Coreference Resolution Survey

Emili Sapena, Lluís Padró, Jordi Turmo
TALP Research Center
Universitat Politècnica de Catalunya
Barcelona, Spain
{esapena, padro, turmo}@lsi.upc.edu

December 4, 2008

Abstract

A vast ammount of unstructured information resides in documents in natural language that can be found in many sources such as World Wide Web, news articles, e-mails and so on. Discovering new knowledge analyzing these text resources is the goal of Text Mining, which is an interdisciplinary field that borrows techniques from Data Mining and Natural Language Processing (NLP).

Coreference resolution is a NLP task which consists of determining the noun phrases and pronouns in a text or discourse that refer to the same entity. The research in coreference resolution has a direct effect on the field of Text Mining and its related NLP areas that need a discourse interpretation such as Information Extraction, Question Answering, Summarization, Machine Translation and so on. Furthermore, in order to *understand* a text document or even a speech, it is mandatory to resolve its coreferences.

This survey is an extended summarization of state of the art of coreference resolution. The key concepts related to coreference and anaphora are presented, the most relevant approaches to coreference resolution are discussed, and existing systems are classified and compared. Finally, the evaluation methods shared by researchers in the area and the commonly used data sets corpora are presented and compared.

1 Introduction

The need for managing the information that resides in the vast amount of documents in natural language is becoming more important. Documents in natural language, as opposed to information stored in databases, are characterized by their unstructured nature. Sources of such unstructured information include the World Wide Web, governmental electronic repositories, news articles, blogs repositories, e-mails, and so on.

Text Mining is the data analysis of text resources where new knowledge is discovered (Hearst, 1999). It is an interdisciplinary field that uses techniques from the general fields of Data Mining and Natural Language Processing, combining methodologies from Information Extraction (IE) (Turmo et al., 2006), Information Retrieval (IR) (Faloutsos and Oard, 1995), Computational Linguistics (Mitkov, 2003), Categorization (Sebastiani and Ricerche, 2002) and Summarization (Mani, 2001). Research in this areas deals with many issues originating from natural language particularities. And the research in natural language tasks, such as coreference resolution, has a significant impact on all these areas.

Coreference resolution is a natural language processing (NLP) task which consists of determining the noun phrases and pronouns in a text or discourse that refer to the same entity. It has a direct effect on the field of Text Mining and its related areas that need a discourse interpretation such as Information Extraction, Question Answering, Summarization, Machine Translation and so on. Furthermore, in order to *understand* a text document or even a speech, it is mandatory to resolve its coreferences.

Coreference resolution is considered a hard and important problem, and a challenge in artificial intelligence (AI). The necessary knowledge to resolve coreferences is not only lexical and syntactic, but also semantic and pragmatic, which implies to go deep in many levels of natural language comprehension.

This survey covers the state of the art on coreference resolution and part of the strongly related task of anaphora resolution. The document is divided in six sections. First, Section 1 is an introduction to the concepts of coreference and anaphora, and an explanation of the most addressed resolution tasks: Direct Coreference and Pronominal Anaphora. Section 2 shows the main steps followed by a general coreference resolution system including some preprocessing issues such as mention detection and previous filtering. An extended summarization of most of the relevant approaches for coreference and anaphora resolution is divided in two sections: Knowledge-based and Corpus-based approaches. On one hand, Section 3, is mainly focused in approaches based on linguistic and cognitive theories for anaphora resolution. Most of the knowledge-based works are hand-written heuristics that do not use machine learning neither annotated corpora. Moreover, most of the knowledge-based approaches rely on previous morphological and syntactic manual analysis of the documents. On the other hand, Section 4 summarizes the approaches based on annotated corpora and is divided in three subsections: Statistical and Manual approaches, Supervised learning, and Weakly supervised and Unsupervised learning. All three subsections include approaches focused on coreference resolution and, most of them, based on an automatic preprocess. Section 5 reviews the corpora and evaluation methods used by most of the coreference resolution systems. Finally, Section 6 is a conclusion reviewing the evolution of coreference resolution approaches so far, and the directions that researchers may follow in this area.

1.1 Coreference

Coreference resolution is the task of determining which *mentions* in a discourse refer to the same entity. A **mention**, normally a noun phrase (NP), is a **referring expression** having an entity as a referent. **Coreference chains** are groups of referring expressions having the same referent. It means, a coreference chain is formed by all the mentions in a discourse that refer to the same entity. The goal of a coreference resolution system is to find coreference chains given an arbitrary text as input.

The example in Figure 1 shows a coreference chain about the entity *”Sergio Aguero”* in a newspaper article:

> Atletico Madrid president Enrique Cerezo has warned Chelsea off star striker **Sergio Aguero.**
>
> Aware of Chelsea owner Roman Abramovich’s interest in **the young Argentine**, Cerezo said last night: ”I will answer as always, **Aguero** is not for sale and we do not want to let **him** go.”

Figure 1: Example of a coreference chain.

As one can see in Figure 1, in order to resolve the coreference chain of *”Sergio Aguero”* several knowledge of diverse nature must be used. First, **morphological** and **syntactic** analysis is required in order to detect all the pronouns, named entities and any other noun phrase referring to some entity. In other words, morphological and syntactic information is needed to detect mentions. **Lexical** procedures are also needed to decide that mentions like *”Sergio Aguero”* and *”Aguero”* might be alias referring to the same entity. However, **semantic** information would be useful to decide that both are persons. Furthermore, **pronoun resolution** should be used to link *him* with *”Aguero”*. In addition, knowledge about **discourse coherence** might be useful to discard possibly missmatches. Finally, **world knowledge** is kind of essential if one wants to add *the young Argentine* in the coreference chain.

1.2 Anaphora vs Coreference

Anaphora, *Cataphora* and Coreference resolution are close but different problems. The goal of **anaphora resolution** is to identify an antecedent for each noun phrase (NP) that depends on other NPs for its interpretation. Specifically, **anaphora** is the linguistic phenomenon of pointing back to a previously mentioned item in the text. The “pointing back” word or phrase is called **anaphor** and the expression to which it refers or for which it stands is its **antecedent**. In the case when the “anaphor” is pointing forward, it means the “anaphor” is found in the text before the “antecendent”, it is called **cataphora** (Mitkov, 2002). Cataphora phenomenon is not as common as anaphora.

It is important to emphasize that a pair of anaphoric items do not need to be coreferential. In some cases, an anaphor is pointing back to its antecedent but they are not referring to a concrete entity.

> **Every dog** has **its** house.

In this example, "its" is the anaphor of "every dog" but "every dog" is not referring to a specific entity.

Although coreference chains are normally composed by some anaphoras and perhaps some cataphoras, there are also other possibilities: some mentions may be coreferential without being anaphoric, as in the following example:

> Some people say **Barcelona** is a good place to visit. I think I'm going to **Barcelona** this summer.

Figure 2: Coreferential mentions do not need to be anaphoric

In this case (Figure 2), "Barcelona" is not an anaphor because there is no dependency on other NP for its interpretation. However, both mentions of "Barcelona" form a coreference chain.

1.3 Kinds of Coreferences and Anaphoras

Anaphoras and Coreferences can be classified following diverse criteria. In the case of anaphoras, they are usually classified depending on:

- Lexical form of the anaphor:
  - **Pronominal anaphora:** The anaphor is a pronoun
  - **Nominal anaphora:** The anaphor is a pronoun, a proper name or a definite NP and the antecedent is a non-pronominal NP.
  - **Verb anaphora:** The anaphor is a verb (*Alice smiled, as did Bob*)
  - **Adverb anaphora:** The anaphor is an adverb (*We are going to the station to meet you there*)
  - **Zero anaphora:** The anaphor is omitted (*Amy had made a special effort that night but @ was disappointed with the results*). Where @ should be 'she' but is omitted.
  - **One-anaphora:** The anaphor is the word *one* (*He fell from the bicycle two times yesterday, and another one today*).
- Location: Anaphor and antecedent can be in the same sentence (**intrasentential**) or in different ones (**intersentential**)

- Type of identity: Anaphor and antecedent are coreferential (**identity-of-reference**) or not (**identity-of-sense**)
- Type of antecedent: Noun phrase, noun anaphora (head noun or nominal group, but not a NP), verb or verb phrase, clause, sentence, sequence of sentences or coordinated antecedents (for example *"Bob and Charlie.... both...."*)

Nominal (which includes pronominal) anaphora is the kind of relation found in coreference chains when its type is identity-of-reference. In the case of coreference and also for nominal anaphora there are two main relation classes:

- Class of coreference relation:
  - **Direct:** identity (*Mike W. Smith* ⇔ *Smith, M.*), synonymy (*baby* ⇔ *infant*), generalization and specialization (*car* ⇔ *vehicle*).
  - **Indirect:** (a.k.a. associative or bridging): part-of (*wheel* ⇔ *car*), set membership (*Gringo Starr* ⇔ *Beatles*)

1.4 Most Frequently Addressed Tasks

There are many types of coreferences and anaphoras as we have seen in section 1.3. However, two of them are the most frequently addressed problems: direct coreference and pronominal anaphora.

Direct coreference is the kind of coreference addressed by most of the coreference resolution systems. Concretely, their main goal is to discover all the entities referred in a document and the chain of mentions to them. Until the first appearing of an annotated corpus and an international contest motivating research in this task (MUC-6, 1995) the computational linguistic community was mainly focused in anaphora resolution. With MUC-6, a wave of new interest in coreference resolution appeared with the challenge of applying machine learning to such a new task. Therefore, most of the newest developed systems, from middle 90s until today, are corpus-based coreference systems, while most of the prior works before 90s are knowledge-based anaphora resolution systems.

Most of the anaphora resolutions researchers have focused their efforts in the task of pronominal anaphora resolution (also named **pronoun resolution**). Other kinds of anaphora tasks like verb or adverb anaphora, or the ones with complex antecedents such as sentences or sequence of sentences are more complicated. Anaphora resolution has been a research topic widely studied during decades, and it still being an interesting problem since it is far from being solved.

2 Coreference Resolution Systems

A coreference resolution system gets a plain text as input and returns the same text with coreference annotations as output. This section describes a generic process of a coreference resolution system. It introduces the different phases of the preprocessing and resolution process and explains the issues and difficulties of each step. Most of the existent coreference resolution systems can be considered instantiations of this general process that consists of four steps. First, it is needed a text processing in order to identify the mentions (step 1) and characterize them (step 2). Next, the system evaluates the coreferentiality of each pair or group of mentions (step 3). And finally, coreference chains are formed (step 4). Following subsections describe each one of these steps.

