Bootstrapping Path-Based Pronoun Resolution

Shane Bergsma
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
bergsma@cs.ualberta.ca

Dekang Lin
Google, Inc.
1600 Amphitheatre Parkway,
Mountain View, California, 94301
lindek@google.com

Abstract

We present an approach to pronoun resolution based on syntactic paths. Through a simple bootstrapping procedure, we learn the likelihood of coreference between a pronoun and a candidate noun based on the path in the parse tree between the two entities. This path information enables us to handle previously challenging resolution instances, and also robustly addresses traditional syntactic coreference constraints. Highly coreferent paths also allow mining of precise probabilistic gender/number information. We combine statistical knowledge with well known features in a Support Vector Machine pronoun resolution classifier. Significant gains in performance are observed on several datasets.

1 Introduction

Pronoun resolution is a difficult but vital part of the overall coreference resolution task. In each of the following sentences, a pronoun resolution system must determine what the pronoun *his* refers to:

(1) John needs *his* friend.

(2) John needs *his* support.

In (1), *John* and *his* corefer. In (2), *his* refers to some other, perhaps previously evoked entity. Traditional pronoun resolution systems are not designed to distinguish between these cases. They lack the specific world knowledge required in the second instance – the knowledge that a person does not usually explicitly need his own support.

We collect statistical path-coreference information from a large, automatically-parsed corpus to address this limitation. A *dependency path* is defined as the sequence of dependency links between two potentially coreferent entities in a parse tree. A path does not include the terminal entities; for example, "John needs his support" and "He needs their support" have the same syntactic path. Our algorithm determines that the dependency path linking the *Noun* and *pronoun* is very likely to connect coreferent entities for the path "*Noun* needs *pronoun*'s friend," while it is rarely coreferent for the path "*Noun* needs *pronoun*'s support."

This likelihood can be learned by simply counting how often we see a given path in text with an initial *Noun* and a final *pronoun* that are from the same/different gender/number classes. Cases such as "John needs her support" or "They need his support" are much more frequent in text than cases where the subject noun and pronoun terminals agree in gender/number. When there is agreement, the terminal nouns are likely to be coreferent. When they disagree, they refer to different entities. After a sufficient number of occurrences of agreement or disagreement, there is a strong statistical indication of whether the path is *coreferent* (terminal nouns tend to refer to the same entity) or *non-coreferent* (nouns refer to different entities).

We show that including path coreference information enables significant performance gains on three third-person pronoun resolution experiments. We also show that coreferent paths can provide the seed information for bootstrapping other, even more important information, such as the gender/number of noun phrases.

2 Related Work

Coreference resolution is generally conducted as a pairwise classification task, using various constraints and preferences to determine whether two

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 33–40,
Sydney, July 2006. ©2006 Association for Computational Linguistics

expressions corefer. Coreference is typically only allowed between nouns matching in gender and number, and not violating any intrasentential syntactic principles. Constraints can be applied as a preprocessing step to scoring candidates based on distance, grammatical role, etc., with scores developed either manually (Lappin and Leass, 1994), or through a machine-learning algorithm (Kehler et al., 2004). Constraints and preferences have also been applied together as decision nodes on a decision tree (Aone and Bennett, 1995).

When previous resolution systems handle cases like (1) and (2), where no disagreement or syntactic violation occurs, coreference is therefore determined by the weighting of features or learned decisions of the resolution classifier. Without path coreference knowledge, a resolution process would resolve the pronouns in (1) and (2) the same way. Indeed, coreference resolution research has focused on the importance of the *strategy* for combining well known constraints and preferences (Mitkov, 1997; Ng and Cardie, 2002), devoting little attention to the development of new features for these difficult cases. The application of *world knowledge* to pronoun resolution has been limited to the semantic compatibility between a candidate noun and the pronoun's context (Yang et al., 2005). We show semantic compatibility can be effectively combined with path coreference information in our experiments below.

Our method for determining path coreference is similar to an algorithm for discovering paraphrases in text (Lin and Pantel, 2001). In that work, the beginning and end nodes in the paths are collected, and two paths are said to be similar (and thus likely paraphrases of each other) if they have similar terminals (i.e. the paths occur with a similar *distribution*). Our work does not need to store the terminals themselves, only whether they are from the same pronoun group. Different paths are not compared in any way; each path is individually assigned a coreference likelihood.

