APPLYING AN LFG PARSER IN COREFERENCE RESOLUTION: EXPERIMENTS AND ANALYSIS

Pascal Denis
Department of Linguistics
University of Texas at Austin

Jonas Kuhn
Computerlinguistik
Universität des Saarlandes

Proceedings of the LFG06 Conference
Universität Konstanz
Miriam Butt and Tracy Holloway King (Editors)
2006
CSLI Publications
http://csli-publications.stanford.edu

Abstract

In this paper, we explore how LFG analyses as produced by the XLE parser with the English ParGram grammar can be used in a probabilistic coreference resolution system. So far, such systems have mainly relied only on information from surface-based NLP tools, reaching reasonable levels of performance while requiring only small amounts of training data. We compare these surface-based approaches with a first attempt at an LFG-based coreference system and another system using the treebank-trained probabilistic parser by Charniak. Based on the (limited) quantity of training data we used, the performance of all three approaches was quite comparable. However, there are some indications that an XLE-based approach may lead to better results if trained on larger training sets.

1 Introduction

The XLE parser coupled with the LFG grammars from the ParGram project and the log-linear disambiguation models developed at PARC (Riezler *et al.*, 2002) is one of the best available parsing systems – in particular if criteria such as depth of analysis and linguistic motivation are taken into account. One of the hopes with such a carefully engineered parsing system is that it can improve the performance of Natural Language Processing (NLP) systems on tasks that have so far been tackled mainly with linguistically unsophisticated, surface-based approaches. The work in this paper is the beginning of an exploration of the impact of using XLE analyses for machine learning based coreference resolution. The contribution of XLE on this task is compared with that of two shallower NLP tools for grammatical analysis, namely Charniak's parser and a simple part-of-speech tagger.

Coreference resolution (CR) provides an interesting testbed for such a comparative study. On the one hand, deep linguistic representations have been largely unexplored by researchers working in robust CR. Thus, most state-of-the-art machine learning systems (McCarthy and Lehnert, 1995; Morton, 2000; Soon *et al.*, 2001) rely on limited and rather shallow knowledge sources. (Some notable exceptions are (Ng and Cardie, 2002b), and more recently (Uryupina, 2006).[1]) Often, the only type of linguistic processing used is part-of-speech tagging and NP chunking. Even at this shallow level of processing and with limited sets of learning features, these systems have managed to achieve reasonably good performances with F-scores in the 60's%. This situation is somewhat at odds with the work of theoretical linguists who have identified numerous linguistic factors bearing on coreference resolution. It is worth noting in this respect that XLE makes a natural candidate for CR: the type of representations it outputs (basically, LFG f-structures) indeed gives us access to many of these factors. An obvious example are *grammatical functions*: at the center of the LFG architecture, they have been also been argued within Centering Theory (Grosz *et al.*, 1995) to play a decisive role in constraining coreference.

On the other hand, there is no guarantee that appealing to deep linguistic systems (and an extended feature set) for CR is likely to improve compared to a surface-based system. There are two issues here, one is theoretical, the other more practical. On the theoretical side, CR is ultimately an AI-complete problem: in the general case, the task involves solving extralinguistic problems for which even a perfect linguistic oracle would not help; that is, linguistic information gives us only *partial* insight. While it is true for all sub-tasks of interpretation that humans will fall back on world and

[1] (Preiss, 2002) compares Charniak and Collins parsers, but the scope of her study is rather limited, since it only deals with anaphora resolution and is not evaluated on available corpora.

situation knowledge to resolve linguistically underdetermined cases, the situation for CR may be particularly challenging for linguistic approaches since the space of possibilities left open after considering linguistic constraints is still quite considerable.[2] This in turn raises the follow-up question of what sorts of linguistic constraints are actually helpful in modeling the data. On the practical side, it is well-known that for deep linguistic analysis, increased robustness will typically go along with an increased level of noise in the analyses. So, the open question is what level of processing gives us the best results. A related question is whether the combination of information from representations of different depth of analysis will improve things. An interesting aspect of comparing XLE with the Charniak parser is that these two parsers differ not only in terms of the level of sophistication of their outputs (phrase structure trees vs. rich feature structures), but also in terms of their efficiency and robustness. While XLE surely provides more detailed information, this comes at a price: despite some coverage improvements (e.g., in the form of disambiguation and the "back-off" fragment mode, i.e., partial analyses provided for sentences that cannot be parsed completely), XLE is still less robust than Charniak's parser.

