Data-Driven Coreference Resolution

Sanda M. Harabagiu and **Răzvan Bunescu**
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
{sanda,razvan}@seas.smu.edu

Abstract

In this paper we present an approach to coreference resolution that integrates empirical methods with machine learning techniques. This approach departs from previous solutions for reference resolution, in that it promotes data-driven techniques instead of relying on combinations of linguistic and cognitive aspects of discourse. The immediate pragmatic result is an enhancement of precision and recall.

Background

Reference resolution is presupposed by any natural language processing (NLP) system that tackles the structure of discourse or dialogue. To be able to summarize texts coherently or to find correct answers to a question from a large collection of on-line documents, we need to have access to the discourse structure. Reference relations are important components of this structure, as they represent identity, part-whole, type-token, or set-membership relations. The subcase of the reference resolution that considers only identity between textual expressions is known as *coreference*. Coreference is the only form of reference that we address in this paper. Thus far, the best-performing and most robust coreference resolution systems have employed knowledge-based techniques. Traditionally, these techniques have combined extensive syntactic, semantic, and discourse knowledge. The acquisition of such knowledge is time-consuming, difficult, and error-prone. Nevertheless, recent results show that empirical methods perform with amazing accuracy (cf. (Mitkov 1998) (Kennedy and Boguraev 1996)). For example, COGNIAC (Baldwin 1997), a system based on seven ordered heuristics, generates high-precision resolution (over 90%) for some cases of pronominal reference.

In our work, we revisited the concept of empirical coreference resolution by developing several different sets of heuristics corresponding to the various forms of coreference, e.g. there are heuristics for the resolution of 3rd person pronouns distinct from heuristics that solve reflexive pronouns or possessive pronouns. Similarly, we have developed separate heuristics for the resolution of definite, bare or indefinite nominals. The resulting system, named COCKTAIL[1], resolves coreference by exploiting several cohesive constraints (e.g. term repetition) combined with lexical and coherence cues (e.g. subjects of communication verbs are more likely to refer to the last person mentioned in the text). Moreover, the COCKTAIL framework uniformly addresses the problem of interaction between different forms of coreference.

Each heuristic implemented in COCKTAIL is the result of mining patterns of coreference in a very large data set obtained with a novel annotation methodology, applied to the MUC-6 and MUC-7 coreference keys used in recent Message Understanding Conferences (MUC) (MUC-6 1996). The heuristics discovered from the MUC data operate under the assumption that the text is preprocessed, to determine referential expressions prior to their application.

The rest of the paper is organized as follows. Section 2 defines our data-driven methodology for coreference resolution whereas Section 3 presents several heuristics encoded in COCKTAIL. Section 4 presents the bootstrapping mechanism. Section 5 reports and discusses the experimental results, whereas Section 6 summarizes the conclusions.

Data-Driven Coreference Resolution

Very generally, what we consider as *data-driven methodology* is a sequence of actions that captures the data patterns capable of resolving a problem with both high precision and high recall. In our case, a data-driven methodology comprises the actions that generate *sets of heuristics* for the coreference resolution problem. We define the *precision* of these heuristics by the number of correct references out of the total number of coreferences resolved, whereas the *recall* of coreference heuristics measures the number of resolved references out of the total number of references known in a test set.

[1] COCKTAIL is a pun on COGNIAC, because COCKTAIL uses multiple sets of ordered heuristics, blended together in a single system.

The data driven methodology used in COCKTAIL is centered around the notion of a *coreference chain*. Due to the *transitivity* of coreference relations, any k coreference relations having at least one common argument generate $k+1$ *coreferring expressions*. The text position induces an order among coreferring expressions. A *coreference structure* is created when a set of coreferring expressions are connected in an oriented graph, such that each node is related only to one of its preceding nodes. In turn, a *coreference chain* is the coreference structure in which every node is connected to its immediately preceding node. Clearly, multiple coreference structures for the same set of coreferring expressions can be mapped in a single coreference chain. As an example, both coreference structures illustrated in Figure 1(a) and (c) are cast into the coreference chain illustrated in Figure 1(b).

