Conundrums in Noun Phrase Coreference Resolution: Making Sense of the State-of-the-Art

Veselin Stoyanov
Cornell University
Ithaca, NY
ves@cs.cornell.edu

Nathan Gilbert
University of Utah
Salt Lake City, UT
ngilbert@cs.utah.edu

Claire Cardie
Cornell University
Ithaca, NY
cardie@cs.cornell.edu

Ellen Riloff
University of Utah
Salt Lake City, UT
riloff@cs.utah.edu

Abstract

We aim to shed light on the state-of-the-art in NP coreference resolution by teasing apart the differences in the MUC and ACE task definitions, the assumptions made in evaluation methodologies, and inherent differences in text corpora. First, we examine three subproblems that play a role in coreference resolution: named entity recognition, anaphoricity determination, and coreference element detection. We measure the impact of each subproblem on coreference resolution and confirm that certain assumptions regarding these subproblems in the evaluation methodology can dramatically simplify the overall task. Second, we measure the performance of a state-of-the-art coreference resolver on several classes of anaphora and use these results to develop a quantitative measure for estimating coreference resolution performance on new data sets.

1 Introduction

As is common for many natural language processing problems, the state-of-the-art in noun phrase (NP) coreference resolution is typically quantified based on system performance on manually annotated text corpora. In spite of the availability of several benchmark data sets (e.g. MUC-6 (1995), ACE NIST (2004)) and their use in many formal evaluations, as a field we can make surprisingly few conclusive statements about the state-of-the-art in NP coreference resolution.

In particular, *it remains difficult to assess the effectiveness of different coreference resolution approaches, even in relative terms.* For example, the 91.5 F-measure reported by McCallum and Wellner (2004) was produced by a system using perfect information for several linguistic subproblems. In contrast, the 71.3 F-measure reported by Yang et al. (2003) represents a fully automatic end-to-end resolver. It is impossible to assess which approach truly performs best because of the dramatically different assumptions of each evaluation.

Results vary widely across data sets. Coreference resolution scores range from 85-90% on the ACE 2004 and 2005 data sets to a much lower 60-70% on the MUC 6 and 7 data sets (e.g. Soon et al. (2001) and Yang et al. (2003)). What accounts for these differences? Are they due to properties of the documents or domains? Or do differences in the coreference task definitions account for the differences in performance? Given a new text collection and domain, what level of performance should we expect?

We have little understanding of which aspects of the coreference resolution problem are handled well or poorly by state-of-the-art systems. Except for some fairly general statements, for example that proper names are easier to resolve than pronouns, which are easier than common nouns, there has been little analysis of which aspects of the problem have achieved success and which remain elusive.

The goal of this paper is to take initial steps toward making sense of the disparate performance results reported for NP coreference resolution. For our investigations, we employ a state-of-the-art classification-based NP coreference resolver and focus on the widely used MUC and ACE coreference resolution data sets.

We hypothesize that performance variation within and across coreference resolvers is, at least in part, a function of (1) the (sometimes unstated) assumptions in evaluation methodologies, and (2) the relative difficulty of the benchmark text corpora. With these in mind, Section 3 first examines three subproblems that play an important role in coreference resolution: *named entity recognition*, *anaphoricity determination*, and *coreference element detection*. We quantitatively measure the impact of each of these subproblems on coreference resolution performance as a whole. Our results suggest that the availability of accurate detectors for anaphoricity or coreference elements could substantially improve the performance of state-of-the-art resolvers, while improvements to named entity recognition likely offer little gains. Our results also confirm that the assumptions adopted in

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 656–664,
Suntec, Singapore, 2-7 August 2009. ©2009 ACL and AFNLP

	MUC	ACE
Relative Pronouns	no	yes
Gerunds	no	yes
Nested non-NP nouns	yes	no
Nested NEs	no	GPE & LOC premod
Semantic Types	all	7 classes only
Singletons	no	yes

Table 1: Coreference Definition Differences for MUC and ACE. (GPE refers to geo-political entities.)

some evaluations dramatically simplify the resolution task, rendering it an unrealistic surrogate for the original problem.

In Section 4, we quantify the difficulty of a text corpus with respect to coreference resolution by analyzing performance on different resolution classes. Our goals are twofold: to measure the level of performance of state-of-the-art coreference resolvers on different types of anaphora, and to develop a quantitative measure for estimating coreference resolution performance on new data sets. We introduce a *coreference performance prediction (CPP)* measure and show that it accurately predicts the performance of our coreference resolver. As a side effect of our research, we provide a new set of much-needed benchmark results for coreference resolution under common sets of fully-specified evaluation assumptions.