In anticipation of the results of the present study, we could not so far observe any significant improvements of overall system performance due to the addition of deeper linguistic information sources. For one thing, this shows that the baseline combination of various surface-oriented information sources established in machine learning-based work on CR already seems to strike a very effective balance of robustness and task-relevant quality, which is not easy to outperform – especially on small training sets. On the other hand, we performed some preliminary meta analyses indicating that larger quantities of training data and a more carefully designed set of learning features may bring out the strengths of deeper information sources.

The rest of this paper is organized as follows. We begin by presenting the task of coreference resolution and the type of machine learning architecture we use to model it. Then, in section 3, we discuss some of the advantages, as well as some of the potential problems, associated with using XLE for CR; there, we also briefly describe how we extracted features out of the XLE output representations. Section 4 presents the experimental set-up. The actual results along with some preliminary analyses of these results are given in section 5.

2 The task of coreference resolution

2.1 Task definition

Coreference Resolution is the automatic detection of text spans in a document that share the same referent in the real world, forming classes of coreferent text spans. Each individual text span is typically known as a *mention*; a class of coreferent mentions is called a *chain*, referring to or describing one *entity*. The present study is concerned with one particular case of coreference, namely *nominal* coreference.[3] As an illustration, the result of applying CR to the following discourse (from the ACE

[2] Note that the type of corpora typically used for training CR systems may make this problem even more acute: the annotations of the MUC and ACE corpora (from the Message Understanding Conferences and the Automatic Content Extraction program, respectively) are often debatable from a linguistic point of view, often stretching the notion of coreference to include phenomena that semanticists would not regard as coreference (e.g., nominal prediction and apposition). (See (van Deemter and Kibble, 2000) for a detailed discussion of the MUC scheme.) Hence in training the systems may have trouble detecting those linguistic generalizations that *do* exist for coreference in the narrower, linguistic sense.

[3] For some recent work on event and abstract entity coreference, see (Byron, 2002).

corpus) in (1a) is given in (1b):

(1) a. [Clinton]$_{m_0}$ told [National Public Radio]$_{m_1}$ that [his]$_{m_2}$ answers to questions about [Lewinsky]$_{m_3}$ were constrained by [Starr]$_{m_4}$'s investigation. [[NPR]$_{m_5}$ reporter Mara Liasson]$_{m_6}$ asked [Clinton]$_{m_7}$ "whether [you]$_{m_8}$ had any conversations with [her]$_{m_9}$ about [her]$_{m_{10}}$ testimony, had any conversations at all."

b. {Clinton$_{m_0}$, his$_{m_2}$, Clinton$_{m_7}$, you$_{m_8}$}$_{e_0}$,
{National Public Radio$_{m_1}$, NPR$_{m_5}$}$_{e_1}$,
{Lewinsky$_{m_3}$, her$_{m_9}$, her$_{m_{10}}$}$_{e_2}$,
{Starr$_{m_4}$}$_{e_3}$,
{NPR reporter Mara Liasson$_{m_6}$}$_{e_4}$

Thus illustrated, the task involves two main steps: (a) the identification of referring mentions,[4] and (b) the partitioning the set of mentions into chains for various entities, i.e. the resolution *per se*. In this paper, we concentrate on the latter task.

2.2 CR as a machine learning problem

Like in other areas of NLP, the last decade of research in coreference resolution has seen an important shift from rule-based systems to systems applying machine learning (ML) techniques (Mitkov, 2002). An important appeal of the latter systems of course lies in their robustness, an important precondition for their integration into larger NLP systems, such as Information Extraction, Question Answering, or Summarization systems.