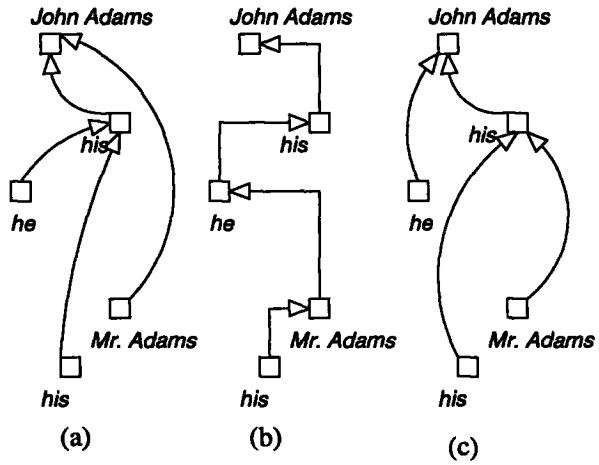

Figure 1: Coreference structures vs. coreference chains.

Given a corpus annotated with coreference data, our data-driven methodology first generates all coreference chains in the data set and then considers all possible combinations of coreference relations that would generate the same coreference chains. For a coreference chain of length l, with nodes n_1, n_2, ... n_{l+1}, multiple coreference structures can be created given that each node n_k ($1 \leq k \leq l$) can be connected to any of the $k-1$ nodes preceding it. From this observation, we find that a number of $1 \times 2 \times ... \times (l-k) ... \times l - 1 = l! - 1$ coreference structures can generate the same coreference chain. This result is very important, since it allows us to automatically generate coreference data.

For each coreference chain, we generate new coreference links when we desire to obtain all corresponding coreference structures. If a coreference chain of length l generates n^l_{new} new relations, then the number of new relations generated by a coreference chain of length $l+1$ is $n^l_{new} = n^{l-1}_{new} + l - 2$. This recursive equation solves $n^l_{new} = 1 + 2 + 3 + ... + (l-2) = \frac{(l-1)(l-2)}{2}$. Table 1 shows the number of coreference chains in each MUC corpus as well as the number of original anaphoric links. It also shows the number of new anaphoric links that were generated on both corpora. The overall expansion factor of coreference links is 13.5. The longest coreference chain in the MUC-6 annotations has 49 relations, whereas the longest chain from MUC-7 has 62 relations.

Corpus	Number of coreference chains	Number of **original** anaphoric relations	Number of **new** anaphoric relations
MUC-6	463	1461	15467
MUC-7	738	2245	34619

Table 1: Annotated coreference data and new relations.

The data-driven approach is a two-tiered procedure. First, new anaphoric links are generated, and coreference rules that have the largest coverage and no negative examples are derived. They represent the seed heuristics, and are manually generated. Since many anaphors remain unresolved, in a second phase, new coreference rules are bootstrapped to enhance the recall of coreference.

As a rule of thumb, to have the best seed heuristics, we consider a heuristic only if there is massive evidence of its coverage in the data. To measure this coverage we need to have a lot of coreference data available. For this purpose we have implemented a coreference annotation procedure, coined as AUTOTAG-COREF[2]. The annotation procedure is:

1. For every coreference annotation R(x,y) in a text T
2. Create its coreference chain CC(R);
2.1. CC(R) is initially NULL;
2.2. Add a new coreference R' to CC(R) if
either - R and R' have a common argument, or
- R' and any relation R" from CC(R) have a common argument
2.3. Sort CC(R) using the text order of the 2nd argument of each relation
3. For every referential expression E in CC
4. For every expression E' that precedes E in CC
5. if R(E,E') is not in CC
then Create NEW_Reference_link(E,E')

We are not aware of any other automated way of creating coreference annotated data and we claim that most of the impressive performance of COCKTAIL is due to AUTOTAG-COREF.