2 Coreference Task Definitions

This paper studies the six most commonly used coreference resolution data sets. Two of those are from the MUC conferences (MUC-6, 1995; MUC-7, 1997) and four are from the Automatic Content Evaluation (ACE) Program (NIST, 2004). In this section, we outline the differences between the MUC and ACE coreference resolution tasks, and define terminology for the rest of the paper.

Noun phrase coreference resolution is the process of determining whether two noun phrases (NPs) refer to the same real-world entity or concept. It is related to anaphora resolution: a NP is said to be *anaphoric* if it depends on another NP for interpretation. Consider the following:

John Hall is the new CEO. He starts on Monday.

Here, *he* is anaphoric because it depends on its antecedent, *John Hall*, for interpretation. The two NPs also corefer because each refers to the same person, JOHN HALL.

As discussed in depth elsewhere (e.g. van Deemter and Kibble (2000)), the notions of coreference and anaphora are difficult to define precisely and to operationalize consistently. Furthermore, the connections between them are extremely complex and go beyond the scope of this paper. Given these complexities, it is not surprising that the annotation instructions for the MUC and ACE data sets reflect different interpretations and simplifications of the general coreference relation. We outline some of these differences below.

Syntactic Types. To avoid ambiguity, we will use the term **coreference element (CE)** to refer to the set of linguistic expressions that participate in the coreference relation, as defined for each of the MUC and ACE tasks.[1] At times, it will be important to distinguish between the CEs that are included in the gold standard — the *annotated CEs* — from those that are generated by the coreference resolution system — the *extracted CEs*.

At a high level, both the MUC and ACE evaluations define CEs as nouns, pronouns, and noun phrases. However, the MUC definition excludes (1) "nested" named entities (NEs) (e.g. "America" in "Bank of America"), (2) relative pronouns, and (3) gerunds, but allows (4) nested nouns (e.g. "union" in "union members"). The ACE definition, on the other hand, includes relative pronouns and gerunds, excludes **all** nested nouns that are not themselves NPs, and allows premodifier NE mentions of geo-political entities and locations, such as "Russian" in "Russian politicians".

Semantic Types. ACE restricts CEs to entities that belong to one of seven semantic classes: person, organization, geo-political entity, location, facility, vehicle, and weapon. MUC has no semantic restrictions.

Singletons. The MUC data sets include annotations only for CEs that are coreferent with at least one other CE. ACE, on the other hand, permits "singleton" CEs, which are not coreferent with any other CE in the document.

These substantial differences in the task definitions (summarized in Table 1) make it extremely difficult to compare performance across the MUC and ACE data sets. In the next section, we take a closer look at the coreference resolution task, analyzing the impact of various subtasks irrespective of the data set differences.

[1] We define the term CE to be roughly equivalent to (a) the notion of *markable* in the MUC coreference resolution definition and (b) the structures that can be *mentions* in the descriptions of ACE.

3 Coreference Subtask Analysis

Coreference resolution is a complex task that requires solving numerous non-trivial subtasks such as syntactic analysis, semantic class tagging, pleonastic pronoun identification and antecedent identification to name a few. This section examines the role of three such subtasks — *named entity recognition*, *anaphoricity determination*, and *coreference element detection* — in the performance of an end-to-end coreference resolution system. First, however, we describe the coreference resolver that we use for our study.

3.1 The RECONCILE$_{ACL09}$ Coreference Resolver

We use the RECONCILE coreference resolution platform (Stoyanov et al., 2009) to configure a coreference resolver that performs comparably to state-of-the-art systems (when evaluated on the MUC and ACE data sets under comparable assumptions). This system is a classification-based coreference resolver, modeled after the systems of Ng and Cardie (2002b) and Bengtson and Roth (2008). First it classifies pairs of CEs as coreferent or not coreferent, pairing each identified CE with all preceding CEs. The CEs are then clustered into coreference chains[2] based on the pairwise decisions. RECONCILE has a pipeline architecture with four main steps: preprocessing, feature extraction, classification, and clustering. We will refer to the specific configuration of RECONCILE used for this paper as RECONCILE$_{ACL09}$.

