An Expectation Maximization Approach to Pronoun Resolution

Colin Cherry and **Shane Bergsma**
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
{colinc,bergsma}@cs.ualberta.ca

Abstract

We propose an unsupervised Expectation Maximization approach to pronoun resolution. The system learns from a fixed list of potential antecedents for each pronoun. We show that unsupervised learning is possible in this context, as the performance of our system is comparable to supervised methods. Our results indicate that a probabilistic gender/number model, determined automatically from unlabeled text, is a powerful feature for this task.

1 Introduction

Coreference resolution is the process of determining which expressions in text refer to the same real-world entity. Pronoun resolution is the important yet challenging subset of coreference resolution where a system attempts to establish coreference between a pronominal anaphor, such as a third-person pronoun like *he, she, it,* or *they*, and a preceding noun phrase, called an antecedent. In the following example, a pronoun resolution system must determine the correct antecedent for the pronouns "his" and "he."

(1) When the president entered the arena with his family, he was serenaded by a mariachi band.

Pronoun resolution has applications across many areas of Natural Language Processing, particularly in the field of information extraction. Resolving a pronoun to a noun phrase can provide a new interpretation of a given sentence, giving a Question Answering system, for example, more data to consider.

Our approach is a synthesis of linguistic and statistical methods. For each pronoun, a list of antecedent candidates derived from the parsed corpus is presented to the Expectation Maximization (EM) learner. Special cases, such as pleonastic, reflexive and cataphoric pronouns are dealt with linguistically during list construction. This allows us to train on and resolve all third-person pronouns in a large Question Answering corpus. We learn lexicalized gender/number, language, and antecedent probability models. These models, tied to individual words, can not be learned with sufficient coverage from labeled data. Pronouns are resolved by choosing the most likely antecedent in the candidate list according to these distributions. The resulting resolution accuracy is comparable to supervised methods.

We gain further performance improvement by initializing EM with a gender/number model derived from special cases in the training data. This model is shown to perform reliably on its own. We also demonstrate how the models learned through our unsupervised method can be used as features in a supervised pronoun resolution system.

2 Related Work

Pronoun resolution typically employs some combination of constraints and preferences to select the antecedent from preceding noun phrase candidates. Constraints filter the candidate list of improbable antecedents, while preferences encourage selection of antecedents that are more recent, frequent, etc. Implementation of constraints and preferences can be based on empirical insight (Lappin and Leass, 1994), or machine learning from a reference-

Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL),
pages 88–95, Ann Arbor, June 2005.

annotated corpus (Ge et al., 1998). The majority of pronoun resolution approaches have thus far relied on manual intervention in the resolution process, such as using a manually-parsed corpus, or manually removing difficult non-anaphoric cases; we follow Mitkov et al.'s approach (2002) with a fully-automatic pronoun resolution method. Parsing, noun-phrase identification, and non-anaphoric pronoun removal are all done automatically.

Machine-learned, fully-automatic systems are more common in noun phrase coreference resolution, where the method of choice has been decision trees (Soon et al., 2001; Ng and Cardie, 2002). These systems generally handle pronouns as a subset of all noun phrases, but with limited features compared to systems devoted solely to pronouns. Kehler used Maximum Entropy to assign a probability distribution over possible noun phrase coreference relationships (1997). Like his approach, our system does not make hard coreference decisions, but returns a distribution over candidates.

The above learning approaches require annotated training data for supervised learning. Cardie and Wagstaff developed an unsupervised approach that partitions noun phrases into coreferent groups through clustering (1999). However, the partitions they generate for a particular document are not useful for processing new documents, while our approach learns distributions that can be used on unseen data. There are also approaches to anaphora resolution using unsupervised methods to extract useful information, such as gender and number (Ge et al., 1998), or contextual role-knowledge (Bean and Riloff, 2004). Co-training can also leverage unlabeled data through weakly-supervised reference resolution learning (Müller et al., 2002). As an alternative to co-training, Ng and Cardie (2003) use EM to augment a supervised coreference system with unlabeled data. Their feature set is quite different, as it is designed to generalize from the data in a labeled set, while our system models individual words. We suspect that the two approaches can be combined.

Our approach is inspired by the use of EM in bilingual word alignment, which finds word-to-word correspondences between a sentence and its translation. The prominent statistical methods in this field are unsupervised. Our methods are most influenced by IBM's Model 1 (Brown et al., 1993).