In a ML setting, the task of coreference resolution is recast as a learning problem, typically a *classification* problem. Specifically, the standard approach for task (b) as addressed in section 2.1 proceeds in two distinct steps (McCarthy and Lehnert, 1995; Morton, 2000; Soon *et al.*, 2001; Ng and Cardie, 2002b,a). For the first step, a *binary* classifier is trained that determines whether or not a *pair* of nominal mentions is coreferential. (If the classifier is probabilistic in nature, it will provide a probability for a pair of mentions being coreferential.) In application, this classifier is applied to (in principle) all pairs of nominal mentions from a document. The task for the second step is to use the pairwise coreferentiality information from step one to construct a consistent partition over the entire set of mentions into chains. Although any clustering algorithm could in principle be used for this, the predominant approach is to make the assumption that a coreferent chain of mentions $m_{i_1}, m_{i_2}, m_{i_3}, \ldots, m_{i_k}$ can be effectively detected by relying only on (the coreferentiality classification of) pairs of textually adjacent mentions from that coreference chain, i.e., $\langle m_{i_1}, m_{i_2} \rangle$ and $\langle m_{i_2}, m_{i_3} \rangle \ldots \langle m_{i_{k-1}}, m_{i_k} \rangle$. (Note that this leaves out the coreferentiality status of $\langle m_{i_1}, m_{i_3} \rangle$, for instance.)[5] In other words, the chain is constructed from a sequence of direct "links" in the text. This means roughly that CR is implicitly reduced to *anaphora resolution* (i.e., the task by which an anaphoric expression is bound to its (unique) antecedent).[6]

[4] Depending on the corpus, these are often restricted to a set of predefined named entities, such as PERSON, LOCATION, ORGANIZATION, etc.

[5] A notable exception is (Kehler, 1997) who uses Dempster's rule of combination to induce a partition from the pairwise classifications.

[6] Note however that in a chain { John, he, his, John, he }, the second mention 'John' is constructed as linked to 'his'.

The most common technique for determining the links for building a chain is for each mention to go backwards in the text, pairing it with preceding mentions, until a pair is hit that is classified as coreferential by step one. (If a probabilistic classifier is used, a probability threshold can be used – e.g., threshold 0.5 to make it equivalent to a non-probabilistic classifier.) This technique is called "Closest-First" selection (e.g., (Soon *et al.*, 2001)). An alternative is to compare the (probabilistic) classifier scores for pairs from a larger text window, picking the highest-scoring pair (above a threshold, typically 0.5) to form the link. This is called "Best-First" selection ((Morton, 2000; Ng and Cardie, 2002b)). Other points of divergence exist between these systems, but they mainly concern the feature set that is used and the sample selection, i.e., the choice of actual training data from the vast number of possibilities arising from arbitrary combination of mentions in the text.[7] Some systems also use separate classifiers for different types of mentions (e.g., pronouns and proper names) (e.g. (Morton, 2000)), instead of using a single classifier.

An issue with these selection strategies as just described is that there is no treatment for mentions in the text that introduce a new referent (i.e., which form the beginning of a new chain). One could use an additional classifier that will say for each mention whether or not it is anaphoric and use the described linking technique only for mentions that are anaphoric (see e.g., (Ng and Cardie, 2002a)). As an alternative solution, (Morton, 2000) changes the original classification task from step one in such a way that non-anaphoric elements are included in the training data as being coreferential with an artificial dummy element. In application, a mention will start a new chain if the dummy element is the most probable antecedent.

