Text and Knowledge Mining for Coreference Resolution

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

Abstract

Traditionally coreference is resolved by satisfying a combination of salience, syntactic, semantic and discourse constraints. The acquisition of such knowledge is time-consuming, difficult and error-prone. Therefore, we present a knowledge-minimalist methodology of mining coreference rules from annotated text corpora. Semantic consistency evidence, which is a form of knowledge required by coreference, is easily retrieved from WordNet. Additional consistency knowledge is discovered by a *meta-bootstrapping* algorithm applied to unlabeled texts.

1 Background

Reference resolution is an important task for discourse or dialogue processing systems since *identity* relations between anaphoric textual entities and their antecedents is a prerequisite to the understanding of text or conversation. Traditionally, coreference resolution has been performed by combining linguistic and cognitive knowledge of language. Linguistic information is provided mostly by syntactic and semantic modeling of language whereas cognitive information is incorporated in computational models of discourse. Computational methods based on linguistic and congitive information were presented in (Hobbs 1978), (Lappin and Leass 1994), (Brennan et al.1987), (Grosz et al.1995) and (Webber 1988). The acquisition of extensive linguistic and discourse knowledge necessary for resolving coreference is time consuming, difficult and error-prone. Neverthless, recent results show that knowledge-poor, empirical methods perform with amazing accuracy on certain forms of coreference (cf. (Mitkov 1998) (Kennedy and Boguraev 1996) (Kameyama 1997)). For example, CogNIAC (Baldwin 1997), a system based on just seven ordered heuristics, generates high-precision resolution (over 90%) for some cases of pronominal reference.

In our work, we approached the coreference resolution problem by trying to determine how much more knowledge is required to supplement the above-mentioned knowledge-poor methods and how to derive that knowledge. To this end we (1) analyze the data to find what types of anaphor-antecedent pairs are most popular in real-world texts; (2) devise knowledge-minimalist rules for handling the majority of those popular cases; and (3) discover what supplementary knowledge is needed for remaining, more difficult cases.

To analyze coreference data we use a corpus of annotated texts. To devise minimalist coreference resolution rules we consider (1) strong indicators of cohesion, such as repetitions, name aliases or appositions; and (2) gender, number and class agreements. WordNet (Miller 1995), the vast semantic knowledge base, provides suplementary knowledge in the form of semantic consistency between coreferring nouns. Additional semantic consistency knowledge is generated by a *bootstrapping* mechanism when our coreference resolution system, Cocktail[1], processes new texts. This bootstrapping mechanism inspired by the technique presented in (Riloff and Jones 1999) targets one of the most problematic forms of knowledge needed for coreference resolution: the semantic consistency of corefering nominals.

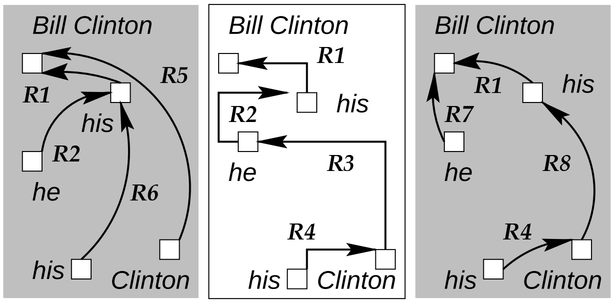
The rest of the paper is organized as follows. Section 2 discusses our text mining methodology for analysing the data and devising knowledge-minimalist rules for resolving the most popular coreference cases. Section 3 presents the knowledge-mining components of Cocktail that use WordNet for deriving semantic consistency as well as gender information. Section 4 presents an entropy-based method for optimally combining coreference rules and Section 5 presents the bootstrapping mechanism. Section 6 reports and discusses the experimental results while Section 7 summarizes the conclusions.

[1] Cocktail is a pun on CogNIAC, because Cocktail uses multiple coreference resolution rules corresponding to different forms of coreference, blended together in a single system.

2 Text Mining for Coreference Resolution

Information used for categorizing coreference resolution cases was mined from 30 documents manually annotated with SGML-coreference tags. The annotations contain information needed to establish a coreference link between an explicitly marked pair of noun phrases from the text. These 30 texts, used for training our procedure, constitute half of the 60 documents annotated for coreference and made available during the MUC-6 and MUC-7 Message Understanding Conferences (MUC)[2]. The remaining 30 documents were used for testing our coreference procedure and bootstrapping supplemental knowledge.