3 Methods

3.1 Problem formulation

We will consider our training set to consist of (p, k, C) triples: one for each pronoun, where p is the pronoun to be resolved, k is the pronoun's context, and C is a candidate list containing the nouns p could potentially be resolved to. Initially, we take k to be the parsed sentence that p appears in.

C consists of all nouns and pronouns that precede p, looking back through the current sentence and the sentence immediately preceding it. This small window may seem limiting, but we found that a correct candidate appeared in 97% of such lists in a labeled development text. Mitkov et al. also limit candidate consideration to the same window (2002). Each triple is processed with non-anaphoric pronoun handlers (Section 3.3) and linguistic filters (Section 3.4), which produce the final candidate lists.

Before we pass the (p, k, C) triples to EM, we modify them to better suit our EM formulation. There are four possibilities for the gender and number of third-person pronouns in English: masculine, feminine, neutral and plural (e.g., *he, she, it, they*). We assume a noun is equally likely to corefer with any member of a given gender/number category, and reduce each p to a category label accordingly. For example, *he*, *his*, *him* and *himself* are all labeled as *masc* for masculine pronoun. Plural, feminine and neutral pronouns are handled similarly. We reduce the context term k to p's immediate syntactic context, including only p's syntactic parent, the parent's part of speech, and p's relationship to the parent, as determined by a dependency parser. Incorporating context only through the governing constituent was also done in (Ge et al., 1998). Finally, each candidate in C is augmented with ordering information, so we know how many nouns to "step over" before arriving at a given candidate. We will refer to this ordering information as a candidate's j term, for jump. Our example sentence in Section 1 would create the two triples shown in Figure 1, assuming the sentence began the document it was found in.

3.2 Probability model

Expectation Maximization (Dempster et al., 1977) is a process for filling in unobserved data probabilistically. To use EM to do unsupervised pronoun reso-

his:	p = *masc* k = p's family C = arena (0), president (1)
he:	p = *masc* k = serenade p C = family (0), *masc* (1), arena (2), president (3)

Figure 1: EM input for our example sentence. j-values follow each lexical candidate.

lution, we phrase the resolution task in terms of hidden variables of an observed process. We assume that in each case, one candidate from the candidate list is selected as the antecedent before p and k are generated. EM's role is to induce a probability distribution over candidates to maximize the likelihood of the (p, k) pairs observed in our training set:

$$\Pr(Dataset) = \prod_{(p,k) \in Dataset} \Pr(p, k) \quad (1)$$

We can rewrite $\Pr(p, k)$ so that it uses a hidden candidate (or antecedent) variable c that influences the observed p and k:

$$\Pr(p, k) = \sum_{c \in C} \Pr(p, k, c) \quad (2)$$

$$\Pr(p, k, c) = \Pr(p, k|c)\Pr(c) \quad (3)$$

To improve our ability to generalize to future cases, we use a naïve Bayes assumption to state that the choices of pronoun and context are conditionally independent, given an antecedent. That is, once we select the word the pronoun represents, the pronoun and its context are no longer coupled:

$$\Pr(p, k|c) = \Pr(p|c)\Pr(k|c) \quad (4)$$

We can split each candidate c into its lexical component l and its jump value j. That is, $c = (l, j)$. If we assume that l and j are independent, and that p and k each depend only on the l component of c, we can combine Equations 3 and 4 to get our final formulation for the joint probability distribution:

$$\Pr(p, k, c) = \Pr(p|l)\Pr(k|l)\Pr(l)\Pr(j) \quad (5)$$

The jump term j, though important when resolving pronouns, is not likely to be correlated with any lexical choices in the training set.

Table 1: Examples of learned pronoun probabilities.

Word (l)	*masc*	*fem*	*neut*	*plur*
company	0.03	0.01	0.95	0.01
president	0.94	0.01	0.03	0.02
teacher	0.19	0.71	0.09	0.01

This results in four models that work together to determine the likelihood of a given candidate. The $\Pr(p|l)$ distribution measures the likelihood of a pronoun given an antecedent. Since we have collapsed the observed pronouns into groups, this models a word's affinity for each of the four relevant gender/number categories. We will refer to this as our **pronoun model**. $\Pr(k|l)$ measures the probability of the syntactic relationship between a pronoun and its parent, given a prospective antecedent for the pronoun. This is effectively a **language model**, grading lexical choice by context. $\Pr(l)$ measures the probability that the word l will be found to be an antecedent. This is useful, as some entities, such as "president" in newspaper text, are inherently more likely to be referenced with a pronoun. Finally, $\Pr(j)$ measures the likelihood of jumping a given number of noun phrases backward to find the correct candidate. We represent these models with table look-up. Table 1 shows selected l-value entries in the $\Pr(p|l)$ table from our best performing EM model. Note that the probabilities reflect biases inherent in our news domain training set.