2.3 Model used in this study

In this preliminary study, we used the set-up proposed by (Soon *et al.*, 2001), which is arguably one of the simplest architectures: it uses a simple sample selection method and a "Closest-First" clustering. The actual training and test procedures for this system are explained below. The main difference with the original Soon *et al* system is in the type of machine learners we used. While (Soon *et al.*, 2001) use Decision Trees, we use maximum entropy (aka, log-linear) models (Berger *et al.*, 1996). More specifically, our coreference model takes the following form, where the classes YES and NO stand for "corefer" and "don't corefer", respectively; m_i and m_j are two mentions, f_i are the features of the model and λ_i their associated parameters:

$$P(\text{YES}|\langle m_i, m_j\rangle) = \frac{exp(\sum_{i=1}^{n} \lambda_i f_i(\langle m_i, m_j\rangle, \text{YES}))}{\sum_{c\in\{\text{YES,NO}\}} exp(\sum_{i=1}^{n} \lambda_i f_i(\langle m_i, m_j\rangle, c))} \quad (2)$$

Parameters were estimated using the limited memory variable metric (LMVM) algorithm implemented in the Toolkit for Advanced Discriminative Modeling (Malouf, 2002).[8] We regularized our model using a Gaussian prior of variance of 1000 — no attempt was made to optimize the prior for each data set. Maxent models are well-suited for the coreference task, because they are able to handle many different, potentially overlapping learning features without making independence assumptions.

[7] Because coreference is a very "rare" relation, looking at all possible pairs of mentions yield a very skewed class distribution.

[8] Available from `tadm.sf.net`.

Previous work on coreference using maximum entropy includes (Kehler, 1997; Morton, 1999, 2000).[9]

For the LFG audience, it may be interesting to note that there is a close parallelism between the Maxent approach and Optimality Theory (compare also (Johnson, 1998; Goldwater and Johnson, 2003)): one can can think of OT as a restricted class of a binary classifiers, where the learning features are called *OT constraints* and a tableau of n candidates corresponds to n classifier decisions. For the coreference task, the OT input would be a particular mention for which we seek an appropriate "linking point", i.e., preceding mention. Each candidate is a pair of the input mention and a potential antecedent. Now harmony evaluation – based on the constraint violation profile of the candidates and the ranking in the grammar – will determine the harmony for each candidate and output the most harmonic one as the winner, i.e., the predicted link. The main difference between OT and the more general Maxent model used in our work is that OT assumes a *strict ranking* of the constraints: that is, lower-ranked constraints are not allowed to "gang up" to beat an higher-ranked constraint. The weighting of the parameters in the Maxent model (= the "strength" of the violable "constraints") is less restricted so that ganging-up effects can happen.

The training and testing procedures proposed in (Soon *et al.*, 2001) are as follows. For training, the text is scanned from left to right and for each anaphoric mention α: (i) a *positive instance* is created between α and its *closest* antecedent m_i, (ii) *negative instances* are created between α and all the (non-coreferential) mentions m_j intervening between α and m_i.

Once trained, the classifier is used to build coreference chains in the following way. For each mention m_i in the text, the preceding text is scanned from right to left, generating pairs of m_i with each of its preceding mentions m_j. Each such pair is submitted to the classifier, which returns a number between 0 and 1 representing the probability of the two mentions to be coreferential. (Soon *et al.*, 2001) use "Closest-First" clustering, which means that the process terminates as soon as the first coreferring mention (i.e., one with probability > 0.5) is found or the beginning of the text is reached.

2.4 Potential limitations of the classification approach

There are at least two potential limitations to the classification approach, both related to the very strong independence assumptions. First, the classifier considers antecedent candidates independently from each other, since only a *single* candidate pair is evaluated at a time. An alternative allowing different NP candidates to be directly compared is to use a *ranker*; this option is explored for pronoun resolution by (Denis and Baldridge, 2007). A second possible limitation has to do with the clustering used: the "Closest-First" and "Best-First" selection algorithms are extremely greedy. They assume that coreference decisions for chain building are independent from one another (McCallum and Wellner, 2003). To take a simple example, consider the following set of mentions {Mr Clinton, Clinton, he}. Under a pairwise classification scenario, the decision regarding the pair ⟨Clinton, he⟩ is done independently from the decision regarding the pair ⟨Clinton, Mr Clinton⟩, although this earlier decision is likely to provide important information for the second decision (e.g., that Clinton is a male). An attempt to solve this problem is provided by (Morton, 2000) and relies on using a discourse model. But this approach is again likely to be greedy, since mistakes made at the beginning are likely to propagate.