In order to generate the massive amount of data essential to our text mining approach, we expand the annotated tags from the training corpus. The expansion techniques make use of the properties of coreference relations. 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 determines 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).



(a) (b) (c)

Figure 1: Coreference structures vs. coreference chains.

The corefrence chain illustrated in Figure 1(b) is generated from the relations $\{R_1, R_2, R_3, R_4\}$. If it is the case that relations R_3 and R_4 are more difficult to identify automatically, since they rely on knowledge that is not easily available, the same set of coreferring expressions can be generated by the relations $\{R_1, R_2, R_5, R_6\}$, as illustrated in Figure 1(a). If relations R_2 and R_3 are difficult to identify automatically, the same set of coreferring expressions can be generated by the relations $\{R_1, R_4, R_7, R_8\}$, as ilustrated in Figure 1(c). From this example, we can see that if all the possible coreference relations were available, we could mine minimalist coreference knowledge that relies on relatively simpler syntactic and semantic information.

Corpus	Number of coreference chains	Number of **original** anaphoric relations	Number of **new** anaphoric relations
MUC-6	319	1461	11690
MUC-7	485	1845	15858

Table 1: Annotated coreference data and new relations.

To find out how many possible coreference relations we can generate from the tags of the training documents, we first compute the number of coreference structures that can be derived. Given a set of coreferring expressions n_1, n_2, ... n_{l+1}[3], if each node n_k ($1 \leq k \leq l$) is connected to any of the $k-1$ nodes preceding it in the document, we can generate $1 \times 2 \times ... \times (l-k) ... \times l - 1 = l! - 1$ coreference structures. The number of new coreference relations highlighted by these structures is computed by the recursive equation $n^l_{new} = n^{l-1}_{new} + l - 2$. The solution of this equation is $n^l_{new} = 1 + 2 + 3 + ... + (l-2) = \frac{(l-1)(l-2)}{2}$. We can thus hypothesize that it is possible to generate, from a set of set of 30 annotated texts, new relations that are an order of magnitude more numerous than the original annotated ones. This hypothesis is confirmed by the data listed in Table 1. Such a large number of new annotations cannot be derived manually and therefore we devised an automatic annotation procedure, called AutoTagCoref[4]. The algorithm for annotation is as follows:

```
1. For every coreference chain CC containing more
   than 2 expressions, e.g.CC={E(1),E(2),...,E(n)}
2.   For (k=n; k¿2; k=k-1)
3.     if (Relation(E(k),E(k+1)) $\neq$ Apposition)
4.       For (i=k+1; $i < n+1$; i=i+1)
5.         Add Annotated-Relation(E(k-1),E(i));
```

[2] (Hirschman et al.1998) describes the annotation procedure for the MUC texts that results in an inter-annotator agreement of 85%, similar to the results presented in (Wiebe et al. 1999) for the annotations of the gold-data set of subjectivity classifications. The same coreference data set was used in the development of several other robust coreference resolution algorithms (cf. (Kameyama 1997) (Cardie and Wagstaff 1999)).

[3] Expressions $n_1, n_2, ... n_{l+1}$ are cast in a coreference chain of length l.

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

Notice that apposition is not included in the above algorithm. AutoTag-Coref considers the coreference established by appositions as a special case. Because appositions can be detected fairly reliably, no other coreference annotation sourced at an appositive expression needs to be generated. Thus, for a coreference chain of length $l+1$, any apposition will reduce the number of new links by $l-2 = \frac{(l-1)(l-2)}{2} - \frac{(l-2)(l-3)}{2}$.

AutoTag-Coref adds not only a large number of new relations, as shown in Table 1, but it also changes the distribution of coreference relations. For example, in the original annotations, 18.4% of the links connect anaphors to proper nouns whereas the new relations changed this percentage to 29.1%. Furthermore, the number of relations connecting pairs of common nouns decreased from 31.5% to almost 10.2%. These observations show that the distribution of coreference data changes when new coreference relations are added. Once we have the expanded coreference data set that we were aiming for, we proceed with following three-step method for data analysis and derivation of coreference rules:

1. *Find the coreference knowledge satisfied by the largest number of anaphor-antecedent pairs.* We have observed that most of the coreference relations with a Proper Noun antecedent involve a repetition of the anaphor, a name alias, or a very similar expression. Also many relations satisfy agreements in number, gender and semantic class between the anaphor and its antecedent. Fewer relations involve semantic knowledge, such as synonymy.