2.1 Identification of mentions

In order to identify the mentions from the plain input text, it is required a preprocess. At least, a part of speech tagger and a NP chunker. Each noun phrase (NP) is considered a mention when resolving direct coreference, and also proper names, named entities and pronouns[1]. Nested NPs may also be mentions. However, annotating everything produces duplicates. For example:

(The company) of ((his) father) in (Michigan)
NP: The company of his father in Michigan
NP: The company of his father
NP: The company
NP: his father
Pronoun: his
Proper name: Michigan

In the example, we find 6 possible mentions where only 4 are necessary. In the case of nested NPs, an acceptable solution is to discard the longest NPs when they share the head. However, any of them are referring to the same real entity (the company), so any of them should be valid, but only one.

Each published coreference system uses its own preprocess pipeline. The diversity of preprocesses used in different systems makes difficult the comparison of those systems performances. Even when they use the same corpus and measure, a different preprocess may have a large influence in final results.

Some researchers prefer to center their efforts in coreference chains resolution avoiding the step of identification of mentions. To do that, they use *true mentions*, it means, the annotated mentions for train or test proposes. In this cases, the system knows that every mention in the input has to be assigned to some coreference chain in the output, while systems without true mentions

[1] Interrogative pronouns (What, Where, etc.) are not considered mentions

will need to discard lots of non-coreferential mentions in further steps. Consequently, the difficulty of the task largely decreases. However, a system based on true mentions can not be applied in real situations because, actually, there is no way to distinguish coreferential mentions from non-coreferential ones without coreference resolution.

2.2 Characterization of mentions

Depending on the information required for further resolution, several processing steps are applied such as parsing, chunking, word sense disambiguation, named entity recognition and classification, semantic role labeling and so on. Some filters such as pleonastic pronouns and anaphoric NPs may also be applied to discard some mentions before resolution.

Sometimes, some preprocessing tasks like named entity recognition, syntactic parsing and others, are assumed as *perfect* in order to avoid carring along processing errors in the pipeline and obtain bad results because of that. In this cases, gold standard syntactic parsing or NE are used.

In this phase, some models opt for filtering mentions before resolution in order to improve system precision and maybe decrease computational costs. Two of the most used filters are pleonastic pronouns and non-anaphoric mentions.

Pleonasm is the use of more words than necessary to express an idea clearly. In English, the pronoun *"It"* sometimes acts as a **pleonastic pronoun** because does not refer to any entity but it is grammaticaly necessary. It is usually found in temporal and meteorological expressions, but can also occur in other sentences:

It is raining.
It is four o'clock.
It is fine.
It is okay.

There are other cases where some authors consider pronoun *"it"* as pleonastic (Mitkov, 1999).

It seems that...
It is known that...
It is important to note that...

However, these cases may be considered cataphoric with a fact, usually introduced by *"that"*. There are also other cases where a pronoun is not anaphoric, for instance, the use of *you* referring to the reader.

Filtering pleonastic pronouns before resolution avoids further missclassifications and makes the task easier for the resolution algorithm which can always consider pronominal mentions as anaphoric.

On a different matter, many coreference resolution systems proposes previous filtering of non-anaphoric NPs in order to facilitate further coreference chains classification. It may be useful in anaphora resolution. However, when resolving coreferences, a coreferential NP does not need to be anaphoric as it has been shown in Figure 2. Consequently, it is not clear the utility of this kind of filtering.

2.3 Classification of Candidates

The real coreference resolution starts at the Classification step where the system evaluates the *coreferentiality* of each pair or group of mentions. Most of the coreference resolution systems consists of a pairwise classifier where each candidate pair of mentions is classified as COREFERENTIAL or NON-COREFERENTIAL, normally with a confidence value or an associated probability. This information is used in final step to form definitive coreference chains.

This classification step has several variations depending for example on the order followed to classify the pairs, the algorithm used for classification or the information used about candidate mentions. A pair of mentions is evaluated using the information gathered in previous steps. A set of heuristics that evaluate the compatibility of that pair in some criteria (a set of features) is used by the classification algorithm to evaluate their coreferentiality.

2.4 Formation of Chains

Apart from some sophisticated systems that directly evaluate the compatibility of group of mentions, normally, after pairwise classification a final step of formation of chains is required. Many works simply link each pair of mentions classified as COREFERENTIAL avoiding possible contradictions, like single-link in clustering algorithms. For example, if the pair of mentions A and B and also the pair B and C have been classified as COREFERENTIAL, then the chain A-B-C is formed independently of the classification of the pair A and C. Other systems take advantage of the probabilities obtained in Classification step to find the best possible coreference chains, using algorithms of different nature such as bell-tree or graph partitioning.

The following two sections summarize the state of the art of two approach branches: knowledge-based and corpus-based. First one, knowledge-based approaches, describes systems developed mainly for the resolution of anaphora without annotated corpora. Anaphora resolution does not require this last step (formation of chains) of the generic algorithm. Moreover, in most cases, steps

1 and 2 are partially skipped because the system relies on previous parsing. The second section, corpus-based approaches, describes systems that fit in the generic algorithm, in spite of some ones that mix steps 3 and 4.

3 Knowledge-based Approaches

This section reviews approaches based on a set of hand-written heuristics that do not use machine learning neither an annotated corpus. Most of them were made between 70s and 90s when no annotated corpora were available. Moreover, the task which most of this approaches were developed to is pronominal anaphora resolution, but not coreference. Notwithstanding, some of these pioneer works showed interesting ways to follow that still are a reference nowadays for anaphora resolution and also for coreference resolution.

This section is divided in three subsections that represent the three most relevant families of knowledge-based approaches. First, the ones focused on sentences, browsing parsed trees and looking for antecedents using morphological and syntactic information. Second, approaches based on cognitive theories about discourse. These works take advantage of discourse rhetorical structure and resolve pronouns assuming discourse coherence. And third, approaches that satisfy constraints combining different kinds of information.

3.1 Based on Parsed Tree

3.1.1 Hobbs' Algorithm

Hobbs' Algorithm (Hobbs, 1977) defines a set of steps to follow in order to resolve anaphoric pronouns. Starting in the pronoun's node of the syntactic tree of a sentence, a breadth-first left-to-right search is done taking care of some conditions when a NP is found. If an antecedent is not found in the same sentence, then the search continues in the previous sentence of the document, and so on. Every time a NP is found, it has to agree in number and gender with the pronoun to be proposed as antecedent.

An example is shown in Figure 3. It shows the process followed by Hobbs' Algorithm to find that "the residence of the king" is the antecedent of "it" in the sentence *"The castle in Camelot remained the residence of the king until 536 when he moved it to London"*. The order followed traversing the parsed tree is shown in Figure 4

Hobbs' Algorithm achieves precision performance about 90% finding antecedents for anaphoric pronouns. Pleonastic pronouns are manually filtered before resolution and the syntactic parsed tree of sentences is complete and always correct (not automatic). Therefore, this high precision results should be seen as ideal. However, it indicates that, once pleonastic pronouns are filtered,

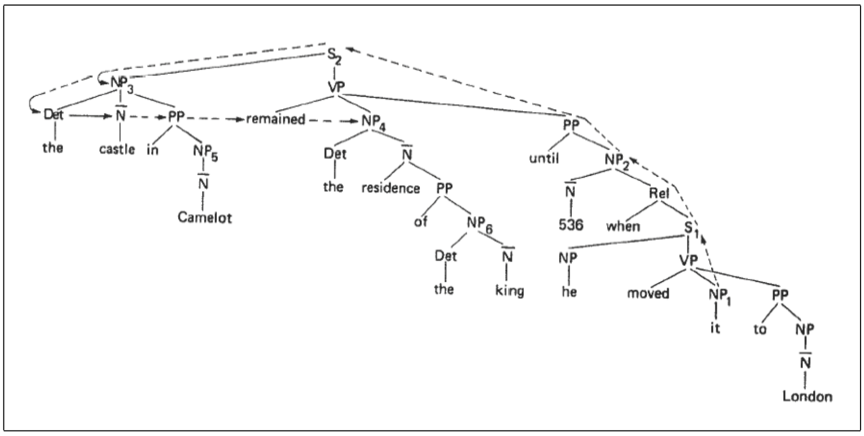

Figure 3: Example of search done by Hobbs' Algorithm finding that "the residence of the king" is the antecedent of "it" in the sentence *"The castle in Camelot remained the residence of the king until 536 when he moved it to London"*.

about 90% of pronouns in a document can be resolved only with morphological and syntactic information.

Another important finding of Hobbs' work was that 98% of pronominal anaphora antecedents are in the same sentence of the pronoun or in the previous one. Therefore, reducing the search scope only to two sentences, a system may improve its precision with a slight loss of recall.

3.1.2 Binding Theory

The Binding theory is part of Principles and Parameters Theory (Chomsky, 1981) and imposes important syntactic intrasentential constraints as to how NPs may corefer. It is helpful in determining impossible antecedents of pronominal anaphors and in assigning possible antecedents to reflexive pronouns. Some of the constraints defined there have been used for automatic anaphora resolution (Ingria and Stallard, 1989; Carvalho, 1989).

The Binding Theory cares about the interpretation of reflexives, pronouns and lexical NPs, formulating an important syntactic constraint for each case. All three constraints use the structural relation of **c-command** which must be introduced first. Given a syntactic tree of a sentence, a node A c-commands a node B if and only if (Haegeman, 1994):

1. A does not dominate B
2. B does not dominate A
3. the first branching node dominating A also dominates B.

Hobbs' Algorithm

1. Begin at the NP node immediately dominating the pronoun
2. Go up the tree to the first NP or S node encountered. Call this node X, and call the path used to reach it p.
3. Traverse all branches below node X to the left of path p in a left-to-right, breadth-first fashion. Propose as the antecedent any NP node that is encountered which has an NP or S node between it and X.
4. If node X is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as antecedent. If X is not the highest S node in the sentence, continue step 5.
5. From node X, go up the tree to the first NP or S node encountered. Call this new node X, and call the path traverse to reach it p.
6. If X is an NP node and if the path p to X did not pass though the N-bar node that X immediately dominates, propose X as the antecedent.
7. Traverse all branches below node X to the *left* of path p in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.
8. If X is an S node, traverse all branches of node X to the *right* of path p in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent.
9. Go to step 4

Figure 4: Traversal order followed to propose antecedents for a pronoun if they agree in gender and number.