Empirical Coreference Resolution

The result of our data-driven methodology is the set of heuristics implemented in COCKTAIL which cover both nominal and pronoun coreference. Each heuristic represents a pattern of coreference that was mined from the large set of coreference data. The heuristics from COCKTAIL can be classified along two directions. First of all, they can be grouped according to the type of coreference they resolve, e.g., heuristics that resolve the

[2]The name was inspired by Riloff's AUTO-SLOG (Riloff 1996), the system capable of automatically acquiring linguistic patterns for Information Extraction.

Heuristics for 3rd person pronouns	Heuristics for nominal reference
∘*Heuristic 1-Pronoun*(H1Pron) *Search in the same sentence for the* **same** *3rd person pronoun Pron'* **if** *(Pron' belongs to coreference chain CC)* **and** *there is an element from CC which is closest to Pron in Text, Pick that element.* **else** *Pick Pron'.* ∘*Heuristic 2-Pronoun*(H2Pron) *Search for PN, the closest proper name from Pron* **if** *(PN agrees in number and gender with Pron)* **if** *(PN belongs to coreference chain CC)* **then** *Pick the element from CC which is closest to Pron in Text.* **else** *Pick PN.* ∘*Heuristic 3-Pronoun*(H3Pron) *Search for Noun, the closest noun from Pron* **if** *(Noun agrees in number and gender with Pron)* **if** *(Noun belongs to coreference chain CC)* **and** *there is an element from CC which is closest to Pron in Text, Pick that element.* **else** *Pick Noun*	∘*Heuristic 1-Nominal*(H1Nom) **if** (*Noun* is the head of an appositive) **then** Pick the preceding NP. ∘*Heuristic 2-Nominal*(H2Nom) **if** (*Noun belongs to an NP, Search for NP' such that Noun'=same_name(head(NP),head(NP'))* **or** *Noun'=same_name(adjunct(NP),adjunct(NP'))*) **then if** *(Noun' belongs to coreference chain CC)* **then** *Pick the element from CC which is closest to Noun in Text.* **else** *Pick Noun'.* ∘*Heuristic 3-Nominal*(H3Nom) **if** *Noun is the head of an NP* **then** *Search for proper name PN such that head(PN)=Noun* **if** *(PN belongs to coreference chain CC)* **and** *there is an element from CC which is closest to Noun in Text, Pick that element.* **else** *Pick PN.*

Table 2: Best performing heuristics implemented in COCKTAIL

anaphors of reflexive pronouns operate differently than those resolving bare nominals. Currently, in COCKTAIL there are heuristics that resolve five types of pronouns (personal, possessive, reflexive, demonstrative and relative) and three forms of nominals (definite, bare and indefinite).

Example of the application of heuristic H2Pron
Mr. Adams$_1$, 69 years old, is the retired chairman of Canadian-based Emco Ltd., a maker of plumbing and petroleum equipment; *he*$_1$ has served on the Woolworth board since 1981.
Example of the application of heuristic H3Pron
"We have got to stop pointing our fingers at *these kids*$_2$ who have no future," he said, "and reach our hands out to *them*$_2$.
Example of the application of heuristic H2Nom
The chairman and *the chief executive officer*$_3$ of Woolworth Corp. have temporarily relinquished their posts while the retailer conducts its investigation into *alleged accounting irregularities*$_4$. Woolworth's board named John W. Adams, an outsider, to serve as interim chairman and *executive officer*$_3$, while a special committee, appointed by the board last week and led by Mr. Adams, investigates *the irregularities*$_4$.

Table 3: Examples of coreference resolution. The same annotated index indicates coreference.

Secondly, for each type of coreference, there is an order in which they are applied. Initially, this order is based on their suitability to resolve coreference, as noticed from the annotated data. The order resulted from the analysis of the distribution of the antecedents in the MUC annotated data. For example, repetitions of named entities and appositives account for the majority of the nominal coreferences, and, therefore, represent anchors for the first heuristics that are applied. Table 2 lists the top performing heuristics of COCKTAIL for pronominal and nominal coreference. Examples of the heuristics operation on the MUC data are presented in Table 3.