[9] In the context of XLE, Maxent models have been used by (Johnson *et al.*, 1999) and (Riezler *et al.*, 2002) for parse selection.

3 Incorporating XLE information

In this section, we motivate the use of XLE for CR by examining a simple example taken from the ACE corpus (from the Automatic Content Extraction program). We also come back to some potential issues that arise when using a deep parser such as XLE. Finally, we discuss the strategy used to extract features from XLE output representations.

3.1 Motivation

The main advantage given by using XLE lies in the richness of the output representations returned by this parser. These representations are rich enough to give us (at least indirect) access to many of the relevant factors identified by linguists as influencing anaphora resolution. In particular, they provide us with morpho-syntactic information (via gender, number, person, and case attributes), syntactic information (via grammatical functions and f-structure configurations[10]), as well as shallow lexical semantics (in the form of animacy, count/mass attributes). As is well-know, an interesting aspect of grammatical functions (GFs) is that they are also correlated to some degree with salience, therefore also giving some partial access into pragmatics. Thus, certain GFs (e.g., subjects) often make more likely antecedents than others. Furthermore, certain "transitions" over GFs (e.g. subject-subject, subject-object) are also potentially useful for coreference in giving us shallow access to discourse structure: parallelism (or contrast) can to a certain extent be captured at the level of grammatical functions. For these reasons, GFs (along with f-structure "paths") will provide most of the features in this pilot study.

To show the importance of GFs for CR, consider the following example from the ACE data:

(3) [He]knew [Brosius]was coming off a bad year, and **he** knew Brosius would be in line to make a decent salary.

In the context of this example, the pronoun he could be resolved to two mentions, namely either Brosius or the preceding pronoun He. Based on surface-based features alone, the pronoun is likely to be resolved to Brosius, since this expression is the closest mention corresponding to the same type of named entity (i.e., a person). Access to the XLE analysis in figure 1 provides us with information that may lead to a better prediction. Intuitively, the first pronominal mention is more "salient" than than the proper name; this is encoded grammatically by the fact that that mention is the subject of the main clause. Note that there is also a parallelism effect here, since the same subject is maintained across clauses. Features based on grammatical functions and transitions over grammatical functions appear to have a potential for correcting some of the mistakes that would be made by simply relying on surfacy features.

3.2 Potential Issues

However, there are a number of places where things can go wrong when using the outputs of deep parsing systems such as XLE. For one thing, statistical disambiguation potentially introduces a level

[10] In LFG, binding principles are stated in terms of the notion of f-command, a relation which is defined directly over f-structures.

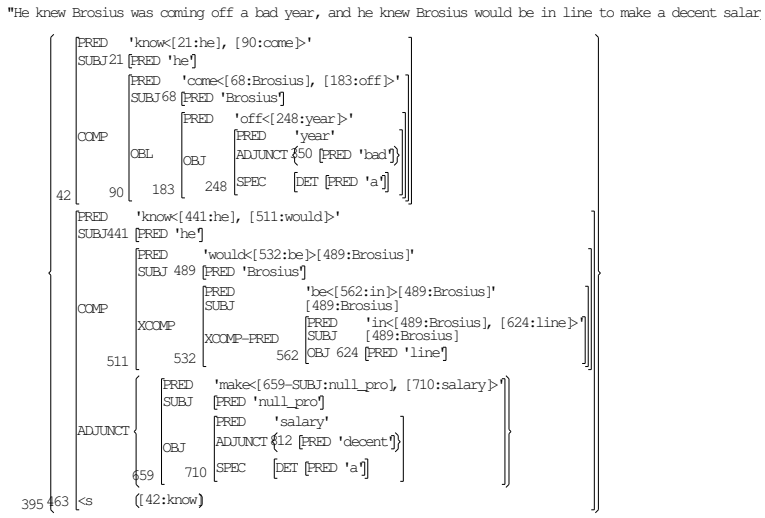

Figure 1: XLE output for sentence (3)