The key constraints introduced in Binding Theory use the c-command relation, grammatical conditions and the concept of **local domain**. Local domain conceptually refers to an immediate context, including the current sentence, were short-distance anaphors may occur. Following, the three key constraints of Binding Theory are listed:

- **A. Reflexives:** A reflexive anaphor must be c-commanded by its antecedent and they must agree in person, gender and number.
- **B. Pronouns:** A pronoun cannot refer to a c-commanding NP within the same local domain.
- **C. NPs:** A non-pronominal NP cannot corefer with an NP that c-commands it.

Constraints B and C are specially useful in order to discard antecedents.

3.2 Based on Rhetorical Structure

A discourse is divided in a set of complete units called **utterances**. An utterance is typically larger than a single sentence, but smaller than the complete discourse. The structure of a discourse offers information about the message that the speaker wants to transmit. On one hand, redundancy, coherence and agreement in the discourse allows the listener to better understand the meaning of the received words. On the other hand, some pronouns, ellipsis, appositions and others are used due to the speaker laziness.

The theories described in this section and their implementations studied this cognitive and linguistic phenomena and take advantage of it in order to resolve pronouns.

3.2.1 Centering and Focusing

Centering (Grosz et al., 1983) is a theory about discourse coherence. It is based on the idea that speaker/writer intention is to keep main entity in focus which entails some uses of referring expressions. During a discourse, the center (the main entity) usually shifts softly and it does not tend to occur when using anaphoric pronouns referring to it. It means, when looking for antecedents of pronouns, it is plausible to assume that center is not changing from previous utterances. In the contrary, when the center changes, it has to be easily inferenced by the listener/reader in a coherent discourse. So, definite NP or other type of clues are usually used in this case.

The theory defines a set of rules and constraints to determine the center and its changes across subsequent pairs of utterances. Each utterance U has a single *backward-looking center*, $Cb(U)$, and a set of *forward-looking centers*, $Cf(U)$. $Cb(U)$ serves to link U to the preceding discourse, while $Cf(U)$ provides a set of entities to which the succeeding discourse may be linked. The *preferred* entity in the forward-looking set is $Cp(U)$.

In a discourse, we have an ordered list of utterances $U_1, U_2 ... U_m$. In the original Centering theory, there are defined three transition states: Continuing, Retaining and Shifting. Continuing is the transition state where backward-looking center is the same in two consecutive utterances ($Cb(U_n) = Cb(U_{n-1})$) and it also agrees with the preferred forward-looking center ($Cb(U_n) = Cp(U_n)$). Retain is the state where backward-looking centers agree ($Cb(U_n) = Cb(U_{n-1})$) but it seems that the center is going to change because the preferred forward-looking center is different ($Cb(U_n) \neq Cp(U_n)$). Finally, Shifting is the transition state where backward-looking centers are different ($Cb(U_n) \neq Cb(U_{n-1})$). Figure 5 list constraints and rules for each utterance U_n. An example of Centering interpretation of a discourse is shown in Figure 6.

Focusing and Centering are theories based on the same linguistic and cognitive phenomena. Actually, Focusing (Sidner, 1979) is the theory of discourse

Constraints	Rules
1. There is only one Cb. 2. Every element of $Cf(U_n)$ must be realized in U_n 3. $Cb(U_n)$ is the highest-ranked element of $Cf(U_{n-1})$ that is realized in U_n	1. If some element of $Cf(U_{n-1})$ is realized as a pronoun in U_n, then so is $Cb(U_n)$ 2. Continuing is preferred over Retaining which is preferred over Shifting.

Figure 5: Constraints and rules of Centering (interpretation of Brennan et al. (1987))

Utterance		Centering	Resolution
U_n	Bob is planning to go out today	Cf = {Bob}	
State = Continuing			

U_{n+1}	He called Charlie to go to the beach	Cb = Bob
		Cf = {Bob, Charlie, the beach}
			He = Bob

State = Retaining			

U_{n+2}	However, Charlie didn't answer his call	Cb = Bob
		Cf = {Charlie, Bob, Bob's call}
			his = Bob

State = Shifting			

U_{n+3}	He was already at the beach	Cb = Charlie
		Cf = {Charlie, the beach}
			He = Charlie

Figure 6: Example of centering.

structure which provided the basis to develop Centering. There are two levels of focusing in discourse: global and local (immediate). Entities relevant during whole discourse are considered in *global focus* while entities in focus across subsequent utterances are in *local focus*. In order to resolve anaphoras, which is the goal of works described in this section, only local focusing is taken into account. To avoid confusion, in the rest of the document "Focusing" will be used to refer only to local focusing.

The main differences between Centering and Focusing arise within their representation models. Sidner's *discourse focus* corresponds roughly to Grosz et al. *backward-looking center* of a utterance U ($Cb(U)$), while *potential foci* in Focusing correspond approximately to *forward-looking centers* ($Cf(U)$) in Centering. However, Focusing also introduces other concepts like, for instance, *actor focus* to handle multiple pronouns in a single utterance. In addition, their constraints to resolve anaphoras are quite different and consequently, further implementations are different depending on the theory they follow. The specific details of each theory are out of the scope of this document. We refer the reader to follow the cites if she/he wants to go deeper in them (Sidner, 1979; Grosz et al., 1983).

Originally, the use of centering and focusing theories for anaphora resolution only provides a search scope limited to entities in the immediately preceding utterance. Both theories bother about focus/center changes between utterances but ignore some intrasentential affairs. Also antecedents further than immediately previous utterances are ignored. Consequently, alternative models for Centering (Hahn and Strube, 1997; Walker et al., 1998; Strube, 1998) and for Focusing (Carter, 1986; Azzam, 1996) extend that search space to handle intrasentential anaphora and distant antecedents.

Carter (1986) argued that intrasentential candidates in Focusing should be preferred over candidates from previous sentences only in the cases where no discourse focus has been established or where discourse focus is rejected for syntactic or selectional reasons. That new rule was applied also for Centering (Walker, 1989).

The BFP (initials of the authors' names) algorithm for pronoun resolution (Brennan et al., 1987) is a practical implementation of Centering theory. It also contributes with some extensions to the original theory. It splits the Shifting transition state into Smooth-shift and Rough-shift. This change helps to define new rules which better find the center and its transitions.

BFP algorithm has some weaknesses mainly inherited from original Centering theory. First, it has difficulties solving global pronouns because it is only scoped in a local search. Second, it does not have any incremental method to process pronouns inside the same sentence. Third, intrasentential pronouns are not always solved because BFP is focused in discourse and it only search antecedents in previous utterances. Finally, candidate ranking is not completely specified and may be cases where several candidates have the same ranking and it is not possible to choose one of them.

Many works deal with these problems and propose some solutions. Most of them combine the main ideas of Centering with Hobbs' algorithm because, in both ways, strengths of one fit with many weaknesses of the other.

- Walker (1989) compares BFP with Hobbs' algorithm using the same test documents and concludes that BFP achieves better performance resolving intersentential pronominal anaphora while Hobbs' algorithm performs better finding intrasentential ones. Consequently, a potential modification is proposed for BFP based on Carter's extension for Focusing (Carter, 1986). The addition of Carter's rule to BFP improves its performance.

- Tetreault (1999) proposes an alternative to BFP called Left-Right Centering (LRC) which adheres to the constraints and rules of Centering theory and also incorporates a search process similar to Hobbs' Algorithm. It can be viewed as an extension of Hobbs' algorithm but including discourse information. Therefore, LRC resolves pronouns incrementally.

- Kameyama (1997) extends Centering in order to consider sentence structure with a hierarchy of clauses as centering-units instead of normal use of utterances as centering-units. It breaks down utterances in clauses which helps to search intrasentential pronouns using the same Centering principles.

- Strube's S-list algorithm (Strube, 1998) is based on the same idea of centering but makes it simpler. In spite of having a set of conditions to determine the current entity in focus, it has an ordered list of entities that grows as new entities are found through the discourse. The list (S-list) is reordered when new entities are added. When searching for antecedents it follows the S-list order. The first one that satisfies the agreement constraints with the pronoun is selected.

In the case of Focusing, Azzam (1996) discovers and solves two of its weaknesses. First, the original algorithm only deals with simple sentences with subject, verb and object optionally followed by prepositional phrases or adverbial adjuncts. However, real sentences might be more complex. Second, entities proposed by the algorithm for anaphora or coreference resolution are the entities in focus and do not include entities of the sentence under consideration. Both problems are solved breaking down complex sentences in small and simpler units named embedded sentences. However, Azzam et al. (1998) studied the incorporation of focusing for coreference resolution in a working coreference system without finding an improvement of the performance.

3.2.2 Discourse Representation Theory

Discourse Representation Theory (DRT) (Kamp and Reyle, 1993) is another discourse theory successfully applied to anaphora resolution. Each sentence is represented in a Discourse Representation Structure (DRS) which is a diagram with discourse referents at the top and conditions at the bottom. A DRS is a semantic representation of the sentence obtained using the sentence syntactic information and can be easily translated to a first-order logic formula. DRSs represent the meaning of the discourse and also impose constraints for pronoun resolution. Table 1 shows an example of how two sentences are represented in DRSs and pronouns are resolved.

Bob wants a new bicycle.	He doesn't have enough money.
$\boxed{\begin{array}{l} x, y \\ \hline x = BOB \\ NEW(y) \\ BICYCLE(y) \\ WANT(x, y) \end{array}}$	$\boxed{\begin{array}{l} z \\ \hline z = x \\ \neg \boxed{\begin{array}{l} u \\ \hline ENOUGH(u) \\ MONEY(u) \\ HAVE(z, u) \end{array}} \end{array}}$

Table 1: Example of two DRSs.

DRT has been adopted by many researchers and some works combine DRT with Focusing to take advantage of both techniques (Cormack, 1993; Abraços and Lopes, 1994).

3.3 Based on Constraints Satisfaction

Constraint Satisfaction approaches, also known as Factor-based approaches, combine information from different sources to resolve anaphoras or coreferences. Some authors split these factors into constraints and preferences. On one hand, constraints must be satisfied in order to accept a candidate as a possible antecedent. On the other hand, preferences are used to score and rank possible antecedents and choose the best one. Anyway, one can consider both as weighted constraints and model it as a constraints satisfaction problem. The antecedent which better satisfies the constraints is the one selected at the end.

