[11] R. Kjeldsen and P. Cohen. The evolution and performance of the GRANT system. *IEEE Expert*, 2(2):73–79, 1987.

[12] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. A mention-synchronous coreference resolution algorithm based on bell tree. In *Proceedings of ACL 2004*, 2004.

[13] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, Boston, 1999.

[14] K. Markert and U. Hahn. On the interaction of metonymies and anaphora. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 1997.

[15] K. Markert, M. Nissim, and N. Modjeska. Using the web for nominal anaphora resolution. In *Proceedings of the EACL Workshop on the Computational Treatment of Anaphora*, pages 39–46, 2003.

[16] R. Mitkov. *Anaphora Resolution*. Longman, London, 2002.

[17] A. Oltramari, A. Gangemi, N. Guarino, and C. Masolo. Restructuring WordNet's top-level: The OntoClean approach. In *Proceedings of LREC2002 (OntoLex workshop)*, 2002.

[18] B. Onyshkevych and S. Nirenburg. A lexicon for knowledge-based mt. *Machine Translation*, 10(1-2):5–57, 1995.

[19] M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. Learning to resolve bridging references. In *Proceedings of ACL 2004*, 2004.

[20] M. Poesio and R. Vieira. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216, 1998.

[21] S. E. Preece. *A spreading activation model for information retrieval*. PhD thesis, University of Illinois, Urbana-Champaign, USA, 1981.

[22] L. Rau. Knowledge organization and access in a conceptual information system. *Information Processing & Management*, 23(4):269–283, 1987.

[23] C. L. Sidner. *Towards a computational theory of definite anaphora comprehension in English discourse*. PhD thesis, MIT, Cambridge, MA, 1979.

[24] H. L. Somers. *Valency and case in computational linguistics*. Edinburgh University Press, 22 George Square, Edinburgh, 1987.

[25] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[26] R. Vieira and M. Poesio. An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593, 2000.

[27] D. L. Waltz and J. B. Pollack. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9:51–74, 1985.