Preprocessing. The RECONCILE$_{ACL09}$ preprocessor applies a series of language analysis tools (mostly publicly available software packages) to the source texts. The OpenNLP toolkit (Baldridge, J., 2005) performs tokenization, sentence splitting, and part-of-speech tagging. The Berkeley parser (Petrov and Klein, 2007) generates phrase structure parse trees, and the de Marneffe et al. (2006) system produces dependency relations. We employ the Stanford CRF-based Named Entity Recognizer (Finkel et al., 2004) for named entity tagging. With these preprocessing components, RECONCILE$_{ACL09}$ uses heuristics to correctly extract approximately 90% of the annotated CEs for the MUC and ACE data sets.

Feature Set. To achieve roughly state-of-the-art performance, RECONCILE$_{ACL09}$ employs a fairly comprehensive set of 61 features introduced in previous coreference resolution systems (see Bengtson and Roth (2008)). We briefly summarize the features here and refer the reader to Stoyanov et al. (2009) for more details.

dataset	docs	CEs	chains	CEs/ch	tr/tst split
MUC6	60	4232	960	4.4	30/30 (st)
MUC7	50	4297	1081	3.9	30/20 (st)
ACE-2	159	2630	1148	2.3	130/29 (st)
ACE03	105	3106	1340	2.3	74/31
ACE04	128	3037	1332	2.3	90/38
ACE05	81	1991	775	2.6	57/24

Table 2: Dataset characteristics including the number of documents, annotated CEs, coreference chains, annotated CEs per chain (average), and number of documents in the train/test split. We use *st* to indicate a standard train/test split.

Lexical (9): String-based comparisons of the two CEs, such as exact string matching and head noun matching.

Proximity (5): Sentence and paragraph-based measures of the distance between two CEs.

Grammatical (28): A wide variety of syntactic properties of the CEs, either individually or as a pair. These features are based on part-of-speech tags, parse trees, or dependency relations. For example: one feature indicates whether both CEs are syntactic subjects; another indicates whether the CEs are in an appositive construction.

Semantic (19): Capture semantic information about one or both NPs such as tests for gender and animacy, semantic compatibility based on WordNet, and semantic comparisons of NE types.

Classification and Clustering. We configure RECONCILE$_{ACL09}$ to use the Averaged Perceptron learning algorithm (Freund and Schapire, 1999) and to employ *single-link clustering* (i.e. transitive closure) to generate the final partitioning.[3]

3.2 Baseline System Results

Our experiments rely on the MUC and ACE corpora. For ACE, we use only the newswire portion because it is closest in composition to the MUC corpora. Statistics for each of the data sets are shown in Table 2. When available, we use the standard test/train split. Otherwise, we randomly split the data into a training and test set following a 70/30 ratio.

[2] A coreference *chain* refers to the set of CEs that refer to a particular entity.

[3] In trial runs, we investigated alternative classification and clustering models (e.g. C4.5 decision trees and SVMs; best-first clustering). The results were comparable.

Scoring Algorithms. We evaluate using two common scoring algorithms[4] — MUC and B^3. The MUC scoring algorithm (Vilain et al., 1995) computes the F1 score (harmonic mean) of precision and recall based on the identifcation of unique coreference links. We use the official MUC scorer implementation for the two MUC corpora and an equivalent implementation for ACE.

The B^3 algorithm (Bagga and Baldwin, 1998) computes a precision and recall score for each CE:

$precision(ce) = |R_{ce} \cap K_{ce}|/|R_{ce}|$

$recall(ce) = |R_{ce} \cap K_{ce}|/|K_{ce}|$,

where R_{ce} is the coreference chain to which ce is assigned in the response (i.e. the system-generated output) and K_{ce} is the coreference chain that contains ce in the key (i.e. the gold standard). Precision and recall for a set of documents are computed as the mean over all CEs in the documents and the F1 score of precision and recall is reported.

B^3 Complications. Unlike the MUC score, which counts links between CEs, B^3 presumes that the gold standard and the system response are clusterings over the same set of CEs. This, of course, is not the case when the system automatically identifies the CEs, so the scoring algorithm requires a mapping between extracted and annotated CEs. We will use the term $twin(ce)$ to refer to the unique annotated/extracted CE to which the extracted/annotated CE is matched. We say that a CE is *twinless* (has no twin) if no corresponding CE is identified. A twinless extracted CE signals that the resolver extracted a spurious CE, while an annotated CE is twinless when the resolver fails to extract it.