This approach does not directly rely on any cognitive or linguistic theory about discourse understanding or representation, but can combine any possible one in order to filter or score candidates. As it is discussed later, works based on constraints satisfaction have been following the way from knowledge-rich solutions, trying to combine pragmatic and semantical information with full parse trees, to knowledge-poor solutions where only PoS tagging and chunking is required (Carter, 1986; Rich and LuperFoy, 1988; Carbonell and Brown, 1988; Lappin and Leass, 1994; Kennedy and Boguraev, 1996; Baldwin, 1997; Mitkov, 1998).

Carter (1986) developed a shallow processing approach implemented in a program called SPAR (Shallow Processing Anaphor Resolver) to resolve nominal anaphoras. SPAR combines different knowledge sources and strategies such as Focusing, Hobbs' algorithm, semantic rules and heuristics. After a syntactic analysis of sentences is done, SPAR constrains the search of antecedents providing a measure of *semantic density.* Antecedents that semantically agree with anaphor are proposed. Next, a set of focus-based rules are applied to anaphors in order to find one or more antecedent candidates. Both, Focusing theory and some rules inspired in Hobbs' algorithm are used adding intrasentential candidates. After that, domain constraints are required to discard inconsistent antecedents. Next, a common-sense inference is done if some anaphors remain unresolved. And finally, if still can not determine antecedents, a set of weak heuristics are activated.

Rich and LuperFoy (1988) proposed Lucy, a distributed architecture for anaphoric pronoun resolution. A set of modules (or factors) score the possible antecedents according to its criterion. A weighted average of all scores is calculated for antecedent ranking, and best ranked is selected as antecedent. Each factor evaluates independently a different aspect such as gender agreement, animacy, semantic consistency and so on (Figure 7). Therefore, Lucy exploits several different linguistic areas and theories.

The process first syntactically and semantically analyzes the input discourse. Next, each factor proposes candidates for pronouns and scores each proposed pair antecedent-anaphor following its own criterion. Each factor also scores pairs proposed by the other factors. Then, a module named Handler averages the scores and ranks them in order to choose the one with highest average. Scores can be negative (which means a negative recommendation) and have an associated confidence value which is used in the averaging formula.

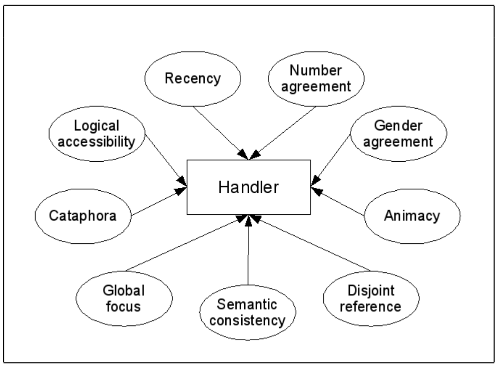

Figure 7: Lucy. Rich and LuperFoy's distributed architecture.

A similar idea to that of Lucy is proposed by Carbonell and Brown (1988) as a general framework for intersentential anaphora resolution. It is also based on a combination of multiple knowledge sources: syntax, semantics, dialogue structure and world-knowledge. Each kind of knowledge is implemented as a set of constraints. On one hand, some constraints act as a filter and directly eliminate candidates which violate them. On the other hand, the rest of constraints, called preferences here, are used as a voting scheme where the stronger preferences are assigned more votes. In the event of a tie, it is consider an ambiguity.

Some of the constraints proposed here are really elaborated but also require lots of knowledge and discourse understanding. For example, a set of constraints named precondition/postcondition constraints pretend to know real-world pragmatic knowledge about the actions expressed in a sentence in order to discard impossible antecedents. For instance:

A) Bob lent Charlie some money. He spent it in a new bicycle.
B) Bob borrowed from Charlie some money. He spent it in a new bicycle.

In example A, precondition/postconditions constraints would discard "Bob" as referent since he does not have the money anymore. He lent it to "Charlie" so, "Charlie" is the only possible antecedent for "He". In the contrary, in the example B, "Charlie" would be discarded as antecedent for the same reason. In this case, "He" is referring to "Bob". In order to implement that, a big amount of data specifying each verb preconditions/postconditions must be generated.

Many works use similar schema of constraints and preferences (or filtering and ranking) to develop multi-strategy solutions. For instance, Lappin and Leass (1994) propose an approach similar to Carbonell and Brown (1988). The main difference between them is that Lappin and Leass only use constraints at morphological and syntactic level avoiding the hard task of semantic and pragmatic constraint implementation, obtaining a reasonable accuracy. Lappin and Leass also designed a filter to find pleonastic pronouns and discard them. Until then, most of works expected a previous manual filtering of non-anaphoric pronouns. However, it relies on perfect morphological and syntactic analyzers. Therefore, Kennedy and Boguraev (1996) proposed a parser-free pronoun resolution system that uses a set of heuristics to estimate the full parser perfect information expected by Lappin and Leass. Also Baldwin (1997) presented a pronoun resolution system only using morphological information and shallow parsing. Baldwin's system is called CogNIAC and achieves high precision on its resolution because it only takes a decision when very high confidence constraints have been satisfied.

Following this knowledge-poor trend, Mitkov (1998) published a robust pronoun resolution system that only relies on part of speech tagging and shallow syntactic information (chunking). It has a set of elaborated constraints that obtain clues (antecedent indicators) to resolve pronouns. The system works like previous ones, a set of constraints filter possible candidates and another set of soft constraints scores and sort them in order to choose the one with highest score.

LaSIE (I and II) is the Large Scale Information Extraction system developed in the University of Sheffield to participate in MUC-6 and MUC-7 (Humphreys et al., 1998). The system is a pipeline of modules each of which processes the entire text before next is invoked. The modules are: Tokenizer, Gazetteer Lookup, Sentence Splitter, Brill Tagger, Tagged Morph, Buchart Parser, Name Matcher, Discourse Interpreter and Template Writer. LaSIE-II not only resolves coreferences but also Named Entities and templates of Information Extraction (template elements, template relations and scenario template). Coreference resolution is done in the Discourse Interpreter module. There, a Domain Model (knowledge of the domain) is represented in a semantic net whose nodes represent concepts, with an associated attribute-value structures recording properties and relations of the concept, and whose arcs model a concept hierarchy and support property inheritance. The parser gives a semantic representation of the input text to the Discourse Interpreter which adds it to the Domain Model to finally become a Discourse Model. Coreference is done when adding new instances to the Discourse Model. If new instance can be paired with an existent one, they are coreferential. To determine when can be paired a similarity score is calculated based on a distance in the concept hierarchy and the number of shared properties. The highest scoring pair is merged and all their properties combined forming an entity. The system participates in MUC-6 (LaSIE-I) and MUC-7 (LaSIE-II) and achieves better performances than most of the other participants.

4 Corpus-based Approaches

This section includes approaches that use corpora for training or to obtain statistical data. These works mostly appeared after MUC-6 (Grishman and Sundheim, 1996) and MUC-7 (MUC, 1998), where annotated coreference documents were published. Therefore, most of the works since middle 90s to present time are of the kind of corpus-based approaches for coreference resolution.

4.1 Statistical and Manual Approaches

Ge et al. (1998) presented a statistical approach to anaphora resolution. They study the probability of a pronoun to be anaphoric with a mention, given a set of attributes. Attributes are distance, number of times the possible referent is mentioned, type of antecedent (NP for example) and so on. One interesting attribute is the one called "Hobbs distance", which is the position that an antecedent has in the list of proposed antecedents when executing a slightly modification of Hobbs' algorithm to resolve a pronoun. The algorithm collects statistics on a annotated training corpus (Penn Tree Bank) and uses these probabilities to resolve pronouns.

Cardie and Wagstaff (1999) proposed a clustering approach for coreference resolution. The order of resolution follows the NP found order in the document and decisions are taken greedily. When a new mention is found in the document, it is classified in one of the existent clusters or a new one is created for it. The degree of relatedness of an NP with existent clusters is represented in a distance metric. That distance is defined as the weighted summatory of a set of features. If the distance is lower than a cut-threshold the NP is assigned to that cluster. Feature function weights are chosen by hand and cut-threshold is chosen to maximize F_1 on the development set.

Harabagiu et al. (2001) studied annotated corpora for coreference in order to acquire constraints for a resolution system. In spite of defining constraints using expert linguistic knowledge, they evaluate pairs of coreferential (or not) mentions in the annotated texts. Positive training instances are extracted from coreference chains pairing any NP inside the chain with any other, without considering the order inside the chain. Thanks to this methodology, one can obtain much more positive samples than only pairing anaphors and antecedents. Negative training pairs are easily acquired pairing NPs from different chains. Studying these samples they develop a set of rules, some of them semantic, that utilize WordNet and some heuristics for disambiguation. The set of coreference rules is then transformed into a corresponding set of soft constraints by estimating the accuracy of each rule on the training data. The resolution is done with a local-search algorithm that, starting with a random solution, changes partitions optimizing by pairwise coreference probabilities given by the soft constraints.

4.2 Supervised Learning

This section summarizes the state of the art of supervised machine learning coreference resolution systems. It is divided in five subsections, four of them corresponding to the models most followed by researchers, namely: Pairwise

Classifiers, Coreference Chains, Graph Partitioning and Conditional models. Last subsection describes some works dedicated to the addition of semantic features.

4.2.1 Pairwise Classifiers

First machine learning systems developed for coreference resolution were based on pairwise classifiers using decision trees (DT) (normally C4.5 or C5: (Quinlan, 1993)). Each pair NP-NP found in the document (following some arbitrary order) is considered as possible coreferential pair. A set of feature functions evaluates pair compatibility, each one according to its own criterion. Then, taking account of feature functions returned values, the DT classifies each pair as coreferential or not. Once all pairs are classified, implicitly, a single-link clustering is done producing final coreferential chains.

McCarthy and Lehnert (1995) developed RESOLVE, a domain specific machine learning system for coreference resolution. RESOLVE learns a DT in order to classify pairs of mentions as coreferential or not. It consists of 8 features, 3 of them domain-specific. The others are lexical, semantic and positional. No syntactic features are used.