3 Path Coreference

We define a *dependency path* as the sequence of nodes and dependency labels between two potentially coreferent entities in a dependency parse tree. We use the structure induced by the minimalist parser Minipar (Lin, 1998) on sentences from the news corpus described in Section 4. Figure 1 gives the parse tree of (2). As a short-form, we write the dependency path in this case as "*Noun* needs *pronoun*'s support." The path itself does not include the terminal nouns "John" and "his."

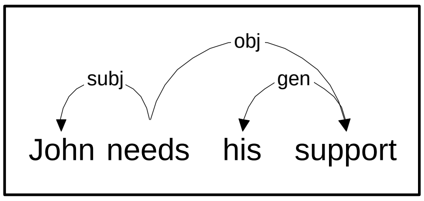

Figure 1: Example dependency tree.

Our algorithm finds the likelihood of coreference along dependency paths by counting the number of times they occur with terminals that are either *likely coreferent* or *non-coreferent*. In the simplest version, we count paths with terminals that are both pronouns. We partition pronouns into seven groups of matching gender, number, and person; for example, the first person singular group contains *I*, *me*, *my*, *mine*, and *myself*. If the two terminal pronouns are from the same group, coreference along the path is likely. If they are from different groups, like *I* and *his*, then they are non-coreferent. Let $N_S(p)$ be the number of times the two terminal pronouns of a path, p, are from the same pronoun group, and let $N_D(p)$ be the number of times they are from different groups. We define the *coreference* of p as:

$$C(p) = \frac{N_S(p)}{N_S(p) + N_D(p)}$$

Our statistics indicate the example path, "*Noun* needs *pronoun*'s support," has a low $C(p)$ value. We could use this fact to prevent us from resolving "his" to "John" when "John needs his support" is presented to a pronoun resolution system.

To mitigate data sparsity, we represent the path with the root form of the verbs and nouns. Also, we use Minipar's named-entity recognition to replace named-entity nouns by the semantic category of their named-entity, when available. All modifiers not on the direct path, such as adjectives, determiners and adverbs, are not considered. We limit the maximum path length to eight nodes.

Tables 1 and 2 give examples of coreferent and non-coreferent paths learned by our algorithm and identified in our test sets. *Coreferent* paths are defined as paths with a $C(p)$ value (and overall number of occurrences) above a certain threshold, indicating the terminal entities are highly likely

Table 1: Example coreferent paths: *Italicized* entities generally corefer.

	Pattern	Example
1.	*Noun* left ... to *pronoun*'s wife	*Buffett* will leave the stock to *his* wife.
2.	*Noun* says *pronoun* intends...	*The newspaper* says *it* intends to file a lawsuit.
3.	*Noun* was punished for *pronoun*'s crime.	*The criminal* was punished for *his* crime.
4.	... left *Noun* to fend for *pronoun-self*	They left *Jane* to fend for *herself*.
5.	*Noun* lost *pronoun*'s job.	*Dick* lost *his* job.
6.	... created *Noun* and populated *pronoun*.	Nzame created *the earth* and populated *it*
7.	*Noun* consolidated *pronoun*'s power.	*The revolutionaries* consolidated *their* power.
8.	*Noun* suffered ... in *pronoun*'s knee ligament.	*The leopard* suffered pain in *its* knee ligament.

to corefer. *Non-coreferent* paths have a $C(p)$ below a certain cutoff; the terminals are highly unlikely to corefer. Especially note the challenge of resolving most of the examples in Table 2 without path coreference information. Although these paths encompass some cases previously covered by Binding Theory (e.g. "Mary suspended her," *her* cannot refer to *Mary* by Principle B (Haegeman, 1994)), most have no syntactic justification for non-coreference *per se*. Likewise, although Binding Theory (Principle A) could identify the reflexive pronominal relationship of Example 4 in Table 1, most cases cannot be resolved through syntax alone. Our analysis shows that successfully handling cases that may have been handled with Binding Theory constitutes only a small portion of the total performance gain using path coreference.