Combining Coreference Heuristics

The order in which heuristics are applied is very important, since a referent may satisfy the conditions of more than one heuristic. Because of this, initially we have grouped the heuristics corresponding to each type of referent (e.g. possessives, reflexives, 3rd person pronouns) into a separate, ordered set. However, this solution does not filter out possible false positives, i.e. cases in which a referent is connected to the wrong anaphor, which belongs to a different coreference chain. To address this problem, we have developed a methodology that proposed a set of coreference chains by maximizing an entropy-based measure.

Given a text T we consider all its referential expressions $\mathcal{RE}(T)=\{NP_1, NP_2, ..., NP_m\}$, a subset of the text noun phrases. To derive the coreference chains spanning the elements from $\mathcal{RE}(T)$ we use a set of heuristics $\mathcal{H}=\{h_1, h_2, ..., h_n\}$. The application of these heuristics generates a partion of $\mathcal{RE}(T)$. Each parition is a set of coreference chains ($Par=\{CC_k^{Par}\}$) such that each $NP_i \in (\mathcal{RE})$ belongs to one and only one of the coreference chains CC_k^{Par} .

We denote by $\mathcal{P}(\mathcal{RE})$ all the possible partitions on $\mathcal{RE}(T)$. For every partition $Par \in \mathcal{P}(\mathcal{RE})$ we define a measure $m(Par, \mathcal{H})$ which estimates the likelihood that Par contains all the correct coreference links from the text T. Formally, given a text T, we look for the most

likely partition defined by coreference chains, given by:

$$Par_{best} = argmax_{Par \in \mathcal{P}(\mathcal{RE})} m(Par, \mathcal{H})$$

in which $m(Par, \mathcal{H})$ is defined by the sum between two factors:

$$m(Par, \mathcal{H}) = m^{+}(Par, \mathcal{H}) + m^{-}(Par, \mathcal{H})$$

The two factors are defined as:

1. $m^{+}(\mathcal{P}, \mathcal{H})$ indicates the *internal cohesion* of each coreference chain from Par. Formally it is defined as a sum ranging over all pairs of referents that belong to the *same* coreference chain in Par:

$$m^{+}(\mathcal{P}, \mathcal{H}) = \sum rel(NP_i, NP_j)$$

2. $m^{-}(\mathcal{P}, \mathcal{H})$ indicates the *discrimination* among all the coreference chains from Par. Formally it is defined as a sum ranging over all pairs of referents that belong to *different* coreference chains in Par:

$$m^{-}(\mathcal{P}, \mathcal{H}) = \sum -rel(NP_i, NP_j)$$

To measure $rel(NP_i, NP_j)$, the likelihood that NP_i and NP_j corefer, we use a binary function $a{:}\mathcal{H} \times \mathcal{RE} \times \mathcal{RE} \rightarrow \{0,1\}$. Given the set $\mathcal{H}$, whenever $h_k \in \mathcal{H}$ can be applied to NP_i and NP_j results as its antecedent, we have $a(h_k, NP_i, NP_j) = 1$; otherwise we have $a(h_k, NP_i, NP_j) = 0$. In this way, for every pair (NP_i, NP_j) we generate a vector:

$$v_{ij} = < a(h_1, NP_i, NP_j), ..., a(h_n, NP_i, NP_j) >$$

If all the coreference data produced by AUTOTAG-COREF is considered, for each pair (NP_i, NP_j) there may be up to 2^n different vectors v_{ij}. For each specific vector v_{ij}, in the data produced by AUTOTAG-COREF there are p positive examples and n negative examples for which the same heuristics as in v_{ij} were applied. Given the numbers p and n associated with each vector v_{ij}, we compute $rel(NP_i, NP_j)$ with the formula:

$$rel(NP_i, NP_j) = \begin{cases} 1 - entropy(v_{ij}) & \text{if } p \geq n \\ entropy(v_{ij}) - 1 & \text{otherwise} \end{cases}$$

where the entropy measure is defined as:

$$entropy(v_{ij}) = -\frac{p}{p+n} log_2 \frac{p}{p+n} - \frac{n}{p+n} log_2 \frac{n}{p+n}$$