Later, Soon et al. (2001) proposed a general corefence resolution system also based on a DT classifier using 12 features. Their features are lexical, syntactic, semantic and positional. However, a study of features contribution reveals that only 3 features are highly informative to such a degree that the other 9 would be the firsts ones to be considered for pruning away by the DT algorithm. The 3 informative features are STRING_MATCH (the two strings are the same), ALIAS (one mention is an alias of the other) and APPOSITIVE (both NPs are in appositive position in the document). The DT learned with these 3 features is only about 2% worst than the DT learned using the 12 features. There is an example of a DT learned in Figure 8. Each feature evaluates a pair of mentions and returns a value *true* (t) or *false* (f) and in some cases there is also an *unknown* (u) value. The feature DISTANCE returns a numeric value indicating sentence distance between both mentions.

Training set creation is done as follows. Each pair of mentions annotated as coreferential in training corpus generates several training instances. The number depends on the number of candidate mentions obtained by the preprocess. Concretely, if $mention_a$ and $mention_b$ are annotated as coreferential, the pair $mention_a - mention_b$ is a positive example and each $mention_i$ between $mention_a$ and $mention_b$ in the document generates a pair $mention_i - mention_b$ which is a negative example for training.

The approach of Soon et al. (2001) achieved reasonably results in common datasets MUC-6 and MUC-7 (62.6% and 60.4% respectively) comparable to that of state-of-the-art nonlearning systems on the same datasets[2]. Consequently,

[2] http://www-nlpir.nist.gov/related_projects/muc/proceedings/co_score_report.html

many further works use Soon's system as a baseline.

```
STR_MATCH = t: +
STR_MATCH = f:
|   J_PRONOUN = f:
|   |   APPOSITIVE = t: +
|   |   APPOSITIVE = f:
|   |   |   ALIAS = t: +
|   |   |   ALIAS = f: -
|   J_PRONOUN = t:
|   |   GENDER = f: -
|   |   GENDER = u: -
|   |   GENDER = t:
|   |   |   I_PRONOUN = t: +
|   |   |   I_PRONOUN = f:
|   |   |   |   DIST > 0: -
|   |   |   |   DIST <= 0:
|   |   |   |   |   NUMBER = t: +
|   |   |   |   |   NUMBER = f: -
```

Figure 8: Example of a decision tree classifier (Soon et al. 2001)

In order to study feature contributions, training instance selection and, in general, propose improvements to Soon's system, Ng and Cardie (2002b) publish a system for coreference with several new features and modifications to the machine learning framework. This system outperforms Soon's due to some changes. First, lots of new features are added and old ones suffer some modifications in order to improve them. Second, pairs are classified with a confidence value and not with straightforward binary decisions. It means that even when a positive match is found the system keep seeking for another one with higher confidence value. Third, instance selection for training is changed in order to avoid pairs pronoun-NP but allow NP-NP and NP-pronoun, assigning lower confidence values to the first ones.

Not only decision trees have been developed in order to learn pairwise classifiers. For instance, Denis and Baldridge (2007) learn two binary classifiers using maximum entropy models. One classifier determines if a pair of mentions are coreferential or not, while the other classify single mentions as anaphoric or not. The second is used as a filter before coreference classifier is applied. This kind of anaphoric filter has been also tried in other works (Ng and Cardie, 2002a) and its goal is to improve system precision without loss of recall. However, normally it does not work as is expected. Note that neither coreferences are always anaphoric, nor anaphoric mentions are always coreferential. Denis and Baldridge (2007) tested how a cascade configuration of both filters causes a loss of recall. However, they implemented a system to combine these informations using Integer Linear Programming (ILP) which improves final results due to a better information combination.

Many other works have trained a pairwise classifier different of decision trees such as RIPPER (Ng and Cardie, 2002b), maximum entropy (Denis and Baldridge, 2007; Ji et al., 2005) or Support Vector Machines (Yang et al., 2006). The later, proposes a new kernel that interprets syntactic parsed trees as features, avoiding the efforts of decoding them into a set of flat syntactic features.

4.2.2 Coreference Chains

A different approach to pairwise classification is proposed by Luo et al. (2004). In this work, a model based on a Bell Tree is proposed. Traversing mentions in a document from beginning to end, a tree is formed generating all different possible combinations of coreferential chains. Each possible combination is a node in the tree and when a new mention is incorporated a new level of nodes representing all possibilities is created. Each edge has a confidence value obtained with a set of weighted feature functions evaluating the compatibility of the new mention incorporated to each formed chain. Every step also performs some pruning processes in order to keep an abordable tree size discarding lowest confidential coreference chain combinations.

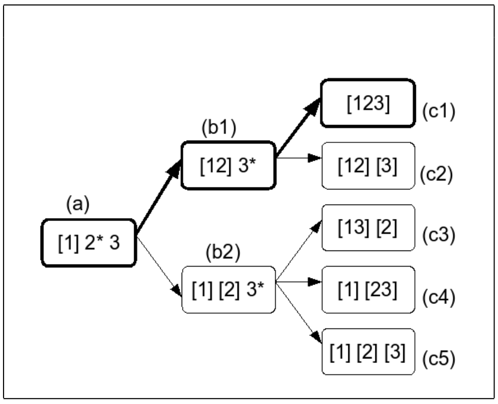

Figure 9: Example of a Bell Tree

It is important to note that feature functions evaluate possible coreference of a new mention with the chain already formed and not only by pairs. In this way, some information about an entity that may be separated in different mentions can be used to evaluate compatibility more accurately. For example, in a document one may find these three mentions in this order: 1) “Alice Smith” 2) “A. Smith” and 3) “She”. A classifier by pairs may fail evaluating gender agreement when classifying the pair “A. Smith” and “She”. However, using the already formed chain “Alice Smith”-“A. Smith”, it is easier to know that “She” is also a good candidate to include in that chain. Figure 9 shows the tree generated for this example. In the first step (a), a new entity is started with mention 1 (Alice Smith) and active mention is 2 (A. Smith). Then, mention 2 can be added to the existent entity o can start a new one. Two edges and two new tree nodes are generated: (b1) and (b2). In our case, the most probable node is (b1) because its edge has higher confidence value. Next, once mentions 1 and 2 are the same entity (b1) the system has to decide what to do with active mention 3. Again two options are possible, put all three mentions in the same entity (c1) or create a new one for mention 3 (c2). Finally (c1) is chosen because its edge has higher confidence value.

4.2.3 Graph Partitioning

Graph partitioning approaches are a natural evolution of pairwise classifiers in order to resolve corefences. Like in coreference chain trees (Section 4.2.2), a set of advantages are easily incorporated when resolving coreferences as groups. Indeed, after pair classification, implicitly a single-link clustering is done in order to finally decide coreference chains. Viewing it from groups point of view one can avoid contradictions in the results and lacks of information found when classifying by pairs.

Generally, graph partitioning is done over an undirected graph in which vertices are mentions and edges are weighted by the scores of feature functions between adjacent mentions. Normally, edge weights are viewed as distances and the algorithm cuts edges further than a threshold r in order to isolate the groups representing independent coreference chains. Both feature function weights and cut-threshold r are learned with training data.

Finley and Joachims (2005) developed a Support Vector Machines (SVM) classifier in order to learn the similarity measure used to join elements of the same coreference chain. This similarity measure is used by *correlation clustering* algorithm. The novelty of the system is that the measure is not learned classifying pairs of elements (as COREF or NO-COREF). The SVM learning algorithm is modified to learn a similarity measure that classify sets of elements in a set of partitions. The *loss function* used for learning is the same function used in the MUC scorer which it directly associates the learning process with the final task. One of the problems found in this method is the impossibility of train all the possible incorrect partitions. To solve it, two approaches are proposed in order to iteratively determine the most relevant partition samples for training. Their results confirm that a groupwise classifier performs better than pairwise, but not comparable results (MUC test using MUC scorer, for example) are published.

Nicolae and Nicolae (2006) presented an algorithm called BestCut to resolve coreferences cutting edges in a graph. First, entities are separated by their type (Person, Organization, Location, Facility and GPE). Pronouns are not included in the process until the end. Second, a classifier decides if each pair of mentions corefer or not with an associated confidence value. Then, all coreferential pairs are linked in a graph where vertices represent the mentions and edges represent that they are coreferential. Edge weights are the confidence values returned by the classifier. BestCut cuts the edges with minimum cut weight. That is, the weight of the cut of a graph into two subgraphs is the sum of the weights of the edges crossing the cut. This process is repeated until a stop condition is

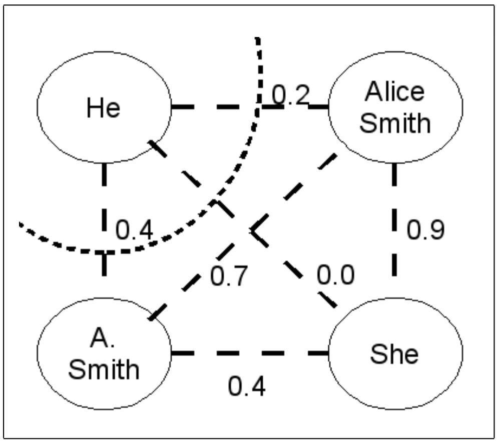

Figure 10: Example of a Graph Partitioning

satisfied. A maximum entropy model is used for training the classifier and its confidence values. The stop condition is also a trained classifier with a set of features that optimizes ECM-F measure over the graph partitions. At the end of the process, pronouns are assigned using the same pairwise coreference classifier. The process presented here is promising as is shown in the results using true mentions. However, it heavily relies in a previous sense disambiguation step for NPs when automatic preprocess is done.

Figure 10 is an example of how the mentions "he", "A. Smith", "Alice Smith" and "She" would be represented as a graph following Nicolaes representation but including pronouns.

A similar approach is proposed in Klenner and Ailloud (2008) called *coreference clustering*. All positively (COREFERENTIAL) classified pairs from a pairwise classifier are the input of a clustering algorithm. The classification cost depends on the number of similar positive and negative instances found by the pairwise classifier in the dataset. The clustering algorithm uses the costs of each pair to finally decide the coreference chains with a minimum cost. Doing clustering after classification of pairs improves final performances because it implies coherence with all the mentions in the final chains.

4.2.4 Conditional Models

Conditional models of identity uncertainty are proposed (McCallum and Wellner, 2003; Singla and Domingos, 2005; Richardson and Domingos, 2006) and applied to coreference resolution (McCallum and Wellner, 2005; Culotta et al., 2007). These models are general enough to include most of machine learning implementations based on feature functions. Therefore, pairwise classifiers, coreference chain trees and also graph partitioning approaches can be considered as particular cases/implementations of these more general models.