In any case, Binding Theory remains a challenge with a noisy parser. Consider: "Alex gave her money." Minipar parses *her* as a possessive, when it is more likely an object, "Alex gave money *to her*." Without a correct parse, we cannot rule out the link between *her* and *Alex* through Binding Theory. Our algorithm, however, learns that the path "*Noun* gave *pronoun*'s money," is non-coreferent. In a sense, it corrects for parser errors by learning when coreference should be blocked, given *any* consistent parse of the sentence.

We obtain path coreference for millions of paths from our parsed news corpus (Section 4). While Tables 1 and 2 give test set examples, many other interesting paths are obtained. We learn coreference is unlikely between the nouns in "Bob married his mother," or "Sue wrote her obituary." The fact you don't marry your own mother or write your own obituary is perhaps obvious, but this is the first time this kind of knowledge has been made available computationally. Naturally, exceptions to the coreference or non-coreference of some of these paths can be found; our patterns represent general trends only. And, as mentioned above, reliable path coreference is somewhat dependent on consistent parsing.

Paths connecting pronouns to pronouns are different than paths connecting both nouns and pronouns to pronouns – the case we are ultimately interested in resolving. Consider "Company A gave its data on its website." The pronoun-pronoun path coreference algorithm described above would learn the terminals in "*Noun*'s data on *pronoun*'s website" are often coreferent. But if we see the phrase "Company A gave Company B's data on its website," then "its" is not likely to refer to "Company B," even though we identified this as a coreferent path! We address this problem with a two-stage extraction procedure. We first bootstrap gender/number information using the pronoun-pronoun paths as described in Section 4.1. We then use this gender/number information to count paths where an initial *noun* (with probabilistically-assigned gender/number) and following pronoun are connected by the dependency path, recording the agreement or disagreement of their gender/number category.[1] These superior paths are then used to re-bootstrap our final gender/number information used in the evaluation (Section 6).

We also bootstrap paths where the nodes in the path are replaced by their grammatical category. This allows us to learn general syntactic constraints not dependent on the surface forms of the words (including, but not limited to, the Binding Theory principles). A separate set of these non-coreferent paths is also used as a feature in our sys-

[1] As desired, this modification allows the first example to provide two instances of noun-pronoun paths with terminals from the same gender/number group, linking each "its" to the subject noun "Company A", rather than to each other.

Table 2: Example non-coreferent paths: *Italicized* entities do *not* generally corefer

	Pattern	Example
1.	*Noun* thanked ... for *pronoun*'s assistance	*John* thanked him for *his* assistance.
2.	*Noun* wanted *pronoun* to lie.	*The president* wanted *her* to lie.
3.	... *Noun* into *pronoun*'s pool	Max put the *floaties* into *their* pool.
4.	... use *Noun* to *pronoun*'s advantage	The company used *the delay* to *its* advantage.
5.	*Noun* suspended *pronoun*	*Mary* suspended *her*.
6.	*Noun* was *pronoun*'s relative.	*The Smiths* were *their* relatives.
7.	*Noun* met *pronoun*'s demands	*The players' association* met *its* demands.
8.	... put *Noun* at the top of *pronoun*'s list.	The government put *safety* at the top of *its* list.

tem. We also tried expanding our coverage by using paths *similar* to paths with known path coreference (based on distributionally similar words), but this did not generally increase performance.

4 Bootstrapping in Pronoun Resolution

Our determination of path coreference can be considered a bootstrapping procedure. Furthermore, the coreferent paths themselves can serve as the seed for bootstrapping additional coreference information. In this section, we sketch previous approaches to bootstrapping in coreference resolution and explain our new ideas.

Coreference bootstrapping works by assuming resolutions in unlabelled text, acquiring information from the putative resolutions, and then making inferences from the aggregate statistical data. For example, we assumed two pronouns from the same pronoun group were coreferent, and deduced path coreference from the accumulated counts.

The potential of the bootstrapping approach can best be appreciated by imagining millions of documents with coreference annotations. With such a set, we could extract fine-grained features, perhaps tied to individual words or paths. For example, we could estimate the likelihood each noun belongs to a particular gender/number class by the proportion of times this noun was labelled as the antecedent for a pronoun of this particular gender/number.