Given models for the four distributions above, we can assign a probability to each candidate in C according to the observations p and k; that is, $\Pr(c|p, k)$ can be obtained by dividing Equation 5 by Equation 2. Remember that $c = (l, j)$.

$$\Pr(c|p, k) = \frac{\Pr(p|l)\Pr(k|l)\Pr(l)\Pr(j)}{\sum_{c' \in C} \Pr(p|l')\Pr(k|l')\Pr(l')\Pr(j')} \quad (6)$$

$\Pr(c|p, k)$ allows us to get fractional counts of (p, k, c) triples in our training set, as if we had actually observed c co-occurring with (p, k) in the proportions specified by Equation 6. This estimation process is effectively the E-step in EM.

The M-step is conducted by redefining our models according to these fractional counts. For example, after assigning fractional counts to candidates

according to $\Pr(c|p,k)$, we re-estimate $\Pr(p|l)$ with the following equation for a specific (p,l) pair:

$$\Pr(p|l) = \frac{N(p,l)}{N(l)} \quad (7)$$

where $N()$ counts the number of times we see a given event or joint event throughout the training set.

Given trained models, we resolve pronouns by finding the candidate $\hat{c}$ that is most likely for the current pronoun, that is $\hat{c} = \text{argmax}_{c \in C} \Pr(c|p,k)$. Because $\Pr(p,k)$ is constant with respect to c, $\hat{c} = \text{argmax}_{c \in C} \Pr(p,k,c)$.

3.3 Non-anaphoric Pronouns

Not every pronoun in text refers anaphorically to a preceding noun phrase. There are a frequent number of difficult cases that require special attention, including pronouns that are:

- Pleonastic: pronouns that have a grammatical function but do not reference an entity. E.g. "It is important to observe it is raining."
- Cataphora: pronouns that reference a future noun phrase. E.g. "In his speech, the president praised the workers."
- Non-noun referential: pronouns that refer to a verb phrase, sentence, or implicit concept. E.g. "John told Mary they should buy a car."

If we construct them naïvely, the candidate lists for these pronouns will be invalid, introducing noise in our training set. Manual handling or removal of these cases is infeasible in an unsupervised approach, where the input is thousands of documents. Instead, pleonastics are identified syntactically using an extension of the detector developed by Lappin and Leass (1994). Roughly 7% of all pronouns in our labeled test data are pleonastic. We detect cataphora using a pattern-based method on parsed sentences, described in (Bergsma, 2005b). Future nouns are only included when cataphora are identified. This approach is quite different from Lappin and Leass (1994), who always include all future nouns from the current sentence as candidates, with a constant penalty added to possible cataphoric resolutions. The cataphora module identifies 1.4% of test data pronouns to be cataphoric; in each instance this identification is correct. Finally, we know of no approach that handles pronouns referring to verb phrases or implicit entities. The unavoidable errors for these pronouns, occurring roughly 4% of the time, are included in our final results.

3.4 Candidate list modifications

It would be possible for C to include every noun phrase in the current and previous sentence, but performance can be improved by automatically removing improbable antecedents. We use a standard set of constraints to filter candidates. If a candidate's gender or number is known, and does not match the pronoun's, the candidate is excluded. Candidates with known gender include other pronouns, and names with gendered designators (such as "Mr." or "Mrs."). Our parser also identifies plurals and some gendered first names. We remove from C all times, dates, addresses, monetary amounts, units of measurement, and pronouns identified as pleonastic.

We use the syntactic constraints from Binding Theory to eliminate candidates (Haegeman, 1994). For the reflexives *himself, herself, itself* and *themselves*, this allows immediate syntactic identification of the antecedent. These cases become unambiguous; only the indicated antecedent is included in C.

We improve the quality of our training set by removing known noisy cases before passing the set to EM. For example, we anticipate that sentences with quotation marks will be problematic, as other researchers have observed that quoted text requires special handling for pronoun resolution (Kennedy and Boguraev, 1996). Thus we remove pronouns occurring in the same sentences as quotes from the learning process. Also, we exclude triples where the constraints removed all possible antecedents, or where the pronoun was deemed to be pleonastic. Performing these exclusions is justified for training, but in testing we state results for all pronouns.