2. *Develop coreference rules from the above observed knowledge.* We formalize the left-hand side of each coreference rule as a conjunct of the conditions that must be satisfied by the anaphor and its candidate antecedent. The right-hand side is always implemented by the function *Cast_in_Chain(Antecedent,Anaphor)*. The role of *Cast_in_Chain* is to check whether the antecedent already belongs to a coreference chain, and in this case a coreference relation is cast between the anaphor and the the closest textual expression from the coreference chain. The distance between two noun phrases NP_1 and NP_2 is measured by the function *Surface_Distance*, which counts the number of NPs when scanning the text in the following way:
   (a) if both NPs are in the same sentence, a function called *Surface_Search*[5] scans the sentence starting at NP_2, going towards NP_1 in a right-to-left order;
   (b) otherwise, *Surface_Search* scans the sentence containing NP_2 in a right-to-left order and then all the previous sentences in the normal left-to-right order, until it reaches NP_1.

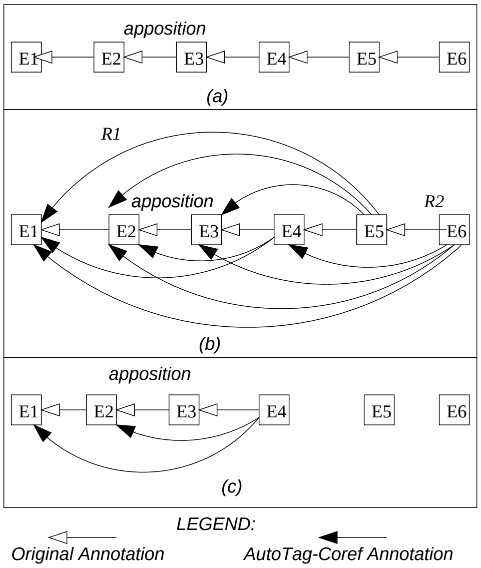


Figure 2: New coreference annotations and the effect of one filter. An example.

3. *Filter out the annotations that can be identified by the current coreference rules.* Whenever the conditions of a rule are satisfied, an antecedent for the anaphor is identified. Therefore no other coreference annotations sourced at the same anaphor are needed and they can be filtered out. After each filter is applied, the remaining annotations represent relations that are not identified by the current set of coreference rules. This tells us what kind of relations are more difficult to identify and, therefore, necessitate more than the minimalist amount of knowledge incorporated into the current set of coreference rules. Figure 2(a) illustrates the original annotations as opposed to the annotations expanded by AutoTag-Coref, represented in Figure 2(b). Figure 2(c) shows the remaining relations when coreference relations R_1 and R_2 from Figure 2(b) were identified by two different rules.

2.1 Coreference rules

Coreference knowledge is implemented as (1) the conditions from the left-hand side of a set of coreference rules; and also as (2) the conditions of the filters that eliminate the annotations discovered by the current set of rules. Since all rules have the same right-hand side, the are differentiated only by the conditions implemented. Consequently, coreference rules can be described by the three filters applied to the coreference data, as they implement the same conditions. Figure 3 illustrates three coreference rules,