Unfortunately, it is unclear how the B^3 score should be computed for twinless CEs. Bengtson and Roth (2008) simply discard twinless CEs, but this solution is likely too lenient — it doles no punishment for mistakes on twinless annotated or extracted CEs and it would be tricked, for example, by a system that extracts only the CEs about which it is most confident.

We propose two different ways to deal with twinless CEs for B^3. One option, B^3all, retains all twinless extracted CEs. It computes the precision as above when ce has a twin, and computes the precision as $1/|R_{ce}|$ if ce is twinless. (Similarly, $recall(ce) = 1/|K_{ce}|$ if ce is twinless.)

The second option, $B^3 0$, discards twinless extracted CEs, but penalizes recall by setting $recall(ce) = 0$ for all twinless annotated CEs. Thus, $B^3 0$ presumes that all twinless extracted CEs are spurious.

Results. Table 3, box 1 shows the performance of RECONCILE$_{ACL09}$ using a default (0.5) coreference classifier threshold. The MUC score is highest for the MUC6 data set, while the four ACE data sets show much higher B^3 scores as compared to the two MUC data sets. The latter occurs because the ACE data sets include singletons.

The classification threshold, however, can be gainfully employed to control the trade-off between precision and recall. This has not traditionally been done in learning-based coreference resolution research — possibly because there is not much training data available to sacrifice as a validation set. Nonetheless, we hypothesized that estimating a threshold *from just the training data* might be effective. Our results (BASELINE box in Table 3) indicate that this indeed works well.[5] With the exception of MUC6, results on all data sets and for all scoring algorithms improve; moreover, the scores approach those for runs using an optimal threshold (box 3) for the experiment as determined by using the **test set**. In all remaining experiments, we learn the threshold from the training set as in the BASELINE system.

Below, we resume our investigation of the role of three coreference resolution subtasks and measure the impact of each on overall performance.

3.3 Named Entities

Previous work has shown that resolving coreference between proper names is relatively easy (e.g. Kameyama (1997)) because string matching functions specialized to the type of proper name (e.g. person vs. location) are quite accurate. Thus, we would expect a coreference resolution system to depend critically on its Named Entity (NE) extractor. On the other hand, state-of-the-art NE taggers are already quite good, so improving this component may not provide much additional gain.

To study the influence of NE recognition, we replace the system-generated NEs of

[4] We also experimented with the CEAF score (Luo, 2005), but excluded it due to difficulties dealing with the extracted, rather than annotated, CEs. CEAF assigns a zero score to each twinless extracted CE and weights all coreference chains equally, irrespective of their size. As a result, runs with extracted CEs exhibit very low CEAF precision, leading to unreliable scores.

[5] All experiments sample uniformly from 1000 threshold values.

Reconcile$_{ACL09}$		MUC6	MUC7	ACE-2	ACE03	ACE04	ACE05
1. DEFAULT THRESHOLD (0.5)	MUC	70.40	58.20	65.76	66.73	56.75	64.30
	B^3all	69.91	62.88	77.25	77.56	73.03	72.82
	B^30	68.55	62.80	76.59	77.27	72.99	72.43
2. BASELINE = THRESHOLD ESTIMATION	MUC	68.50	62.80	65.99	67.87	62.03	67.41
	B^3all	70.88	65.86	78.29	79.39	76.50	73.71
	B^30	68.43	64.57	76.63	77.88	75.41	72.47
3. OPTIMAL THRESHOLD	MUC	71.20	62.90	66.83	68.35	62.11	67.41
	B^3all	72.31	66.52	78.50	79.41	76.53	74.25
	B^30	69.49	64.64	76.83	78.27	75.51	72.94
4. BASELINE with perfect NEs	MUC	69.90	-	66.37	70.35	62.88	67.72
	B^3all	72.31	-	78.06	80.22	77.01	73.92
	B^30	67.91	-	76.55	78.35	75.22	72.90
5. BASELINE with perfect CEs	MUC	85.80*	81.10*	76.39	79.68	76.18	79.42
	B^3all	76.14	75.88	78.65	80.58	77.79	76.49
	B^30	76.14	75.88	78.65	80.58	77.79	76.49
6. BASELINE with anaphoric CEs	MUC	82.20*	71.90*	86.63	85.58	83.33	82.84
	B^3all	72.52	69.26	80.29	79.71	76.05	74.33
	B^30	72.52	69.26	80.29	79.71	76.05	74.33

Table 3: Impact of Three Subtasks on Coreference Resolution Performance. A score marked with a * indicates that a 0.5 threshold was used because threshold selection from the training data resulted in an extreme version of the system, i.e. one that places all CEs into a single coreference chain.