First, the pairwise model is defined as following. Given a pair of mentions x_i and x_j, a binary random variable y_{ij} is 1 if both mentions are coreferential. A set of feature functions evaluate the compatibility of that pair. For example, f_k returns 1 if x_i and x_j agree in number. Each feature f_k has an associated real-valued parameter λ_k.

$$p(y_{ij}|x_i, x_j) = \frac{1}{Z_{x_i,x_j}} exp \sum_k \lambda_k f_k(x_i, x_j, y_{ij}) \tag{1}$$

where Z is a normalization parameter. Normally, in a pairwise classifier, if the probability of x_i and x_j to be coreferential is higher than a 0.5 the classifier assigns them as coreferential. A learning process determine best possible values of parameters λ.

Using the same notation, a more general model, First-Order Logic Model (or Groupwise Model), is obtained defining a vector of mentions $\mathbf{x}^j = \{... x_i ...\}$ which includes a group of mentions candidates to form a coreference chain. This time, feature functions evaluate the agreement of all the mentions in that possible group.

$$p(y_j|\mathbf{x}^j) = \frac{1}{Z_{\mathbf{x}^j}} exp \sum_k \lambda_k f_k(\mathbf{x}^j, y_j) \tag{2}$$

where Z is a normalization parameter. This model includes relational information between elements to resolve. Feature functions are not limited to compare some characteristic of two elements or to evaluate a particular characteristic of an element, but also adds the possibility of functions evaluating the compatibility of a feature in a group of elements (Culotta et al., 2007).

The models define how the most probable partitioning can be found depending on the feature functions, even pairwise or groupwise. Decisions can not be taken independently in a pair or group of elements, but they depend on the configuration of the others. However, in practice, enumerating all possible configurations in order to find the most probable can result in intractable combinatorial growth (de Salvo Braz et al., 2005). Consequently, a set of reductions and practical implementations are proposed and tested with promising performances (McCallum and Wellner, 2005; Culotta et al., 2007).

4.2.5 Adding Semantic Features

Several machine learning systems incorporate semantic features like WordNet similarities/distances or aliases. There are many WordNet similarities using different relations (IS-A, synonymy, homonymy...) in order to evaluate the similarity between two word senses. For example, a possible distance between two word senses is the number of word senses in the shortest path following only hyperonymy relations. However, due to word sense disambiguation errors, WordNet similarities may result useless noisy features. Some works studied how to incorporate additional usefull semantic information to ML systems for coreference resolution.

Ji et al. (2005) added semantic relations to refine decisions taken by a pair classifier. The classifier first determines coreferential pairs using first a set of hand-written rules and then maximum entropy models. Once classification is done with a associated confidence value, a search of semantic relations between mentions that seem non-coreferential is performed. After that, a set of rules are applied to mentions with relations. This step helps to improve precision by pruning incorrect coreference links between mentions and also improve recall by recovering missed links. Finally, a pairwise classification is done again using confidence values of the first classification as another feature and using also the recent incorporated semantic information.

Other works studied incorporation of semantic features into pairwise classifiers (Ponzetto and Strube, 2006; Ng, 2007). Ponzetto and Strube (2006) developed a set of features to add to their maximum-entropy-trained pair classifier. Features from three different knowledge sources are proposed: Semantic Role Labeling, WordNet and Wikipedia.

- Semantic Role Labeling. The semantic role of each mention is automatically annotated with a parser. Then, two features are added for each pair of mentions I_SEMROLE and J_SEMROLE indicating the semantic role of each mention i and j.
- Wikipedia. Wikipedia is a multilingual Web-based free-content encyclopedia. Each non-pronominal mention ($mention_i$) is searched in Wikipedia (querying the head lemma or the Named Entity) and the response's article ($article_i$) is assigned to it. Sometimes a disambiguation page is found instead of a direct article. In this cases, the article finally assigned depends on the other mention of the pair mention-mention in classification process. There are also other cases where no article can be assigned to a mention. Six kinds of features are added to each pair of mentions:
    - I/J_GLOSS_CONTAINS: True when the first paragraph of $article_{i/j}$ contains $mention_{j/i}$.
    - I/J_RELATED_CONTAINS: True when $article_{i/j}$ links to $article_{j/i}$.
    - I/J_CATEGORIES_CONTAINS: True when categories of $article_{i/j}$ contain $mention_{j/i}$.
    - GLOSS_OVERLAP: An overlap score between first paragraphs of $article_i$ and $article_j$.
    - WIKI_RELATEDNESS_BEST: Given several relatedness scores (following Wikipedia categories of the articles in different ways) this feature chooses the highest one.
    - WIKI_RELATEDNESS_AVG: The average of all relatedness scores.

- WordNet. There are several measures of distance/similarity using synsets of WordNet. Ponzetto and Strube (2006) developed two features WN_SIMILARITY_BEST and WN_SIMILARITY_AVG which respectively return the best score of all available WordNet similarities and their average. In addition, they avoid disambiguation and all possible synsets of $mention_i$ are scored versus all possible synsets of $mention_j$.

Ng (2007) studies incorporation of shallow semantic features. Based on a decision tree pair classifier, a set of features such as "semantic agreement", "semantic ACE class" and "semantic similarity" are proposed. Also other features are incorporated and tested like patterns, anaphoricity and coreferentiality. Semantic Agreement feature is similar to the ones based on WordNet but tries to avoid the common disambiguation errors when assigning senses to nouns or NP. In Ng's work, he looks for nouns in apposition with Named Entities which already have been assigned a semantic class. In these cases, the sense of the noun is determined by the semantic class of the appositive NE. Second feature "semantic ACE class" takes as main classes the ones used in ACE. The feature considers two mentions to be semantically compatible if and only if both mentions have a common ACE semantic class. Last feature is similar to the most used WordNet distance but here it incorporates previous word sense disambiguation based on nouns found around the repetitions of the noun to disambiguate in the document.

4.3 Weakly Supervised and Unsupervised Learning

Due to lack of big amounts of annotated data for training coreference resolution systems, some researchers have explored weakly supervised and unsupervised approaches. Co-Training and bootstrapping are two resources widely used in this kind of situations. Also, unsupervised clustering and generative models are taken into account.

Müller et al. (2002) applied co-training for coreference resolution in German texts. Starting with few manually annotated documents, the system learns a set of features and gradually annotates more texts. The performance of the experiment is not significantly better than systems trained with manual annotated data. However, the use of Co-Training seems to be able to save manual annotation work.

Bean et al. (2004) proposed a system with contextual role knowledge. First, few "easy" coreferences are annotated as a seed for subsequent bootstrapping. Using a NERC, Named Entities of the same class which are equal or almost equal, using some heuristics, are annotated as coreferential. Also antecedents for reflexive pronouns are searched and automatically annotated. These two kinds of annotation have high precision but low recall and are the initial seeds. Once initial annotation is done, a set of caseframes are extracted from pairs anaphor/antecedent. There are three kinds of caseframes: network, lexical and semantic. For example, a network pattern is *murder of <NP>* and *killed*

<*patient*>. Using these caseframes, some filters and a set of “knowledge sources” (similar to feature functions of other works), a Dempster-Shafer decision is done in order to finally obtain coreference chains.

Haghighi and Klein (2007) developed a nonparametric Bayesian approach for coreference resolution and also cross-document coreference. The model is fully generative and produces each mention from a combination of global entity properties and local attentional state. It uses some information from annotated corpora and true mentions for training and test but the approach is unsupervised. It is a novelty approach never used before for coreference resolution and might be showing a new research line to follow. Actually, Ng (2008) contributes with three modifications to that model in order to solve its potential weaknesses and improve the results. Ng (2008) also proposes an Expectation Maximization (EM) clustering model which is unsupervised, although it is used in a weakly supervised manner in the experiments using an only labeled document.

5 Corpora and Evaluation

This section reviews the most utilized annotated corpora and metrics used in the state of the art to compare the performances of different systems. First, two corpora are introduced: MUC and ACE. Then, there is a brief description of the most popular metrics: MUC-scorer, ACE-value, B-CUBED and CEAF.

5.1 MUC

The Message Understanding Conferences (MUC) were competitions in Information Extraction initiated in 1987 and founded by DARPA (Grishman and Sundheim, 1996; MUC, 1998). The goal was to encourage the development of new and better methods for many tasks related to Information Extraction. Many research teams competed against one another. Coreference resolution was included in the competition in MUC-6 (1995) and MUC-7 (1997). Annotated corpora in English for coreference is copyrighted by the Linguistic Data Consortium[3].

5.2 ACE

Automatic Content Extraction (ACE)[4] is a program to support automatic processing of human language in text form (NIST, 2003). Promoted by National Institute of Standards and Technology (NIST), the program is devoted to three source types. These are, namely, newswire, broadcast news (with text derived

[3] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T02

[4] http://www.nist.gov/speech/tests/ace/

from ASR), and newspaper (with text derived from OCR). Also texts in languages different than English are available.

5.3 Metrics

Evaluation of coreference systems performance can be done using different metrics. The score for a particular run of a system is a measure of how well it is performing the task. Normally, a metric compares the expected results (*key* from here on) with system output ones (*response* from here on). Therefore, a metric should indicate the way to follow to improve a system while, at the same time, should be useful to compare different systems. However, coreference tasks have been scored using several different metrics and, as discussed in section 5.4, performance comparatives must be done carefully.

In this subsection the most widely used metrics are explained. First, there are two metrics associated with world-wide coreference resolution contests: MUC-score and ACE-value. Second, two metrics, B-CUBED and CEAF, are presented. Those were developed in order to avoid some weaknesses of previous ones.

5.3.1 MUC-scorer

MUC-scorer (Vilain et al., 1995) was used in the MUC task. It is a link-based metric which evaluates precision, recall and their harmonic F-measure. First, a count of common links between *key* links and *response* ones is performed. The link precision is the number of common links divided by the number of response links, while recall is the number of common links divided by the number of key links. First harmonic (F_1) is calculated as any other precision and recall harmonic measure:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

There are two important weaknesses in this metric:

- MUC scoring algorithm does not give any credit to coreferential mentions detected but not included in any coreference chain in the response. This is because only links are considered for evaluation. It is known that a single mention can not constitute a coreference chain by itself. Therefore, no single mentions are found in key documents. However, for a coreference system, it is important to distinguish between a non coreferential mention and a mention that is coreferential but the other mentions of its chain may not be detected.