[5] This search was implemented in the FASTUS system (cf. (Kameyama 1997)) and it is reported to model successfully the parse tree search proposed in (Hobbs 1978).

```
RULE-1-Filter-1-Pronoun (R1F1Pron)
  If (( Syntactic_Category(Anaphor)== Pronoun) AND  Repetition (Anaphor, Antecedent) )
    then  Cast_in_Chain(Antecedent, Anaphor)
RULE-1-Filter-1-Nominal (R1F1Nom)
  If (( Syntactic_Category(Anaphor)== Common Noun) AND  (Anaphor == Apposition(Antecedent) )
    then  Cast_in_Chain(Antecedent, Anaphor)
RULE-2-Filter-1-Nominal (R2F1Nom)
  If (( Syntactic_Category(Anaphor)==  Syntactic_Category(Antecedent)==Proper Noun) AND  Same-Category(Antecedent,Anaphor) )
     If  ( Category(Anaphor) == PERSON) AND (  Last_Name(Antecedent)==Last_Name(Anaphor) )  AND
                    AND (Gender(Antecedent) = Gender(Anaphor) AND  Surface_Distance(Anaphor,Antecedent)=min)
            then  Cast_in_Chain(Antecedent, Anaphor)
     If  ( Category(Anaphor) == ORGANIZATION) AND  Acronym(Anaphor,Antecedent))
            then  Cast_in_Chain(Antecedent, Anaphor)
```

Figure 3: Example of coreference rules.

the first one recognizes pronoun repetitions, the second one identifies coreference due to appositions and the third one is based on name alias identification. They correspond to the conditions applied by the first filter.

Filter 1

The first filter implements strong cohesion indicators, thus imposing high confidence in the coreference rules:
1) *repetitions* of the same expression (pronoun or nominal);
2) *appositions* and arguments of the same copulative verb (e.g. *be, become, make*);
3) *name alias* recognitions, comprising acronyms or forms of addressing people (e.g. *Bill Clinton* and *Mr. Clinton*);
4) the anaphor and the antecedent share *the same head* and have compatible adjuncts.

The applications of the coreference rules determined by the first filter eliminates 83% of the expanded annotations on the MUC-6 corpus, leaving only 1871 relations from the initial 11,690, whereas on the MUC-7 corpus, a reduction of 82% takes place, leaving only 2834 links from a total of 15,858 initial expanded relations.

Filter 2

The second filter uses weaker indicators of coreference, but makes the salience factor more relevant. Unlike filter 1, where the four conditions are applied disjunctively, the second filter imposes three conditions that must be satisfied concurently. The conditions are:
1) The anaphor and the antecedent share the same *semantic category.* A named entity tagger provides the semantic category information for proper names whereas WordNet defines the category of common nouns.
2) The anaphor and the antecedent *agree in number, gender and person.* The number information is provided by the part-of-speech tagger, the gender information is mined from WordNet with a method described in the next section.
3) No other text expression satisfying the above two conditions is at a smaller *Surface_Distance* from the anaphor.

The only semantic category considered by this filter is the PERSON category, recognized by (a) a named entity recognizer; (b) as a hyponym of the WordNet synset {*person, individual*}, or (c) as personal pronouns. Only the non-personal pronouns can corefer with any other noun category. An exception to the first condition is implemented in our system, by allowing the pronoun *they* to corefer also with any hyponym of the synset {*social group*}, comprising such nouns as *police*, *army* or *school.* The second filter reduces the number of unresolved relations to 521 in the MUC-6 data and to 767 in the MUC-7 data.

Filter 3

The third filter applies only to pairs of coreferring common nouns that are not PERSONS. The filter tests the semantic consistency of the coreference relation by using lexico-semantic information available from WordNet, as described in the next section. From the performance evaluation of other coreference resolution systems (e.g. (Kameyama 1997)), we know that more than 30% of the missed corereference links are due to the lack of semantic consistency information between an anaphoric noun and its antecedent noun. After this filter is applied, when comparing the number of original coreference annotations against the remaining annotations, we obtained a maximum recall of 91.3% for the MUC-6 data and of 88.7% for the MUC-7 data.

3 Knowledge Mining for Coreference Resolution

WordNet is used to acquire gender information for the agreement conditions and to mine patterns of semantic consistency between pairs of nouns.

Acquiring gender information from WordNet

We formalize the gender information through an expression G, which may be either an atomic expression, representing one of the gender attributes of a nominal, or a disjunct of two or three of them, as illustrated in Table 2. The gender attributes may have the values:

- m for masculine nouns;
- f for feminine nouns; and
- n for all the nouns that either are not from the Person category or are polysemous[6] and at least one of the senses does not belong to the Person category.

G	Noun examples	#
$m \vee f \vee n$	client, leader, neighbour	807
$m \vee f$	lawyer, loser, patron, newborn	5217
$m \vee n$	king, antique, father	42
$f \vee n$	maiden, mezzo, nanny, harpy	81
m	groom, housefather, nobleman	208
f	woman, daughter, bride, sheika	417

Table 2: Distribution of gender information.