RECONCILE$_{ACL09}$ with gold-standard NEs and retrain the coreference classifier. Results for each of the data sets are shown in box 4 of Table 3. (No gold standard NEs are available for MUC7.) Comparison to the BASELINE system (box 2) shows that using gold standard NEs leads to improvements on all data sets with the exception of ACE2 and ACE05, on which performance is virtually unchanged. The improvements tend to be small, however, between 0.5 to 3 performance points. We attribute this to two factors. First, as noted above, although far from perfect, NE taggers generally perform reasonably well. Second, only 20 to 25% of the coreference element resolutions required for these data sets involve a proper name (see Section 4).

Conclusion #1: Improving the performance of NE taggers is not likely to have a large impact on the performance of state-of-the-art coreference resolution systems.

3.4 Coreference Element Detection

We expect CE detection to be an important subproblem for an end-to-end coreference system. Results for a system that assumes perfect CEs are shown in box 5 of Table 3. For these runs, RECONCILE$_{ACL09}$ uses only the annotated CEs for both training and testing. Using perfect CEs solves a large part of the coreference resolution task: the annotated CEs divulge anaphoricity information, perfect NP boundaries, and perfect information regarding the coreference relation defined for the data set.

We see that focusing attention on all and only the annotated CEs leads to (often substantial) improvements in performance on all metrics over all data sets, especially when measured using the MUC score.

Conclusion #2: Improving the ability of coreference resolvers to identify coreference elements would likely improve the state-of-the-art immensely — by 10-20 points in MUC F1 score and from 2-12 F1 points for B^3.

This finding explains previously published results that exhibit striking variability when run with annotated CEs vs. system-extracted CEs. On the MUC6 data set, for example, the best published MUC score using extracted CEs is approximately 71 (Yang et al., 2003), while multiple systems have produced MUC scores of approximately 85 when using annotated CEs (e.g. Luo et al. (2004), McCallum and Wellner (2004)).

We argue that providing a resolver with the annotated CEs is a rather unrealistic evaluation: determining whether an NP is part of an annotated coreference chain is precisely the job of a coreference resolver!

Conclusion #3: Assuming the availability of CEs unrealistically simplifies the coreference resolution task.

3.5 Anaphoricity Determination

Finally, several coreference systems have successfully incorporated anaphoricity determination modules (e.g. Ng and Cardie (2002a) and Bean and Riloff (2004)). The goal of the module is to determine whether or not an NP is anaphoric. For example, pleonastic pronouns (e.g. *it is raining*) are special cases that do not require coreference resolution.

Unfortunately, neither the MUC nor the ACE data sets include anaphoricity information for all NPs. Rather, they encode anaphoricity information implicitly for annotated CEs: a CE is considered anaphoric if is not a singleton.[6]

To study the utility of anaphoricity information, we train and test only on the "anaphoric" **extracted** CEs, i.e. the extracted CEs that have an annotated twin that is not a singleton. Note that for the MUC datasets all extracted CEs that have twins are considered anaphoric.

Results for this experiment (box 6 in Table 3) are similar to the previous experiment using perfect CEs: we observe big improvements across the board. This should not be surprising since the experimental setting is quite close to that for perfect CEs: this experiment also presumes knowledge of when a CE is part of an annotated coreference chain. Nevertheless, we see that anaphoricity infomation is important. First, good anaphoricity identification should reduce the set of extracted CEs making it closer to the set of annotated CEs. Second, further improvements in MUC score for the ACE data sets over the runs using perfect CEs (box 5) reveal that accurately determining anaphoricity can lead to substantial improvements in MUC score. ACE data includes annotations for singleton CEs, so knowling whether an annotated CE is anaphoric divulges additional information.

Conclusion #4: An accurate anaphoricity determination component can lead to substantial improvement in coreference resolution performance.

4 Resolution Complexity

Different types of anaphora that have to be handled by coreference resolution systems exhibit different properties. In linguistic theory, binding mechanisms vary for different kinds of syntactic constituents and structures. And in practice, empirical results have confirmed intuitions that different types of anaphora benefit from different classifier features and exhibit varying degrees of difficulty (Kameyama, 1997). However, performance evaluations rarely include analysis of where state-of-the-art coreference resolvers perform best and worst, aside from general conclusions.