3.5 EM initialization

Early in the development of this system, we were impressed with the quality of the pronoun model $\Pr(p|l)$ learned by EM. However, we found we could construct an even more precise pronoun model for common words by examining unambiguous cases in our training data. Unambiguous cases are pronouns having only one word in their candidate list C. This could be a result of the preprocessors described in

Sections 3.3 and 3.4, or the pronoun's position in the document. A $\Pr_U(p|l)$ model constructed from only unambiguous examples covers far fewer words than a learned model, but it rarely makes poor gender/number choices. Furthermore, it can be obtained without EM. Training on unambiguous cases is similar in spirit to (Hindle and Rooth, 1993). We found in our development and test sets that, after applying filters, roughly 9% of pronouns occur with unambiguous antecedents.

When optimizing a probability function that is not concave, the EM algorithm is only guaranteed to find a local maximum; therefore, it can be helpful to start the process near the desired end-point in parameter space. The unambiguous pronoun model described above can provide such a starting point. When using this **initializer**, we perform our initial E-step by weighting candidates according to $\Pr_U(p|l)$, instead of weighting them uniformly. This biases the initial E-step probabilities so that a strong indication of the gender/number of a candidate from unambiguous cases will either boost the candidate's chances or remove it from competition, depending on whether or not the predicted category matches that of the pronoun being resolved.

To deal with the sparseness of the $\Pr_U(p|l)$ distribution, we use add-1 smoothing (Jeffreys, 1961). The resulting effect is that words with few unambiguous occurrences receive a near-uniform gender/number distribution, while those observed frequently will closely match the observed distribution. During development, we also tried clever initializers for the other three models, including an extensive language model initializer, but none were able to improve over $\Pr_U(p|l)$ alone.

3.6 Supervised extension

Even though we have justified Equation 5 with reasonable independence assumptions, our four models may not be combined optimally for our pronoun resolution task, as the models are only approximations of the true distributions they are intended to represent. Following the approach in (Och and Ney, 2002), we can view the right-hand-side of Equation 5 as a special case of:

$$\exp\left(\begin{array}{c}\lambda_1 \log \Pr(p|l) + \lambda_2 \log \Pr(k|l) + \\ \lambda_3 \log \Pr(l) + \lambda_4 \log \Pr(j)\end{array}\right) \quad (8)$$

where $\forall i : \lambda_i = 1$. Effectively, the log probabilities of our models become feature functions in a log-linear model. When labeled training data is available, we can use the Maximum Entropy principle (Berger et al., 1996) to optimize the λ weights.

This provides us with an optional supervised extension to the unsupervised system. Given a small set of data that has the correct candidates indicated, such as the set we used while developing our unsupervised system, we can re-weight the final models provided by EM to maximize the probability of observing the indicated candidates. To this end, we follow the approach of (Och and Ney, 2002) very closely, including their handling of multiple correct answers. We use the limited memory variable metric method as implemented in Malouf's maximum entropy package (2002) to set our weights.

4 Experimental Design

4.1 Data sets

We used two training sets in our experiments, both drawn from the AQUAINT Question Answering corpus (Vorhees, 2002). For each training set, we manually labeled pronoun antecedents in a corresponding **key** containing a subset of the pronouns in the set. These keys are drawn from a collection of complete documents. For each document, all pronouns are included. With the exception of the supervised extension, the keys are used only to validate the resolution decisions made by a trained system. Further details are available in (Bergsma, 2005b).

The development set consists of 333,000 pronouns drawn from 31,000 documents. The development key consists of 644 labeled pronouns drawn from 58 documents; 417 are drawn from sentences without quotation marks. The development set and its key were used to guide us while designing the probability model, and to fine-tune EM and smoothing parameters. We also use the development key as labeled training data for our supervised extension.

The test set consists of 890,000 pronouns drawn from 50,000 documents. The test key consists of 1209 labeled pronouns drawn from 118 documents; 892 are drawn from sentences without quotation marks. All of the results reported in Section 5 are determined using the test key.

4.2 Implementation Details

To get the context values and implement the syntactic filters, we parsed our corpora with Minipar (Lin, 1994). Experiments on the development set indicated that EM generally began to overfit after 2 iterations, so we stop EM after the second iteration, using the models from the second M-step for testing. During testing, ties in likelihood are broken by taking the candidate closest to the pronoun.