Gender attributes are assigned by the two following heuristics:

Heuristic 1 If a collocation fom a WordNet synset contains the word *male*, the expression G for the whole sysnet is m. If the collocation contains the words *female* or *woman*, $G{=}f$.

Heuristic 2 Consider the first four words from the synset gloss. If any of the gloss words have been assigned gender information, propagate the same information to the defined synset as well.

Each hyponym of the concept {*person, individual, human*}, categorized as Person has expression G initialized to $f \vee m$, since all lexemes represent persons, that can be either males or females. Whenever one of the two heuristics previously defined can be applied at any node S from this subhierarchy, three operations take place:

▷ *Operation 1:* We update G with the new expression brough forward by the heuristic.

▷ *Operation 2:* We propagate all the expression to the hyponyms of S;

▷ *Operation 3:* We revisit the whole Person subhierarchy, in search for concepts D that are defined with glosses that use any of the words from synset S or any word from any of its hyponyms. Whenever we find such a word, we update its G expression to $G(S)$. We also note that many words are polysemous, thus a word w may have multiple senses under the Person sub-hierarchy and moreover, each sense might have a different G expression. In this case, all words from the synsets containing w receive the disjunct of the gender attributes corresponding to each sense of w.

Mining semantic information from WordNet

We used the WordNet knowledge base to mine *patterns of WordNet paths* that connect pairs of coreferring nouns from the annotated chains. The paths are combinations of any of the following WordNet relations:

- Synonym connecting all elements of a synset;
- Is-A connecting nouns and verbs from the same hierarchies. We also consider the reversed Is-A relation, denote rIs-A;
- Gloss connecting any element of a synset with the genus of its glossed definition. We also consider its reverse relation, named Defines;
- In-Gloss connecting any element of a synset with one of the first four words of its glossed definition. We also consider its reversed relation, named In-Definition
- Has-Part connecting a concept to its meronyms. We also consider the reversed Is-Part relation;
- Morpho-Derivation connecting a word to its morphological derivations.
- Collide-Sense connecting several senses of the same word.

To determine the *confidence* of the path we consider three factors:

◇Factor f_1 has only two values. It is set to 1 when another coreference chain contains elements in the same NPs as the anaphor and the anetcedent. For example, if NP_1 is *" the professor's son"* and NP_2 is *"his father"*, the semantic consistency between *father* and *professor* is more likely, given that *his* and *son* corefer. Otherwise, f_1 is set to 0.

◇Factor f_2 favors (a) relations that are considered "stronger" (e.g. Synonymy, Gloss); and (b) shorter paths. For this purpose we assign the following weights to each relation considered: w(Synonym) = 1.0; w(Is-A) = 0.9; w(Gloss) = 0.9; w(In-Gloss) = 0.3; w(Has-Part) = 0.7; w(Morpho-Derivation) = 0.6; and w(Collide-Sense) = 0.5. When computing the f_2 factor, we assume that whenever at least two relations of the same kind repeat, we should consider the sequence of relations equivalent to a single relation, having the weight devided by the length of the sequence. If we denote by nr_{rel} the number of different relation types encountered in a path, and $nr_{same}(rel)$ denotes the number of links of type rel in a sequence, then we define f_2 with the formula:

$$f_2 = \frac{1}{nr_{rel}} \sum_{rel \in Path} \frac{w(rel)}{nr_{same}(rel)}$$

◇Factor f_3 is a semantic measure operating on a conceptual space. When searching for a lexico-semantic path, a search space SS is created, which contains all WordNet content words that can be reached from the candidate antecedent or the anaphor in at most five combinations of the seven relations used by the third filter. We denote by N the total number of nouns and verbs in the search space. C represents the number of nouns and verbs that can be reached by both nominals. In addition nr_{total} is the number of concepts along all paths established, whereas

[6] A polysemous noun has multiple semantic senses and therefore has multiple entries in the WordNet dictionary.