In this section, we analyze the behavior of our coreference resolver on different types of anaphoric expressions with two goals in mind. First, we want to deduce the strengths and weaknesses of state-of-the-art systems to help direct future research. Second, we aim to understand why current coreference resolvers behave so inconsistently across data sets. Our hypothesis is that the distribution of different types of anaphoric expressions in a corpus is a major factor for coreference resolution performance. Our experiments confirm this hypothesis and we use our empirical results to create a *coreference performance prediction (CPP)* measure that successfully estimates the expected level of performance on novel data sets.

4.1 Resolution Classes

We study the *resolution complexity* of a text corpus by defining *resolution classes*. Resolution classes partition the set of anaphoric CEs according to properties of the anaphor and (in some cases) the antecedent. Previous work has studied performance differences between pronominal anaphora, proper names, and common nouns, but we aim to dig deeper into subclasses of each of these groups. In particular, we distinguish between proper and common nouns that can be resolved via string matching, versus those that have no antecedent with a matching string. Intuitively, we expect that it is easier to resolve the cases that involve string matching. Similarly, we partition pronominal anaphora into several subcategories that we expect may behave differently. We define the following nine *resolution classes*:

Proper Names: Three resolution classes cover CEs that are named entities (e.g. the PERSON, LOCATION, ORGANIZATION and DATE classes for MUC and ACE) and have a prior referent[7] in the text. These three classes are distinguished by the type of antecedent that can be resolved against the proper name.

(1) **PN-e:** a proper name is assigned to this *exact string match* class if there is at least one preceding CE in its gold standard coreference chain that exactly matches it.

(2) **PN-p:** a proper name is assigned to this *partial string match* class if there is at least one preceding CE in its gold standard chain that has some content words in common.

(3) **PN-n:** a proper name is assigned to this *no string match*

[6] Also, the first element of a coreference chain is usually non-anaphoric, but we do not consider that issue here.

[7] We make a rough, but rarely inaccurate, assumption that there are no cataphoric expressions in the data.

	MUC6			MUC7			ACE2			ACE03			ACE04			ACE05			*Avg*	
	#	%	scr	#	%	scr	#	%	scr	#	%	scr	#	%	scr	#	%	scr	%	scr
PN-e	273	17	.87	249	19	.79	346	24	.94	435	25	.93	267	16	.88	373	31	.92	22	.89
PN-p	157	10	.68	79	6	.59	116	8	.86	178	10	.87	194	11	.71	125	10	.71	9	.74
PN-n	18	1	.18	18	1	.28	85	6	.19	79	4	.15	66	4	.21	89	7	.27	4	.21
CN-e	292	18	.82	276	21	.65	84	6	.40	186	11	.68	165	10	.68	134	11	.79	13	.67
CN-p	229	14	.53	239	18	.49	147	10	.26	168	10	.24	147	9	.40	147	12	.43	12	.39
CN-n	194	12	.27	148	11	.15	152	10	.50	148	8	.90	266	16	.32	121	10	.20	11	.18
1+2Pr	48	3	.70	65	5	.66	122	8	.73	76	4	.73	158	9	.77	51	4	.61	6	.70
G3Pr	160	10	.73	50	4	.79	181	12	.83	237	13	.82	246	14	.84	69	60	.81	10	.80
U3Pr	175	11	.49	142	11	.49	163	11	.45	122	7	.48	153	9	.49	91	7	.49	9	.48

Table 4: Frequencies and scores for each resolution class.

class if no preceding CE in its gold standard chain has any content words in common with it.

Common NPs: Three analogous string match classes cover CEs that have a common noun as a head: (4) **CN-e** (5) **CN-p** (6) **CN-n**.

Pronouns: Three classes cover pronouns:

(7) **1+2Pr:** The anaphor is a 1st or 2nd person pronoun.

(8) **G3Pr:** The anaphor is a gendered 3rd person pronoun (e.g. "she", "him").

(9) **U3Pr:** The anaphor is an ungendered 3rd person pronoun.

As noted above, resolution classes are defined for annotated CEs. We use the twin relationship to match extracted CEs to annotated CEs and to evaluate performance on each resolution class.