Since no such corpus exists, researchers have used coarser features learned from smaller sets through supervised learning (Soon et al., 2001; Ng and Cardie, 2002), manually-defined coreference patterns to mine specific kinds of data (Bean and Riloff, 2004; Bergsma, 2005), or accepted the noise inherent in unsupervised schemes (Ge et al., 1998; Cherry and Bergsma, 2005).

We address the drawbacks of these approaches by using coreferent paths as the assumed resolutions in the bootstrapping. Because we can vary the threshold for defining a coreferent path, we can trade-off coverage for precision. We now outline two potential uses of bootstrapping with coreferent paths: learning gender/number information (Section 4.1) and augmenting a semantic compatibility model (Section 4.2). We bootstrap this data on our automatically-parsed news corpus. The corpus comprises 85 GB of news articles taken from the world wide web over a 1-year period.

Table 3: Gender classification performance (%)

Classifier	F-Score
Bergsma (2005) Corpus-based	85.4
Bergsma (2005) Web-based	90.4
Bergsma (2005) Combined	92.2
Duplicated Corpus-based	88.0
Coreferent Path-based	90.3

4.1 Probabilistic Gender/Number

Bergsma (2005) learns noun gender (and number) from two principal sources: 1) mining it from manually-defined lexico-syntactic patterns in parsed corpora, and 2) acquiring it on the fly by counting the number of pages returned for various gender-indicating patterns by the Google search engine. The web-based approach outperformed the corpus-based approach, while a system that combined the two sets of information resulted in the highest performance (Table 3). The combined gender-classifying system is a machine-learned classifier with 20 features.

The time delay of using an Internet search engine within a large-scale anaphora resolution effort is currently impractical. Thus we attempted

Table 4: Example gender/number probability (%)

Word	*masc*	*fem*	*neut*	*plur*
company	0.6	0.1	98.1	1.2
condoleeza rice	4.0	92.7	0.0	3.2
pat	58.3	30.6	6.2	4.9
president	94.1	3.0	1.5	1.4
wife	9.9	83.3	0.8	6.1

to duplicate Bergsma's corpus-based extraction of gender and number, where the information can be stored in advance in a table, but using a much larger data set. Bergsma ran his extraction on roughly 6 GB of text; we used roughly 85 GB.

Using the test set from Bergsma (2005), we were only able to boost performance from an F-Score of 85.4% to one of 88.0% (Table 3). This result led us to re-examine the high performance of Bergsma's web-based approach. We realized that the corpus-based and web-based approaches are not exactly symmetric. The corpus-based approaches, for example, would not pick out gender from a pattern such as "John and his friends..." because "*Noun* and *pronoun*'s NP" is not one of the manually-defined gender extraction patterns. The web-based approach, however, would catch this instance with the "John * his/her/its/their" template, where "*" is the Google wild-card operator. Clearly, there are patterns useful for capturing gender and number information beyond the pre-defined set used in the corpus-based extraction.

We thus decided to capture gender/number information from coreferent paths. If a noun is connected to a pronoun of a particular gender along a coreferent path, we count this as an instance of that noun being that gender. In the end, the probability that the noun is a particular gender is the proportion of times it was connected to a pronoun of that gender along a coreferent path. Gender information becomes a single intuitive, accessible feature (i.e. the probability of the noun being that gender) rather than Bergsma's 20-dimensional feature vector requiring search-engine queries to instantiate.

We acquire gender and number data for over 3 million nouns. We use add-one smoothing for data sparsity. Some example gender/number probabilities are given in Table 4 (cf. (Ge et al., 1998; Cherry and Bergsma, 2005)). We get a performance of 90.3% (Table 3), again meeting our requirements of high performance and allowing for a fast, practical implementation. This is lower than Bergsma's top score of 92.2% (Figure 3), but again, Bergsma's top system relies on Google search queries for each new word, while ours are all pre-stored in a table for fast access.

We are pleased to be able to share our gender and number data with the NLP community.[2] In Section 6, we show the benefit of this data as a probabilistic feature in our pronoun resolution system. Probabilistic data is useful because it allows us to rapidly prototype resolution systems without incurring the overhead of large-scale lexical databases such as WordNet (Miller et al., 1990).