nr_{path} is the number of concepts along the path with the best scoring f_2. The formula computing f_3, inspired by Salton and Buckley's *tf-idf* weighting scheme (Salton and Buckley 1988), is:

$$f_3 = 0.5 + \frac{0.5 \times nr_{path}(SS)}{nr_{total}(SS)} log \frac{C}{N}$$

The confidence measure of the path, denoted by R, combines all three factors in a way similar to van Rijsbergen's *E-Formula* (van Rijsbergen 1979), used for evaluating the performance of Information Retrieval systems. The formula that computes the confidence value of a paths, R, is:

$$R = \frac{1 + b^2}{\frac{b^2}{f_3} + \frac{1}{f_2}} + \frac{(b^2 - 1)}{b^2} f_1$$

The selection of the value b plays an important role in the overall performance of COCKTAIL. Since we are more interested in the precision of the lexico-semantic path than in the recall of all possible paths, we select $b = 2.7$. Table 3 lists some of the patterns determined on the training data and their confidence factors.

Noun↔Noun	Pattern	R
helicopter↔chopper	IS-A	0.92
site↔terrain	IS-A: RIS-A	0.76
concern↔maker	SYNONYM:GLOSS	0.84
regime↔government	SYNONYM	1.0
beacon↔signal	IN-GLOSS:USE-GLOSS	0.72

Table 3: Patterns of semantic consistency.

4 Combining Coreference Rules

The order in which coreference rules are applied is very important, since sometimes, for the same anaphor, different antecedents are indicated by different coreference rules. One solution is to use the same order in which coreference rules have been devised. This order gives preference to proper noun antecedents over pronominal antecedents or common noun antecedents. Such an order determines what rule should be applied when several candidate antecedents are identified.

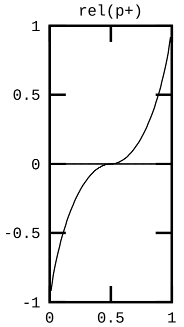


Figure 4: A function of coreference confidence.

An alternative is not to use a predefined order, but to find for each anaphor what rule should be applied such that the resulting coreference chains are as precise as possible. For this purpose, for each rule R_i from the set of coreference rules $\mathcal{R}$ =$\{R_1, R_2, ..., R_n\}$ we compute p, the number of times when the application of rule R_i in the training corpus is correct, and n, the number of times when the application of rule R_i was not correct. This allows us to define the *confidence* of using R_i for establishing coreference between two noun phrases NP_j and NP_k by using the formula:

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

where the entropy measure is defined as:

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

The rationale for $rel(R_i, NP_j, NP_k)$ is given by the fact that the entropy indicates how much information is still needed to establish the coreference between NP_j and NP_k with certainty. As illustrated in Figure 4, if $p_+(R_i) = \frac{p}{p+n}$ then the closer $p_+(R_i)$ is to 1, the more confidence we have in the coreference relation between NP_j and NP_k. Similarly, the closer $p_+(R_i)$ is to 0, the more confident we are that NP_j and NP_k do not corefer.

The confidence measure of each rule is used in determining the most precise coreference chains spanning a text. Given a text $\mathcal{T}$ we consider all its referential expressions $\mathcal{RE}(\mathcal{T})$=$\{NP_1, NP_2, ..., NP_m\}$, a subset of the text noun phrases. To derive the coreference chains spanning the elements from $\mathcal{RE}(\mathcal{T})$ we use the set of coreference rules $\mathcal{R}$. A given application of the rules from $\mathcal{R}$ generates a partition on $\mathcal{RE}(\mathcal{T})$. Each partition is a set of coreference chains Par=$\{CC_k^{Par}\}$ such that each $NP_j \in \mathcal{RE}(\mathcal{T})$ belongs to one and only one of the coreference chains CC_k^{Par}. Each partition corresponds a possible combination of coreference chains spanning $\mathcal{RE}(\mathcal{T})$. If $\mathcal{P}(\mathcal{RE})$ are all the possible partitions on $\mathcal{RE}(\mathcal{T})$, our goal is to find the best partition, i.e. the partition that contains all the correct coreference chains established on $\mathcal{RE}(\mathcal{T})$. If every partition $Par \in \mathcal{P}(\mathcal{RE})$ is assigned a measure $m(Par, \mathcal{R})$ which computes the likelihood that Par contains all the correct coreference links from the text $\mathcal{T}$, established by the rules from $\mathcal{R}$, then the best partition is given by:

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

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

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

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

$$m^{+}(Par,\mathcal{R}) = \sum_{R_i \in \mathcal{R}} rel(R_i, NP_j, NP_k)$$

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

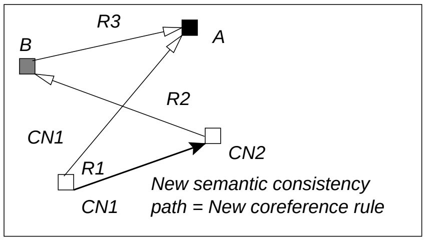
$$m^{-}(Par,\mathcal{R}) = \sum_{R_i \in \mathcal{R}} -rel(R_i, NP_j, NP_k)$$

Learning method

At training time, for each R_i from $\mathcal{R}$ we compute the *entropy*(R_i) on the training corpus. At testing time, given a new, test text $\mathcal{T}^T$ and its referential expressions $\mathcal{RE}(\mathcal{T}^T)$ =$\{NP_1^T, NP_2^T, ..., NP_t^T\}$, we find the best partition by using a local search algorithm, namely by applying hillclimbing to the values of the $m(Par^i,\mathcal{R})$ measure for each possible partition of $\mathcal{T}^T$. The initial partition is Par^0 =$\{\{NP_1^T\},\{NP_2^T\}, ..., \{NP_t^T\}\}$, consisting of t distinct coreference chains, each containing a single referential expression. The other partitions are generated in a recursive manner. For each partition Par^i, with $1 \leq i \leq 2^t$, the new partitions Par_j^{i+1} are generated by combining any pairs of chains from Par^i. If Par^i has k chains, then $k(k+1)/2$ new partitions can be generated and ordered according to their m measures. At each step of the process of generating a new partition, the hillclimbing algorithm selects the best partition Par_j^{i+1}, having the highest m score. When $m(Par^{i+1},\mathcal{R}) < m(Par^i,\mathcal{R})$ the search terminates, since the maximum of the m measure has been reached. However, this is often a local maximum. To avoid local maxima, instead of selecting only the best scoring partition, we consider all the first p paritions, where p is called the *patience* of hillclimbing. In our case, we chose $p = 5$.

5 Bootstrapping for Coreference Resolution

We considered *bootstrapping*, the new machine learning technique presented in (Riloff and Jones 1999), as an ideal vehicle for enhancing the semantic consistency constraints between common nouns. Boostrapping is known to operate on unlabeled data by using only some knowledge seeds. The rules implemented in *COCKTAIL* do not capture all the coreference patterns, but they are fairly precise, thus they can be viewed as the knowledge seeds for bootstrapping. When applied to new, unlabeled texts, the coreference rules from *COCKTAIL* discover new pairs of common nouns that might corefer. For example, Figure 5(a) illustrates the application of three coreference rules on a new text. Two anaphors CN_1 and CN_2 are common nouns and no semantic consistency information accounts for their coreference. However, if the antecedent of anaphor CN_1 is sought, coreference rule R_1 indicates expression A to be the antecedent. Similarly, the antecedent of anaphor CN_2 is discovered by rule R_2 as being expression B, which corefers with A, because of rule R_3. When the coreference chain is built, expression CN_1 is directly linked to expression CN_2, thus enabling new semantic consistency information discovered from WordNet paths. Figure 5(b) illustrates one of the paths that were discovered and its corresponding coreference rule.



(a)

Semantic consistency Path:
Mropho-Derivation : Is-A : Collide-Sense

```
Coreference Rule:
  If (x is Morpho-Derivation ( Anaphor) ) AND
      AND (y is one of the hypernyms of x) AND
      AND (z is SYNONYM of y) AND
      AND (z is SYNONYM of anaphor)
            then Cast_in_Chain(Anaphor,antecedent)
```

(b)

Figure 5: Bootstrapping new rules.