The rationale for the formula of $rel(NP_i, NP_j)$ is given by the fact that the entropy indicates how much information is still needed for establishing the coreference between NP_i and NP_j with certainty. As illustrated in Figure 2, if $p_+ = \frac{p}{p+n}$ then the closer p_+ is to 1, the more confidence we have in the coreference relation between NP_i and NP_j, and thus $rel(p^+)$ is closer to 1. Similarly, the closer p_+ is to 0, the more confidence we have in that NP_i and NP_j are not coreferent. When NP_i and NP_j do not corefer, $rel(p^+)$ is -1. This explains why we add the negative of $rel(NP_i, NP_j)$ in the formula of $m^{-}(Par, \mathcal{H})$.

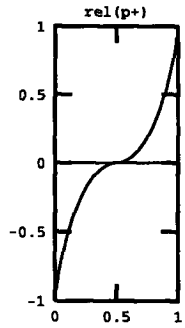

Figure 2: A function of coreference confidence.

Bootstrapping for Coreference Resolution

One of the major drawbacks of existing coreference resolution systems is their inability to recognize many forms of coreference displayed by many real-world texts. Recall measures of current systems range between 36% and 59% for both knowledge-based and statistical techniques. Knowledge based-systems would perform better if more coreference constraints were available whereas statistical methods would be improved if more annotated data were available. Since knowledge-based techniques outperform inductive methods, we used high-precision coreference heuristics as knowledge seeds for machine learning techniques that operate on large amounts of unlabeled data. One such technique is *bootstrapping*, which was recently presented in (Riloff and Jones 1999) as an ideal framework for text learning tasks that have knowledge seeds.

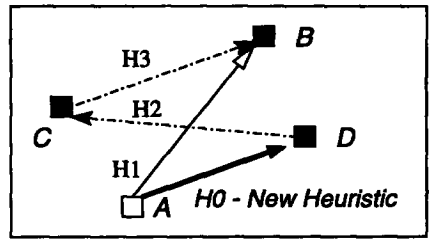

Figure 3: Bootstrapping new heuristics.

The coreference heuristics are the seeds of our bootstrapping framework for coreference resolution. When applied to large collections of texts, the heuristics determine classes of coreferring expressions. By generating coreference chains out of all these coreferring expressions, often new heuristics are uncovered. For example, Figure 3 illustrates the application of three heuristics and the generation of data for a new heuristic rule. In COCKTAIL, after a heuristic is applied, a new coreference chain is calculated. For the example illustrated in Figure 3, if the reference of expression A is sought, heuristic H_1 indicates expression B to be the antecedent. When the coreference chain is built, expression A is directly linked to expression D, thus uncovering a new heuristic H_0.

As a rule of thumb, we do not consider a new heuristic unless there is massive evidence of its coverage in the data. To measure the coverage we use the *FOIL_Gain measure*, as introduced by the FOIL inductive algorithm (Cameron-Jones and Quinlan 1993). Let H_{new} be the new heuristic and H_1 a heuristic that is already in the seed set. Let p_0 be the number of positive coreference examples of H_{new} (i.e. the number of coreference relations produced by the heuristic that can be found in the test data) and n_0 the number of negative examples of H_{new} (i.e. the number of relations generated by the heuristic which cannot be found in the test data). Similarly, p_1 and n_1 are the positive and negative examples of H_1. The new heuristics are scored by their *FOIL_Gain* distance to the existing set of heuristics, and the best scoring one is added to the COCKTAIL system. The *FOIL_Gain* formula is:

$$FOIL\_Gain(H_1, H_0) = k(log_2 \frac{p_1}{p_1 + n_1} - log_2 \frac{p_0}{p_0 + n_0})$$

where k is the number of positive examples covered by both H_1 and H_0. Heuristic H_0 is added to the seed set if there is no other heuristic providing larger *FOIL_Gain* to any of the seed heuristics. This mechanism of discovering and adding new heuristics to the set of coreference rules enables the following bootstrapping algorithm:

MUTUAL BOOTSTRAPPING LOOP

1. *Score all candidate heuristics with FOIL_Gain*
2. *Best_h=closest candidate to heuristics(COCKTAIL)*
3. *Add Best_h to heuristics(COCKTAIL)*
4. *Apply all heuristics(COCKTAIL) to the test data by combining the new and the old heuristics.*
5. *Goto step 1 if new heuristics could be uncovered and the precision and recall did not converge.*

Evaluation

To measure the performance of COCKTAIL we have used the MUC-6 and MUC-7 annotated data and computed the *precision*, the *recall* and van Rijsbergen's *F-measure* (which combines recall and precision equally) values. The performance measures have been obtained automatically using the MUC-6 coreference scoring program (Vilain et al. 1995). We performed cross-validations, by randomly selecting 10 texts from the MUC annotated corpus as test data, and the test of the MUC texts as training data. At the next step we chose randomly 10 new texts for test and trained on the remaining 50 texts. We repeated the selection on test data until we used the entire collection of coreference annotated texts. Table 4 lists the results.

Table 4 shows that the seed set of heuristics had good precision but poor recall. By combining the heuristics with the entropy-based measure, we the precision dropped drastically. However, the entropy measures helped both better precision and recall. In the future we intend to compare the overall effect of heuristics that recognize referential expressions on the overall performance of the system.

	Precision	Recall	F-measure
COCKTAIL heuristics	87.1%	61.7%	72.2%
COCKTAIL heuristics combined	76.7%	57.3%	71.3%
COCKTAIL +bootstrapping	92.0%	73.9%	81.9%

Table 4: Bootstrapping effect on COCKTAIL

Conclusion

We have introduced a new data-driven method for coreference resolution, implemented in the COCKTAIL system. Unlike other knowledge-poor methods for coreference resolution (Baldwin 1997) (Mitkov 1998), COCKTAIL filters its most performant heuristics through massive training data, generated by its AUTOTAG-COREF component. Furthermore, by using an entropy-based method we determine the best partition of coreferring expressions in *coreference chains*, and thus allow new heuristics to be learned and applied along with the initial ones. New heuristics are learned by applying a bootstrapping methodology. Due to the central role played by the notion of coreference chain, COCKTAIL provides a flexible approach of coordinating context-dependent and context-independent coreference constraints and preferences for partitioning nominal expressions into coreference equivalence classes.

References

Brack Baldwin. 1997. CogNIAC: high precision coreference with limited knowledge and linguistic resour ces. In *Proceedings of the ACL'97/EACL'97 Workshop on Operational factors in practical, robust anaphora resolution*, pages 38–45, Madrid, Spain.

Joseph F. Cameron-Jones and Ross Quinlan. 1993. Avoiding Pitfalls When Learning Recursive Theories. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 1050-1055.

Christopher Kennedy and Branimir Bogureav. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96).*

Ruslan Mitkov. 1998. Robust pronoun resolution with limited knowledge. In *Proceedings of COLING-ACL'98*, pages 869–875.

1996. *Proceedings of the Sixth Message Understanding Conference (MUC-6)* ,Morgan Kaufmann, San Mateo, CA.

Ellen Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049, Portland, OR, July.

Ellen Riloff and Rosie Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99).*

Marc Vilain, John Burger, John Aberdeen, Dan Connolly and Lynette Hirshman. 1995. A model-theoretic coreference scoring scheme. *Proceedings of the Sixth Message Understanding Conference (MUC-6)* ,Morgan Kaufmann, San Mateo, CA.