4.2 Semantic Compatibility

Researchers since Dagan and Itai (1990) have variously argued for and against the utility of collocation statistics between nouns and parents for improving the performance of pronoun resolution. For example, can the verb parent of a pronoun be used to select antecedents that satisfy the verb's selectional restrictions? If the verb phrase was *shatter it*, we would expect *it* to refer to some kind of brittle entity. Like path coreference, semantic compatibility can be considered a form of *world knowledge* needed for more challenging pronoun resolution instances.

We encode the semantic compatibility between a noun and its parse tree parent (and grammatical relationship with the parent) using mutual information (MI) (Church and Hanks, 1989). Suppose we are determining whether *ham* is a suitable antecedent for the pronoun *it* in *eat it*. We calculate the MI as:

$$\mathrm{MI}(\mathrm{eat{:}obj}, \mathrm{ham}) = \log \frac{\Pr(\mathrm{eat{:}obj{:}ham})}{\Pr(\mathrm{eat{:}obj})\Pr(\mathrm{ham})}$$

Although semantic compatibility is usually only computed for possessive-noun, subject-verb, and verb-object relationships, we include 121 different kinds of syntactic relationships as parsed in our news corpus.[3] We collected 4.88 billion *parent:rel:node* triples, including over 327 million possessive-noun values, 1.29 billion subject-verb and 877 million verb-direct object. We use small probability values for unseen Pr(*parent:rel:node*), Pr(*parent:rel*), and Pr(*node*) cases, as well as a default MI when no relationship is parsed, roughly optimized for performance on the training set. We

[2] Available at http://www.cs.ualberta.ca/~bergsma/Gender/

[3] We convert prepositions to relationships to enhance our model's semantics, e.g. *Joan:of:Arc* rather than *Joan:prep:of*

include both the MI between the noun and the pronoun's parent as well as the MI between the pronoun and the noun's parent as features in our pronoun resolution classifier.

Kehler et al. (2004) saw no apparent gain from using semantic compatibility information, while Yang et al. (2005) saw about a 3% improvement with compatibility data acquired by searching on the world wide web. Section 6 analyzes the contribution of MI to our system.

Bean and Riloff (2004) used bootstrapping to extend their semantic compatibility model, which they called contextual-role knowledge, by identifying certain cases of easily-resolved anaphors and antecedents. They give the example "Mr. Bush disclosed the policy by reading it." Once we identify that *it* and *policy* are coreferent, we include *read:obj:policy* as part of the compatibility model.

Rather than using manually-defined heuristics to bootstrap additional semantic compatibility information, we wanted to enhance our MI statistics automatically with coreferent paths. Consider the phrase, "Saddam's wife got a Jordanian lawyer for her husband." It is unlikely we would see "wife's husband" in text; in other words, we would not know that *husband:gen:wife* is, in fact, semantically compatible and thereby we would discourage selection of "wife" as the antecedent at resolution time. However, because "*Noun* gets ... for *pronoun*'s husband" is a coreferent path, we could capture the above relationship by adding a *parent:rel:node* for every pronoun connected to a noun phrase along a coreferent path in text.

We developed context models with and without these path enhancements, but ultimately we could find no subset of coreferent paths that improve the semantic compatibility's contribution to training set accuracy. A mutual information model trained on 85 GB of text is fairly robust on its own, and any kind of bootstrapped extension seems to cause more damage by increased noise than can be compensated by increased coverage. Although we like knowing audiences have noses, e.g. "the audience turned up its nose at the performance," such phrases are apparently quite rare in actual test sets.

5 Experimental Design

The noun-pronoun path coreference can be used directly as a feature in a pronoun resolution system. However, path coreference is undefined for cases where there is no path between the pronoun and the candidate noun – for example, when the candidate is in the previous sentence. Therefore, rather than using path coreference directly, we have features that are true if $C(p)$ is above or below certain thresholds. The features are thus set when coreference between the pronoun and candidate noun is likely (a coreferent path) or unlikely (a non-coreferent path).