As a rule of thumb, we do not consider a new coreference rule based on semantic consistency information unless coverage in the data warrants it. At training time, we selected N path patterns because the majority of the paths matching these patterns had the relevance larger than a threshold, t_R. Whenever new semantic consistency is uncovered by a path $Path_{new}$, its relevance $R(Path_{new})$ must be larger than the threshold t_R. However, the new paths might not have been encontered in the training data and still encode relevant semantic consistency information. To account for this case, a new value for the t'_R is selected, which determines a different entropy for each coreference rule based on semantic consistency constraints. Consequently, a different set of coreference chains is generated for each training text, thus changing both the precision and the recall of *COCKTAIL*. This mechanism of discovering and adding new rules to the set of coreference rules enables the following bootstrapping algorithm:

```
◇Generate all candidate Paths from new texts
MUTUAL BOOTSTRAPPING LOOP
1. Score all candidate paths by their relevance
2. Add the best candidates and encode them as rules
3. Adjust the relevance threshold
4. Goto step 1 if the F-measure did not degrade
            under MIN_Performance
```

(Riloff and Jones 1999) note that the performance of the mutual bootstrapping algorithm can deteriorate rapidly if erroneous rules are entered. To make the algorithm more robust we use the same solution by introducing a second level of bootrapping. The outer level, called *meta-bootstrapping* identifies the most reliable k rules based on semantic consistency and discard all the others before restarting the mutual bootstrapping loop again. In our experiments we have retained only those rules for which the new performance, given by the F-measure was larger than the median of the past four loops. The formula for the van Rijsbergen's F-measure combines the precision P with the recall R in $F = \frac{2\times P\times R}{P+R}$.

6 Evaluation

To measure the performance of COCKTAIL we have trained the system on 30 MUC-6 and MUC-7 texts and tested it on the remaining 30 documents. We computed the *precision*, the *recall* and the *F-measure*. The performance measures have been obtained automatically using the MUC-6 coreference scoring program (Vilain et al. 1995). Table 4 lists the results.

	Precision	Recall	F-measure
COCKTAIL rules	87.1%	61.7%	72.3%
COCKTAIL rules combined	91.3%	58.6%	71.8%
COCKTAIL +bootstrapping	92.0%	73.9%	81.9%

Table 4: Bootstrapping effect on COCKTAIL

Table 4 shows that the seed set of rules had good precision but poor recall. By combining the rules with the entropy-based measure, we obtained further enhancement in precision, but the recall dropped. The application of the bootstrapping methodology determined an enhancement of recall, and thus of the F-measure. In the future we intend to compare the overall effect of rules that recognize referential expressions on the overall performance of the system.

7 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 rules through massive training data, generated by its AUTOTAG-COREF component. Furthermore, by using an entropy-based method we determine the best partition of corefering expressions in *coreference chains*. New rules are learned by applying a bootstrapping methodology that uncovers additional semantic consistency data.

References

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

Susan E. Brennan, Marilyn Walker Friedman and Carl J. Pollard. 1987. A centering approach to pronouns. In *Proceedings of ACL-87*, pages 155-162.

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

Barbara J. Grosz, Aravind K. Joshi and Scott Weinstein. 1995. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2).

Lynette Hirshman, Patricia Robinson, John Burger and Marc Vilain. 1998. The role of Annotated Training Data. Unpublished manuscript.

Jerry R. Hobbs. 1978. Resolving pronoun references. *Lingua*, 44:311–338.

Shalom Lappin and Herbert Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–562.

Megumi Kameyama. 1997. Recognizing Referential Links: An Information Extraction Perspective. In *Proceedings of the ACL'97/EACL'97 Workshop on Operational factors in practical, robust anaphora resolution*, pages 46–53, Madrid, Spain.

Christopher Kennedy and Branimir Bogureav. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of COLING-96*.

George A. Miller. 1995. WordNet: A Lexical Database. *Communication of the ACM*, 38(11):39–41.

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

Ellen Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of 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 AAAI-99*.

Gerald Salton and Chris Buckley. 1988. Term-weighting approaches in automatic retrieval. *Information Processing and Management*, 24(5):513–523.

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

C.J. van Rijsbergen. 1979. *Information Retrieval*. Butterworths, Glasgow, UK.

Bonnie Webber. Discourse deixis: Reference to discourse segments. In *Proceedings of ACL-88*, pages 113–121.

Janyce M. Wiebe, Rebecca F. Bruce and Thomas P. O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of ACL-99*, pages 246-253.