4.2 Scoring Resolution Classes

To score each resolution class separately, we define a new variant of the MUC scorer. We compute a **MUC-RC** score (for MUC Resolution Class) for class C as follows: we assume that all CEs that do not belong to class C are resolved correctly by taking the correct clustering for them from the gold standard. Starting with this correct partial clustering, we run our classifier on all ordered pairs of CEs for which the **s**econd CE is of class C, essentially asking our coreference resolver to determine whether each member of class C is coreferent with each of its preceding CEs. We then count the number of unique correct/incorrect links that the system introduced on top of the correct partial clustering and compute precision, recall, and F1 score. This scoring function directly measures the impact of each resolution class on the overall MUC score.

4.3 Results

Table 4 shows the results of our resolution class analysis on the test portions of the six data sets. The # columns show the frequency counts for each resolution class, and the % columns show the distributions of the classes in each corpus (i.e. 17% of all resolutions in the MUC6 corpus were in the **PN-e** class). The **scr** columns show the MUC-RC score for each resolution class. The right-hand side of Table 4 shows the average distribution and scores across all data sets.

MUC6	MUC7	ACE2	ACE03	ACE04	ACE05
0.92	0.95	0.91	0.98	0.97	0.96

Table 5: Correlations of resolution class scores with respect to the average.

These scores confirm our expectations about the relative difficulty of different types of resolutions. For example, it appears that proper names are easier to resolve than common nouns; gendered pronouns are easier than 1st and 2nd person pronouns, which, in turn, are easier than ungendered 3rd person pronouns. Similarly, our intuition is confirmed that many CEs can be accurately resolved based on exact string matching, whereas resolving against antecedents that do not have overlapping strings is much more difficult. The average scores in Table 4 show that performance varies dramatically across the resolution classes, but, on the surface, appears to be relatively consistent across data sets.

None of the data sets performs exactly the same, of course, so we statistically analyze whether the behavior of each resolution class is similar across the data sets. For each data set, we compute the correlation between the vector of MUC-RC scores over the resolution classes and the average vector of MUC-RC scores for the remaining five data sets. Table 5 contains the results, which show high correlations (over .90) for all six data sets. These results indicate that the relative performance of the resolution classes is consistent across corpora.

4.4 Coreference Performance Prediction

Next, we hypothesize that the distribution of resolution classes in a corpus explains (at least partially) why performance varies so much from cor-

	MUC6	MUC7	ACE2	ACE03	ACE04	ACE05
P	0.59	0.59	0.62	0.65	0.59	0.62
O	0.67	0.61	0.66	0.68	0.62	0.67

Table 6: Predicted (P) vs Observed (O) scores.

pus to corpus. To explore this issue, we create a *Coreference Performance Prediction (CPP)* measure to predict the performance on new data sets. The CPP measure uses the empirical performance of each resolution class observed on previous data sets and forms a predicton based on the make-up of resolution classes in a new corpus. The distribution of resolution classes for a new corpus can be easily determined because the classes can be recognized superficially by looking only at the strings that represent each NP.

We compute the CPP score for each of our six data sets based on the average resolution class performance measured on the other five data sets. The predicted score for each class is computed as a weighted sum of the observed scores for each resolution class (i.e. the mean for the class measured on the other five data sets) weighted by the proportion of CEs that belong to the class. The predicted scores are shown in Table 6 and compared with the MUC scores that are produced by RECONCILE$_{ACL09}$.[8]

Our results show that the CPP measure is a good predictor of coreference resolution performance on unseen data sets, with the exception of one outlier – the MUC6 data set. In fact, the correlation between predicted and observed scores is 0.731 for all data sets and 0.913 excluding MUC6. RECONCILE$_{ACL09}$'s performance on MUC6 is better than predicted due to the higher than average scores for the common noun classes. We attribute this to the fact that MUC6 includes annotations for nested nouns, which almost always fall in the **CN-e** and **CN-p** classes. In addition, many of the features were first created for the MUC6 data set, so the feature extractors are likely more accurate than for other data sets.

Overall, results indicate that coreference performance is substantially influenced by the mix of resolution classes found in the data set. Our CPP measure can be used to produce a good estimate of the level of performance on a new corpus.

5 Related Work

The bulk of the relevant related work is described in earlier sections, as appropriate. This paper studies complexity issues for NP coreference resolution using a "good", i.e. near state-of-the-art, system. For state-of-the-art performance on the MUC data sets see, e.g. Yang et al. (2003); for state-of-the-art performance on the ACE data sets see, e.g. Bengtson and Roth (2008) and Luo (2007). While other researchers have evaluated NP coreference resolvers with respect to pronouns vs. proper nouns vs. common nouns (Ng and Cardie, 2002b), our analysis focuses on measuring the complexity of data sets, predicting the performance of coreference systems on new data sets, and quantifying the effect of coreference system subcomponents on overall performance. In the related area of anaphora resolution, researchers have studied the influence of subsystems on the overall performance (Mitkov, 2002) as well as defined and evaluated performance on different classes of pronouns (e.g. Mitkov (2002) and Byron (2001)). However, due to the significant differences in task definition, available datasets, and evaluation metrics, their conclusions are not directly applicable to the full coreference task.