We now evaluate the utility of path coreference within a state-of-the-art machine-learned resolution system for third-person pronouns with nominal antecedents. A standard set of features is used along with the bootstrapped gender/number, semantic compatibility, and path coreference information. We refer to these features as our "probabilistic features" (Prob. Features) and run experiments using the full system trained and tested with each absent, in turn (Table 5). We have 29 features in total, including measures of candidate distance, frequency, grammatical role, and different kinds of parallelism between the pronoun and the candidate noun. Several reliable features are used as hard constraints, removing candidates before consideration by the scoring algorithm.

All of the parsing, noun-phrase identification, and named-entity recognition are done automatically with Minipar. Candidate antecedents are considered in the current and previous sentence only. We use SVM^{light} (Joachims, 1999) to learn a linear-kernel classifier on pairwise examples in the training set. When resolving pronouns, we select the candidate with the farthest positive distance from the SVM classification hyperplane.

Our training set is the anaphora-annotated portion of the American National Corpus (ANC) used in Bergsma (2005), containing 1270 anaphoric pronouns[4]. We test on the ANC Test set (1291 instances) also used in Bergsma (2005) (highest resolution accuracy reported: 73.3%), the anaphora-labelled portion of AQUAINT used in Cherry and Bergsma (2005) (1078 instances, highest accuracy: 71.4%), and the anaphoric pronoun subset of the MUC7 (1997) coreference evaluation formal test set (169 instances, highest *precision* of 62.1 reported on all pronouns in (Ng and Cardie, 2002)). These particular corpora were chosen so we could test our approach using the same data as comparable machine-learned systems exploiting probabilistic information sources. Parameters

[4] See http://www.cs.ualberta.ca/~bergsma/CorefTags/ for instructions on acquiring annotations

Table 5: Resolution accuracy (%)

Dataset		ANC	AQT	MUC
1	Previous noun	36.7	34.5	30.8
2	No Prob. Features	58.1	60.9	49.7
3	No Prob. Gender	65.8	71.0	68.6
4	No MI	71.3	73.5	69.2
5	No $C(p)$	72.3	73.7	69.8
6	Full System	73.9	75.0	71.6
7	Upper Bound	93.2	92.3	91.1

were set using cross-validation on the training set; test sets were used only once to obtain the final performance values.

Evaluation Metric: We report results in terms of accuracy: Of all the anaphoric pronouns in the test set, the proportion we resolve correctly.

6 Results and Discussion

We compare the accuracy of various configurations of our system on the ANC, AQT and MUC datasets (Table 5). We include the score from picking the noun immediately preceding the pronoun (after our hard filters are applied). Due to the hard filters and limited search window, it is not possible for our system to resolve every noun to a correct antecedent. We thus provide the performance upper bound (i.e. the proportion of cases with a correct answer in the filtered candidate list). On ANC and AQT, each of the probabilistic features results in a statistically significant gain in performance over a model trained and tested with that feature absent.[5] On the smaller MUC set, none of the differences in 3-6 are statistically significant, however, the relative contribution of the various features remains reassuringly constant.

Aside from missing antecedents due to the hard filters, the main sources of error include inaccurate statistical data and a classifier bias toward preceding *pronouns* of the same gender/number. It would be interesting to see whether performance could be improved by adding WordNet and web-mined features. Path coreference itself could conceivably be determined with a search engine.

Gender is our most powerful probabilistic feature. In fact, inspecting our system's decisions, gender often rules out coreference regardless of path coreference. This is not surprising, since we based the acquisition of $C(p)$ on gender. That is, our bootstrapping assumption was that the majority of times these paths occur, gender indicates coreference or lack thereof. Thus when they occur in our test sets, gender should often sufficiently indicate coreference. Improving the orthogonality of our features remains a future challenge.

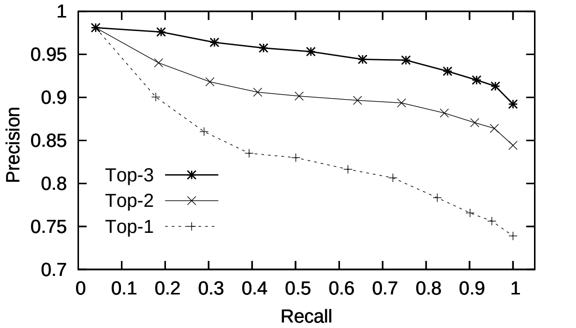

Figure 2: ANC pronoun resolution accuracy for varying SVM-thresholds.