- The algorithm penalizes links independently of their type or their coreference chain and does not distinguish which ones are important errors. This weakness causes at least two undesirable consequences. First, the metric intrinsically favors systems producing fewer coreference chains and may result in higher F-measures for worse systems. A response with few but correct links may obtain really high precision score. Second, a system that joins, for example, two big coreference chains in one, is penalized only once, while a system that distinguish these two big chains but introduces an incorrect link with some other mention has the same penalization.

Consequently, a system that performs really better than other for a human understanding, might have the same score using MUC-scorer or even worse. In the following example it is clearly exposed.

Bob$_1$ is planning to go out today. **He**$_2$ called **Charlie**$_3$ to go to **the beach**$_4$. However, **Charlie**$_5$ didn't answer **his**$_6$ call because **he**$_7$ already was at **the beach**$_8$.
key chains: $\{1, 2, 6\}_{Bob}$, $\{3, 5, 7\}_{Charlie}$, $\{4, 8\}_{beach}$
System1 chains: $\{1, 2, 6, 7\}_{Bob}$, $\{3, 5\}_{Charlie}$, $\{4, 8\}_{beach}$
System2 chains: $\{1, 2, 3, 5, 6, 7\}_{Bob/Charlie}$, $\{4, 8\}_{beach}$

In a human point of view, *System1*, that detects the three coreference chains but includes mention 7 (he) in Bob's chain, would be considered quite good because only fails in one pronoun but seems that it is "understanding" that there are two people. However, *System2*, a system that joins Bob and Charlie in the same chain, would not be considered as good by a human.

Recall scores are 4/5 for *System1* and 5/5 for *System2*. And precision scores are 4/5 and 5/6 respectively. Consequently, MUC-scorer determines that the first system (good for a human) has a F-measure of 80.0% while the second one (bad for a human) obtains 90.9%.

Despite its unintuitive results in some cases, MUC scorer is the most widely used scoring algorithm in the state of the art of coreference resolution at least for two reasons. First, MUC corpora and MUC-scorer were the first available. And second, it is easy to understand and implement.

5.3.2 ACE-value

ACE-value (NIST, 2003) is the scoring algorithm used to evaluate the ACE task. Each error found in the response has an associated cost. An error can be a false-alarm (mention included in the response but not in the key), a miss (the opposite) or a missclassification of a coreference chain. The cost associated to each error depends on the type of the entity (e.g. PERSON, LOCATION, ORGANIZATION) and on the kind of mention (e.g. NAME, NOMINAL, PRONOUN). The final cost is the sum of the costs of all the errors made and is normalized versus the cost that would have a system with an unannotated output. Finally, the final score is the substraction of the normalized cost from 1.

A perfect response will obtain an score of 100% but note that an score of 0% is not the worst possible, is the one obtained by a system without any identified coreferential mention. The score of a system could be negative. The interpretation of the score is that a system with a better score than another has made less or less important errors than the other. However, it is important to emphasize that an ACE-value of, for example, 85% does not mean that the system performs correctly 85% of coreferences. It means that this system error cost is the 15% of the error cost of a system with an unannotated output.

ACE-value is used in several works in the state of the art. However, since the cost is entity-type and mention-type dependent, it needs an annotated corpus not only with coreference chains but also with entity types and mention types. Of course, ACE corpus has these annotations, but MUC hasn't.

5.3.3 B-CUBED

B-CUBED (Bagga and Baldwin, 1998) is a coreference scoring algorithm appeared to overcome the weaknesses of MUC-scorer. The main difference between these two algorithms is that MUC-scorer is a link-based measure while B-CUBED is mention-based. Concretely, B-CUBED precision and recall are calculated for each mention and then final precision and final recall are the total average. This average has the particularity that each mention can have a different weight. For each $mention_i$ we define:

- C_i as the number of correct response mentions in the coreference chain where $mention_i$ is included in the response.
- R_i as the total number of response mentions in the coreference chain where $mention_i$ is included in the response.
- K_i as the total number of key mentions in the coreference chain where $mention_i$ is included in the key.

$$Precision_i = \frac{C_i}{R_i} \; ; \; Recall_i = \frac{C_i}{K_i} \tag{4}$$

Although it is not clearly specified in the original document, we assume that Precision is evaluated for each mention included in the response and Recall for each mention included in the key. Total precision and recall are the weighted average of each mention precision and recall:

$$Total\,Precision = \sum_{i \in response} w_i * Precision_i \tag{5}$$

$$Total\,Recall = \sum_{i \in key} w_i * Recall_i \tag{6}$$

Where w_i is the weight associated to $mention_i$. These weights depends on the task and, when scoring coreference resolution they are normally fixed to the inverse of number of response mentions when calculating precision and to the inverse of number of key mentions when calculating recall. F_1 is calculated as in equation 5.3.1.

Some authors (Luo, 2005) criticize B-CUBED claiming that a response with all mentions in the same chain obtains 100% of recall and a response with mentions identified but without any link (i.e. each mention is a coreference chain itself) obtains precision of 100%. from our point of view, this is not incorrect when response includes all key mentions and only these ones. If response includes mentions not included in the key, precision decreases. On the contrary, mentions included in the key but not included in the response causes a decrease of recall. Therefore, this is useful when evaluating systems with automatic mention detection. However, if a system knows a priori the key mentions and classifies them in coreferences chains, this metric may be considered too generous.

The use of B-CUBED in the state of the art is not really extended. Although, it is easy to implement and some researches also evaluate their results with this metric for further comparisons.

5.3.4 CEAF

Luo (2005) proposed a Constrained Entity-Alignment F-Measure (CEAF) for evaluating coreference resolution. The algorithm is more complex but it avoids the problems of the previous ones. CEAF is computed based on the best one-to-one map between key coreference chains and response ones. This is one of the main differences with MUC-scorer and B-CUBED. These two algorithms allow a mention to be used multiple times when scoring. It causes them to be too generous with their scores and in some cases produces unintuitive results as in the example of Section 5.3.1. CEAF handles the evaluation as a one-to-one map, so a mention will never get double credit. Moreover, with CEAF one can interpret results intuitively. For example, when the score of a coreference system is 85%, one can say that this system performs correctly 85% of coreferences.

We refer the reader to the original paper (Luo, 2005) for specific details of the algorithm. Here we expose a simplification in order to understand the main ideas of the algorithm. First let G be all possible one-to-one maps between key coreference chains and response coreference chains. This means that each coreference chain in the response is uniquely associated with one in the key, for each possible map $g \in G$. Then, we define a function ϕ which evaluates the similarity of two coreference chains (for example, one of the key K_i and one of the response R_j):

$$\phi(K_i, R_j) = |K_i \cap R_j| \tag{7}$$

This function could be defined differently depending on the task one wants to evaluate. We show here a simple one which counts the common mentions in K_i and R_j. Next, we find the map which maximizes the sum of similarities, and we call it $g*$:

$$g* = argmax_{g \in G} \sum_{(i,j) \in g} \phi(K_i, R_j) \tag{8}$$

Finally, we can define precision, recall and F_1 as follows:

$$Precision = \frac{\sum_{(i,j) \in g*} \phi(K_i, R_j)}{\sum_i \phi(R_i, R_i)} \tag{9}$$

$$Recall = \frac{\sum_{(i,j) \in g*} \phi(K_i, R_j)}{\sum_i \phi(K_i, K_i)} \tag{10}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{11}$$

CEAF is a novel metric and it is not yet extended in the state of the art. It also does not help, the fact that it is not as easy to implement as the others. However, some works already publish their performances evaluating it with CEAF and it would be expected that progressively many researchers take into account this metric.

5.4 Comparing Results

Due to differences between metrics and preprocesses, coreference systems comparisons must be done carefully. Table 2 and Table 3 compares some of the state-of-the-art performances in MUC and ACE, respectively. Even if dataset, scorer and preprocess are shown in these tables, each system has its own nuances that might be not represented here. Only some representative systems have been included in these tables.

Both tables have a column called *Preprocess* which indicates how the mentions used in the coreference resolution system have been obtained. Mentions might be obtained using an automatic preprocess (auto) or from the manually annotated key (true mentions). In the first case, many mentions are non-coreferential and should be discarded by the system, which increases the difficulty, while in the second case only a chain assignation is needed. There is also a special case (auto+keys) where an automatic preprocess is done but true mentions not detected by the system are also included increasing recall.

Authors	Dataset	Scorer	Preprocess	Performance (F_1 %)
Humphreys et al. (1998)	MUC-6	MUC	auto	61.0
	MUC-7	MUC	auto	61.8
Cardie and Wagstaff (1999)	MUC-6	MUC	auto	54.0
Soon et al. (2001)	MUC-6	MUC	auto	62.6
	MUC-7	MUC	auto	60.4
Ng and Cardie (2002b)	MUC-6	MUC	auto	70.4
	MUC-7	MUC	auto	63.4
Harabagiu et al. (2001)	MUC-6	MUC	true mentions	81.9
Luo et al. (2004)	MUC-6	MUC	true mentions	85.7
McCallum and Wellner (2005)	MUC-6	MUC	true mentions	73.4
Haghighi and Klein (2007)	MUC-6	MUC	true mentions	70.3

Table 2: Results comparative for MUC-6 and MUC-7.

6 Conclusion

Coreference resolution research has been very active last decade since the appearing of annotated corpora and the first machine learning systems. Many advances have been possible mainly thanks to the existence of two evaluation frameworks: MUC and ACE. ACE dataset is newer and larger than MUC's and contains richer annotations. More recently, the ACE datasets are being increasingly used by researchers in the area.

Regarding the metrics, although the great majority of published papers have been using MUC-scorer to evaluate experiments, this metric has some weaknesses as is discussed in Section 5.3.1. Considering the other options for evaluation – ACE-value, B-CUBED and CEAF – the best alternative for MUC-scorer seems to be CEAF which measures more accurately the performance on the task of coreference resolution.