The EM-produced models need to be smoothed, as there will be unseen words and unobserved (p, l) or (k, l) pairs in the test set. This is because problematic cases are omitted from the training set, while all pronouns are included in the key. We handle out-of-vocabulary events by replacing words or context-values that occur only once during training with a special **unknown** symbol. Out-of-vocabulary events encountered during testing are also treated as unknown. We handle unseen pairs with additive smoothing. Instead of adding 1 as in Section 3.5, we add $\delta_p = 0.00001$ for (k, l) pairs, and $\delta_w = 0.001$ for (p, l) pairs. These δ values were determined experimentally with the development key.

4.3 Evaluation scheme

We evaluate our work in the context of a fully automatic system, as was done in (Mitkov et al., 2002). Our evaluation criteria is similar to their *resolution etiquette*. We define accuracy as the proportion of pronouns correctly resolved, either to any coreferent noun phrase in the candidate list, or to the pleonastic category, which precludes resolution. Systems that handle and state performance for all pronouns in unrestricted text report much lower accuracy than most approaches in the literature. Furthermore, automatically parsing and pre-processing texts causes consistent degradation in performance, regardless of the accuracy of the pronoun resolution algorithm. To have a point of comparison to other fully-automatic approaches, note the resolution etiquette score reported in (Mitkov et al., 2002) is 0.582.

5 Results

5.1 Validation of unsupervised method

The key concern of our work is whether enough useful information is present in the pronoun's category, context, and candidate list for unsupervised learning of antecedents to occur. To that end, our first set of experiments compare the pronoun resolution accuracy of our EM-based solutions to that of a previous-noun baseline on our test key. The results are shown in Table 2. The columns split the results into three cases: all pronouns with no exceptions; all cases where the pronoun was found in a sentence containing no quotation marks (and therefore resembling the training data provided to EM); and finally all pronouns excluded by the second case. We compare the following methods:

1. **Previous noun**: Pick the candidate from the filtered list with the lowest j value.
2. **EM, no initializer**: The EM algorithm trained on the test set, starting from a uniform E-step.
3. **Initializer, no EM**: A model that ranks candidates using only a pronoun model built from unambiguous cases (Section 3.5).
4. **EM w/ initializer**: As in (2), but using the initializer in (3) for the first E-step.
5. **Maxent extension**: The models produced by (4) are used as features in a log-linear model trained on the development key (Section 3.6).
6. **Upper bound**: The percentage of cases with a correct answer in the filtered candidate list.

For a reference point, picking the previous noun before applying any of our candidate filters receives an accuracy score of 0.281 on the "All" task.

Looking at the "All" column in Table 2, we see EM can indeed learn in this situation. Starting from uniform parameters it climbs from a 40% baseline to a 60% accurate model. However, the initializer can do slightly better with precise but sparse gender/number information alone. As we hoped, combining the initializer and EM results in a statistically significant[1] improvement over EM with a uniform starting point, but it is not significantly better than the initializer alone. The advantage of the EM process is that it produces multiple models, which can be re-weighted with maximum entropy to reach our highest accuracy, roughly 67%. The λ weights that achieve this score are shown in Table 3.

Maximum entropy leaves the pronoun model $\Pr(p|l)$ nearly untouched and drastically reduces the

[1] Significance is determined throughout Section 5 using McNemar's test with a significance level $\alpha = 0.05$.

Table 2: Accuracy for all cases, all excluding sentences with quotes, and only sentences with quotes.

Method	All	No“ ”	Only“ ”
1 Previous noun	0.397	0.399	0.391
2 EM, no initializer	0.610	0.632	0.549
3 Initializer, no EM	0.628	0.642	0.587
4 EM w/ initializer	0.632	0.663	0.546
5 Maxent extension	0.669	0.696	0.593
6 Upper bound	0.838	0.868	0.754

Table 3: Weights set by maximum entropy.

Model	$\Pr(p\|l)$	$\Pr(k\|l)$	$\Pr(l)$	$\Pr(j)$
Lambda	0.931	0.056	0.070	0.167

Table 4: Comparison to SVM.

Method	Accuracy
Previous noun	0.398
EM w/ initializer	0.664
Maxent extension	0.708
SVM	0.714

influence of all other models (Table 3). This, combined with the success of the initializer alone, leads us to believe that a strong notion of gender/number is very important in this task. Therefore, we implemented EM with several models that used only pronoun category, but none were able to surpass the initializer in accuracy on the test key. One factor that might help explain the initializer's success is that despite using only a $\Pr_U(p|l)$ model, the initializer also has an implicit factor resembling a $\Pr(l)$ model: when two candidates agree with the category of the pronoun, add-1 smoothing ensures the more frequent candidate receives a higher probability.