Previous work has developed methods to predict system performance on NLP tasks given data set characteristics, e.g. Birch et al. (2008) does this for machine translation. Our work looks for the first time at predicting the performance of NP coreference resolvers.

6 Conclusions

We examine the state-of-the-art in NP coreference resolution. We show the relative impact of perfect NE recognition, perfect anaphoricity information for coreference elements, and knowledge of all and only the annotated CEs. We also measure the performance of state-of-the-art resolvers on several classes of anaphora and use these results to develop a measure that can accurately estimate a resolver's performance on new data sets.

Acknowledgments. We gratefully acknowledge technical contributions from David Buttler and David Hysom in creating the Reconcile coreference resolution platform. This research was supported in part by the Department of Homeland Security under ONR Grant N0014-07-1-0152 and Lawrence Livermore National Laboratory subcontract B573245.

[8] Observed scores for MUC6 and 7 differ slightly from Table 3 because this part of the work did not use the OPTIONAL field of the key, employed by the official MUC scorer.

References

A. Bagga and B. Baldwin. 1998. Algorithms for Scoring Coreference Chains. In *In Linguistic Coreference Workshop at LREC 1998.*

Baldridge, J. 2005. *The OpenNLP project. http://opennlp.sourceforge.net/.*

D. Bean and E. Riloff. 2004. Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2004).*

Eric Bengtson and Dan Roth. 2008. Understanding the Value of Features for Coreference Resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting Success in Machine Translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754. Association for Computational Linguistics.

Donna Byron. 2001. The Uncommon Denominator: A Proposal for Consistent Reporting of Pronoun Resolution Results. *Computational Linguistics*, 27(4):569–578.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *LREC.*

J. Finkel, S. Dingare, H. Nguyen, M. Nissim, and C. Manning. 2004. Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web. In *Joint Workshop on Natural Language Processing in Biomedicine and its Applications at COLING 2004.*

Yoav Freund and Robert E. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. In *Machine Learning*, pages 277–296.

Megumi Kameyama. 1997. Recognizing Referential Links: An Information Extraction Perspective. In *Workshop On Operational Factors In Practical Robust Anaphora Resolution For Unrestricted Texts.*

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics.*

X. Luo. 2005. On Coreference Resolution Performance Metrics. In *Proceedings of the 2005 Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing.*

Xiaoqiang Luo. 2007. Coreference or Not: A Twin Model for Coreference Resolution. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2007).*

A. McCallum and B. Wellner. 2004. Conditional Models of Identity Uncertainty with Application to Noun Coreference. In *18th Annual Conference on Neural Information Processing Systems.*

Ruslan Mitkov. 2002. *Anaphora Resolution*. Longman, London.

MUC-6. 1995. Coreference Task Definition. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 335–344.

MUC-7. 1997. Coreference Task Definition. In *Proceedings of the Seventh Message Understanding Conference (MUC-7).*

V. Ng and C. Cardie. 2002a. Identifying Anaphoric and Non-Anaphoric Noun Phrases to Improve Coreference Resolution. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002).*

V. Ng and C. Cardie. 2002b. Improving Machine Learning Approaches to Coreference Resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

NIST. 2004. *The ACE Evaluation Plan.*

S. Petrov and D. Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2007).*

W. Soon, H. Ng, and D. Lim. 2001. A Machine Learning Approach to Coreference of Noun Phrases. *Computational Linguistics*, 27(4):521–541.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, Ellen Riloff, David Buttler, and David Hysom. 2009. Reconcile: A Coreference Resolution Research Platform. Computer Science Technical Report, Cornell University, Ithaca, NY.

Kees van Deemter and Rodger Kibble. 2000. On Coreferring: Coreference in MUC and Related Annotation Schemes. *Computational Linguistics*, 26(4):629–637.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A Model-Theoretic Coreference Scoring Theme. In *Proceedings of the Sixth Message Understanding Conference (MUC-6).*

Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference Resolution Using Competition Learning Approach. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 176–183.