Nevertheless, note the decrease in performance on each of the datasets when $C(p)$ is excluded (#5). This is compelling evidence that path coreference is valuable in its own right, beyond its ability to bootstrap extensive and reliable gender data.

Finally, we can add ourselves to the camp of people claiming semantic compatibility is useful for pronoun resolution. Both the MI from the pronoun in the antecedent's context and vice-versa result in improvement. Building a model from enough text may be the key.

The primary goal of our evaluation was to assess the benefit of path coreference within a competitive pronoun resolution system. Our system does, however, outperform previously published results on these datasets. Direct comparison of our scoring system to other current top approaches is made difficult by differences in preprocessing. Ideally we would assess the benefit of our probabilistic features using the same state-of-the-art preprocessing modules employed by others such as (Yang et al., 2005) (who additionally use a search engine for compatibility scoring). Clearly, promoting competitive evaluation of pronoun resolution scoring systems by giving competitors equivalent real-world preprocessing output along the lines of (Barbu and Mitkov, 2001) remains the best way to isolate areas for system improvement.

Our pronoun resolution system is part of a larger information retrieval project where resolution ac-

[5] We calculate significance with McNemar's test, p=0.05.

curacy is not necessarily the most pertinent measure of classifier performance. More than one candidate can be useful in ambiguous cases, and not every resolution need be used. Since the SVM ranks antecedent candidates, we can test this ranking by selecting more than the top candidate (Top-n) and evaluating coverage of the true antecedents. We can also resolve only those instances where the most likely candidate is above a certain distance from the SVM threshold. Varying this distance varies the precision-recall (PR) of the overall resolution. A representative PR curve for the Top-n classifiers is provided (Figure 2). The corresponding information retrieval performance can now be evaluated along the Top-n / PR configurations.

7 Conclusion

We have introduced a novel feature for pronoun resolution called path coreference, and demonstrated its significant contribution to a state-of-the-art pronoun resolution system. This feature aids coreference decisions in many situations not handled by traditional coreference systems. Also, by bootstrapping with the coreferent paths, we are able to build the most complete and accurate table of probabilistic gender information yet available. Preliminary experiments show path coreference bootstrapping can also provide a means of identifying pleonastic pronouns, where pleonastic neutral pronouns are often followed in a dependency path by a terminal noun of different gender, and cataphoric constructions, where the pronouns are often followed by nouns of matching gender.

References

Chinatsu Aone and Scott William Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 122–129.

Catalina Barbu and Ruslan Mitkov. 2001. Evaluation tool for rule-based anaphora resolution methods. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 34–41.

David L. Bean and Ellen Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *HLT-NAACL*, pages 297–304.

Shane Bergsma. 2005. Automatic acquisition of gender information for anaphora resolution. In *Proceedings of the Eighteenth Canadian Conference on Artificial Intelligence (Canadian AI'2005)*, pages 342–353.

Colin Cherry and Shane Bergsma. 2005. An expectation maximization approach to pronoun resolution. In *Proceedings of the Ninth Conference on Natural Language Learning (CoNLL-2005)*, pages 88–95.

Kenneth Ward Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL'89)*, pages 76–83.

Ido Dagan and Alan Itai. 1990. Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, volume 3, pages 330–332, Helsinki, Finland.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–171.

Liliane Haegeman. 1994. *Introduction to Government & Binding theory: Second Edition*. Basil Blackwell, Cambridge, UK.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf and C. Burges, editors, *Advances in Kernel Methods*. MIT-Press.

Andrew Kehler, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of HLT/NAACL-04*, pages 289–296.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Ruslan Mitkov. 1997. Factors in anaphora resolution: they are not the only things that matter. a case study based on two different approaches. In *Proceedings of the ACL '97 / EACL '97 Workshop on Operational Factors in Practical, Robust Anaphora Resolution*, pages 14–21.

MUC-7. 1997. Coreference task definition (v3.0, 13 Jul 97). In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Xiaofeng Yang, Jian Su, and Chew Lim Tan. 2005. Improving pronoun resolution using statistics-based semantic compatibility information. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 165–172, June.