As was stated in Section 3.4, sentences with quotations were excluded from the learning process because the presence of a correct antecedent in the candidate list was less frequent in these cases. This is validated by the low upper bound of 0.754 in the only-quote portion of the test key. We can see that all methods except for the previous noun heuristic score noticeably better when ignoring those sentences that contain quotation marks. In particular, the difference between our three unsupervised solutions ((2), (3) and (4)) are more pronounced. Much of the performance improvements that correspond to our model refinements are masked in the overall task because adding the initializer to EM does not improve EM's performance on quotes at all. Developing a method to construct more robust candidate lists for quotations could improve our performance on these cases, and greatly increase the percentage of pronouns we are training on for a given corpus.

5.2 Comparison to supervised system

We put our results in context by comparing our methods to a recent supervised system. The comparison system is an SVM that uses 52 linguistically-motivated features, including probabilistic gender/number information obtained through web queries (Bergsma, 2005a). The SVM is trained with 1398 separate labeled pronouns, the same training set used in (Bergsma, 2005a). This data is also drawn from the news domain. Note the supervised system was not constructed to handle all pronoun cases, so non-anaphoric pronouns were removed from the test key and from the candidate lists in the test key to ensure a fair comparison. As expected, this removal of difficult cases increases the performance of our system on the test key (Table 4). Also note there is no significant difference in performance between our supervised extension and the SVM. The completely unsupervised EM system performs worse, but with only a 7% relative reduction in performace compared to the SVM; the previous noun heuristic shows a 44% reduction.

5.3 Analysis of upper bound

If one accounts for the upper bound in Table 2, our methods do very well on those cases where a correct answer actually appears in the candidate list: the best EM solution scores 0.754, and the supervised extension scores 0.800. A variety of factors result in the 196 candidate lists that do not contain a true antecedent. 21% of these errors arise from our limited candidate window (Section 3.1). Incorrect pleonastic detection accounts for another 31% while non-

noun referential pronouns cause 25% (Section 3.3). Linguistic filters (Section 3.4) account for most of the remainder. An improvement in any of these components would result in not only higher final scores, but cleaner EM training data.

6 Conclusion

We have demonstrated that unsupervised learning is possible for pronoun resolution. We achieve accuracy of 63% on an all-pronoun task, or 75% when a true antecedent is available to EM. There is now motivation to develop cleaner candidate lists and stronger probability models, with the hope of surpassing supervised techniques. For example, incorporating antecedent context, either at the sentence or document level, may boost performance. Furthermore, the lexicalized models learned in our system, especially the pronoun model, are potentially powerful features for any supervised pronoun resolution system.

References

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

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

Shane Bergsma. 2005a. Automatic acquisition of gender information for anaphora resolution. In *Proceedings of the 18th Conference of the Canadian Society for Computational Intelligence (Canadian AI 2005)*, pages 342–353, Victoria, BC.

Shane Bergsma. 2005b. Corpus-based learning for pronominal anaphora resolution. Master's thesis, Department of Computing Science, University of Alberta, Edmonton. http://www.cs.ualberta.ca/~bergsma/Pubs/thesis.pdf.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Claire Cardie and Kiri Wagstaff. 1999. Noun phrase coreference as clustering. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 82–89.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

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.

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

Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Harold Jeffreys, 1961. *Theory of Probability*, chapter 3.23. Oxford: Clarendon Press, 3rd edition.

Andrew Kehler. 1997. Probabilistic coreference in information extraction. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 163–173.

Christopher Kennedy and Branimir Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 113–118.

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

Dekang Lin. 1994. Principar - an efficient, broad-coverage, principle-based parser. In *Proceedings of COLING-94*, pages 42–48, Kyoto, Japan.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55.

Ruslan Mitkov, Richard Evans, and Constantin Orasan. 2002. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 168–186.

Christoph Müller, Stefan Rapp, and Michael Strube. 2002. Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 352–359.

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.

Vincent Ng and Claire Cardie. 2003. Weakly supervised natural language learning without redundant views. In *HLT-NAACL 2003: Proceedings of the Main Conference*, pages 173–180.

Franz J. Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, PA, July.

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.

Ellen Vorhees. 2002. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text REtrieval Conference (TREC)*.