Taking into account the state of the art, the immediate future of the research in coreference resolution seems to be evolving mainly in three different ways. First, models for supervised learning might be improved with better training instance selection methods and new procedures for a better combination of the available information: the groupwise approaches that form coreference chains taking care of the whole group of mentions are more appropriate than pairwise ones. Second, the addition of semantic and pragmatic knowledge to the systems should improve final performances: it is well-known that some ambiguities at syntactic or lexical level need world knowledge or discourse comprehension to be solved. Thus, the use of ontologies and other resources for disambiguation

Authors	Dataset	Scorer	Preprocess	Performance (F_1 %)
Ng (2005)	ACE-bnews	MUC	auto	64.9
	ACE-bnews	B-CUBED	auto	65.6
	ACE-npaper	MUC	auto	69.3
	ACE-npaper	B-CUBED	auto	66.7
	ACE-nwire	MUC	auto	54.7
	ACE-nwire	B-CUBED	auto	66.4
Nicolae and Nicolae (2006)	ACE-phase2	MUC	auto	63.8
Ng (2007)	ACE-bnews	MUC	auto	64.7
	ACE-bnews	CEAF	auto	61.7
	ACE-npaper	MUC	auto	64.6
	ACE-npaper	CEAF	auto	61.5
	ACE-nwire	MUC	auto	63.3
	ACE-nwire	CEAF	auto	63.6
	ACE-02	MUC	auto	64.2
	ACE-02	CEAF	auto	62.3
Ponzetto and Strube (2006)	ACE-bnews 03	MUC	auto+keys	69.5
	ACE-nwire 03	MUC	auto+keys	71.7
Luo et al. (2004)	ACE-dev	ACE-value	true mentions	89.8
	ACE-feb02	ACE-value	true mentions	90.0
	ACE-sep02	ACE-value	true mentions	88.0
Ji et al. (2005)	ACE 2004	MUC	true mentions	82.4
Nicolae and Nicolae (2006)	ACE-phase2	MUC	true mentions	89.6
Ponzetto and Strube (2006)	ACE (bn+nw) 03	MUC	true mentions	70.7
Denis and Baldridge (2007)	ACE-bnews	MUC	true mentions	69.2
	ACE-npaper	MUC	true mentions	72.5
	ACE-nwire	MUC	true mentions	67.5
Culotta et al. (2007)	ACE 2004	B-CUBED	true mentions	79.3
Haghighi and Klein (2007)	ACE 04 nwire	MUC	true mentions	64.2
	ACE 04 bnews	MUC	true mentions	62.3

Table 3: Results comparative for ACE.

is a line of research to follow. Finally, an interesting open line is the research of unsupervised and weakly supervised approaches, since the scarce availability of annotated corpora for training and test is a bottleneck for the research in supervised technology.

References

Abraços, J. and J.G. Lopes. 1994. Extending DRT with a focusing mechanism for pronominal anaphora and ellipsis resolution. *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 1128–1132.

Azzam, Saliha, Kevin Humphreys, and Robert Gaizauskas. 1998. Evaluating a focus-based approach to anaphora resolution. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics*, pages 74–78, Morristown, NJ, USA. Association for Computational Linguistics.

Azzam, Saliha. 1996. Resolving anaphors in embedded sentences. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 263–268, Morristown, NJ, USA. Association for Computational Linguistics.

Bagga, A. and B. Baldwin. 1998. Algorithms for scoring coreference chains. *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC98)*, pages 563–566.

Baldwin, B. 1997. CogNIAC: high precision coreference with limited knowledge and linguistic resources. *Proceedings of the ACL*, 97:38–45.

Bean, D., E. Riloff, S. Dumais, D. Marcu, and S. Roukos. 2004. Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. *HLT-NAACL 2004: Main Proceedings*, pages 297–304.

Brennan, Susan E., Marilyn W. Friedman, and Carl J. Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 155–162, Morristown, NJ, USA. Association for Computational Linguistics.

Carbonell, J.G. and R.D. Brown. 1988. Anaphora resolution: a multi-strategy approach. *Proceedings of the 12th conference on Computational linguistics-Volume 1*, pages 96–101.

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

Carter, David Maclean. 1986. A shallow processing approach to anaphor resolution. Technical Report UCAM-CL-TR-88, University of Cambridge, Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom, phone +44 1223 763500, May.

Carvalho, Ariadne Maria Brito Rizzoni. 1989. *Logic grammars and pronominal anaphora*. Ph.D. thesis, Berkshire, UK, UK.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Foris, Dordrecht.

Cormack, S. 1993. *Anaphora Resolution in Discourse Representation Theory*. Ph.D. thesis, PhD thesis, University of Edinburgh.

Culotta, A., M. Wick, and A. McCallum. 2007. First-Order Probabilistic Models for Coreference Resolution. *Proceedings of NAACL HLT*, pages 81–88.

de Salvo Braz, R., E. Amir, and D. Roth. 2005. Lifted First-Order Probabilistic Inference. *IJCAI*.

Denis, P. and J. Baldridge. 2007. Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. *Proceedings of NAACL HLT*, pages 236–243.

Faloutsos, Christos and Douglas W. Oard. 1995. A survey of information retrieval and filtering methods. Technical report, College Park, MD, USA.

Finley, T. and T. Joachims. 2005. Supervised clustering with support vector machines. *ACM International Conference Proceeding Series*, 119:217–224.

Ge, N., J. Hale, and E. Charniak. 1998. A statistical approach to anaphora resolution. *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170.

Grishman, R. and B. Sundheim. 1996. Message Understanding Conference-6: a brief history. *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 466–471.

Grosz, Barbara J., Aravind K. Joshi, and Scott Weinstein. 1983. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics*, pages 44–50, Morristown, NJ, USA. Association for Computational Linguistics.

Haegeman, L. 1994. *Introduction to Government and Binding Theory.* Blackwell Publishers.

Haghighi, Aria and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 848–855, Prague, Czech Republic, June. Association for Computational Linguistics.

Hahn, Udo and Michael Strube. 1997. Centering in-the-large: computing referential discourse segments. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 104–111, Morristown, NJ, USA. Association for Computational Linguistics.

Harabagiu, S.M., R.C. Bunescu, and S.J. Maiorano. 2001. Text and knowledge mining for coreference resolution. *North American Chapter Of The Association For Computational Linguistics*, pages 1–8.

Hearst, Marti A. 1999. Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10, Morristown, NJ, USA. Association for Computational Linguistics.

Hobbs, Jerry R. 1977. Pronoun resolution. *SIGART Bull.*, (61):28–28.

Humphreys, K., R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. 1998. University of sheffield: Description of the lasie-ii system as used for muc-7. *Proceedings of the Seventh Message Understanding Conferences (MUC-7).*

Ingria, Robert and David Stallard. 1989. A computational mechanism for pronominal reference. In *Meeting of the Association for Computational Linguistics*, pages 262–271.

Ji, H., D. Westbrook, and R. Grishman. 2005. Using semantic relations to refine coreference decisions. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 17–24.

Kameyama, M. 1997. Intrasentential Centering: A Case Study. In *eprint arXiv:cmp-lg/9707005*, pages 7005–+, July.

Kamp, H. and U. Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer Academic.

Kennedy, C. and B. Boguraev. 1996. Anaphora for everyone: pronominal anaphora resolution without a parser. *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 113–118.

Klenner, M. and É. Ailloud. 2008. Enhancing Coreference Clustering. In *Proceedings of the Second Workshop on Anaphora Resolution.* WAR II.

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

Luo, X., A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. *Proc. of ACL*, 4:136–143.

Luo, X. 2005. On coreference resolution performance metrics. *Proc. of HLT-EMNLP*, pages 25–32.

Mani, I. 2001. Automatic Summarization. *Computational Linguistics*, 28(2).

McCallum, A. and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. *IJCAI Workshop on Information Integration on the Web.*

McCallum, A. and B. Wellner. 2005. Conditional models of identity uncertainty with application to noun coreference. *Advances in Neural Information Processing Systems*, 17:905–912.

McCarthy, J.F. and W.G. Lehnert. 1995. Using decision trees for coreference resolution. *Proceedings of the Fourteenth International Conference on Artificial Intelligence*, pages 1050–1055.

Mitkov, R. 1998. Robust pronoun resolution with limited knowledge. *Proceedings of the 36th annual meeting on Association for Computational Linguistics*, pages 869–875.

Mitkov, R. 1999. Anaphora resolution: The state of the art. *Unpublished Manuscript.*

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

Mitkov, R. 2003. The Oxford Handbook of Computational Linguistics. *Computational Linguistics*, 30(1).

1998. *MUC-7 — Proceedings of the 7th Message Understanding Conference.* http://www.muc.saic.com/.

Müller, C., S. Rapp, and M. Strube. 2002. Applying Co-Training to reference resolution. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 352–359.

Ng, V. and C. Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7.

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

Ng, V. 2005. Machine Learning for Coreference Resolution: From Local Classification to Global Ranking. *Ann Arbor*, 100.

Ng, V. 2007. Shallow semantics for coreference resolution. *IJCAI 2007*, pages 1689–1694.

Ng, Vincent. 2008. Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing.* EMNLP-08.

Nicolae, C. and G. Nicolae. 2006. Best Cut: A Graph Algorithm for Coreference Resolution. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 275–283.

NIST, US. 2003. The ACE 2003 Evaluation Plan. *US National Institute for Standards and Technology (NIST), Gaithersburg, MD.[online*, pages 2003–08.

Ponzetto, S.P. and M. Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199.

Quinlan, J.R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rich, E. and S. LuperFoy. 1988. An architecture for anaphora resolution. *Proceedings of the second conference on Applied natural language processing, February*, pages 09–12.

Richardson, M. and P. Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1):107–136.

Sebastiani, Fabrizio and Consiglio Nazionale Delle Ricerche. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47.

Sidner, Candace L. 1979. Towards a computational theory of definite anaphora comprehension in english discourse. Technical report, Cambridge, MA, USA.

Singla, P. and P. Domingos. 2005. Discriminative training of Markov logic networks. *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 868–873.

Soon, W.M., H.T. Ng, and D.C.Y. Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.

Strube, Michael. 1998. Never look back: an alternative to centering. In *Proceedings of the 17th international conference on Computational linguistics*, pages 1251–1257, Morristown, NJ, USA. Association for Computational Linguistics.

Tetreault, J. 1999. Analysis of syntax-based pronoun resolution methods.

Turmo, J., A. Ageno, and N. Català. 2006. Adaptive information extraction. *ACM Computing Surveys (CSUR)*, 38(2).

Vilain, M., J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. *Proceedings of the 6th conference on Message understanding*, pages 45–52.

Walker, Marilyn A., Aravind K. Joshi, and Ellen F. Prince. 1998. *Centering Theory in Discourse*. Clarendon Press, Oxford.

Walker, Marilyn A. 1989. Evaluating discourse processing algorithms. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, pages 251–261, Morristown, NJ, USA. Association for Computational Linguistics.

Yang, X., J. Su, and C.L. Tan. 2006. Kernel-based pronoun resolution with structured syntactic knowledge. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 41–48.