of noise by potentially filtering out correct analyses. This is likely to affect us, since here we only consider the unique *most probable* parse for each sentence (rather than the whole parse forest or even the n best parses). Second, XLE simply fails to produce output for some sentences. In our experiments, we found no parse for 4.3% and 5.1% in training data and in test data, respectively.[11] By comparison, note that Charniak parser only missed less than 0.1% in both the training data and the test data.[12] Also of interest is the fact that the XLE parser outputs "fragment" parses for a significant number of the parsed sentences: 24.4% in training data and 24.7% in test. This in turn raises the following questions: (i) for XLE, to what extent will the additional precision gained by using the XLE representations be able to out-weigh the noise, and (ii) more generally, are the richer outputs still more useful than shallower, but more robust, representations?

As a concrete illustration of these issues, consider the case of grammatical functions and the problem of their identification. GFs can potentially be identified (or at least approximated) using representations reflecting various levels of processing. But crucially, the shallower the processing is, the higher the recall of the identification will be, but the lower its precision will be. Thus, GFs in English at least can be first approximated in terms of part-of-speech (POS) contexts: for instance, subjects are often found before a verbal form, objects after a verbal form, while obliques tend to occur after prepositions. While entirely robust, this strategy of solely relying on linear order is prone to make many errors. For instance, some embedded NPs will be wrongly identified as subjects (say, in relative clauses), while others will be wrongly treated as obliques (say, in PP modifying a head noun). Some of these errors will be handled by going one level up in terms of linguistic processing, and using actual phrase structure configurations to capture GFs: e.g., `[S [NP VP]]` vs. `[S [VP [NP]]]` for the subject/object contrast. While more reliable, these representations are harder to obtain with precision: these sorts of configurations are more reliable, but they are not error-free: e.g., the first NP in a dative-shift construction will be wrongly treated as a direct object. At the end of the spectrum, in XLE GFs

[11] We used one of the latest releases of XLE (June 18, 2006) and of the English grammar (December 5, 2005). The parser was used with its default parameters.

[12] We used the August 16, 2005 version (`ftp.cs.brown.edu/pub/nlparser/`); we also used the default parameters.

can be simply read off the output as attributes, but the problem here is that they might not always be available.[13] There is no general answer to the question how the trade-off between robustness and quality will affect a particular practical performance task.

A possible issue we are facing for the more linguistically sophisticated approaches lies in the set of learning features: too small a feature set might not give us enough to properly model the data. We are likely to suffer from this problem, since we only focus on GFs and GF paths here. One of the main goals of the present study was to set up the machinery for incorporating rich linguistic learning features in the CR task. There is a large space of sophisticated features and feature combination that should be carefully explored.

In our experiments we are also only doing manual feature selection (i.e., filtering of the vast number of feature combinations that are possible), which is typically inferior to automatic feature selection techniques.

Finally, there are some potential issues of a more fundamental kind. We mentioned the AI-completeness point in the introduction already. For a linguistics-rich approach this means that even perfect syntactic information may have a limited effect on performance. Since the various linguistic factors involved in coreference resolution are not sufficient for specifying a deterministic procedure, it is not necessarily the case that the richer linguistic information sources (with the unavoidable noise in the output of any parser) can add task-relevant information that is not already accessible in a more surface-oriented approach. Surface-oriented approaches may actually have an advantage picking up patterns correlating to extra-linguistic factors, without an intermediate representation that may add noise.

A further potential issue has to do with the size of the training data: for the surface-oriented learning features used in most machine-learning based work on CR, learning curves show that already a relatively small quantity of training data already provides sufficient information to acquire the relevant generalizations. Performance figures tend to plateau when adding more training data. Now, if more sophisticated features (and in particular combinations of features) are used, it is quite possible that considerably larger training sets would be required to pick up certain patterns. In our experiments, a number of features that appear interesting from a theoretical point of view were only instantiated in very few training examples; so, data sparsity issues are likely to influence the results.

Somewhat related to the previous two issues, we may note that the CR task is of a somewhat peculiar nature: a considerable proportion of the coreference linking decisions are almost trivial, some of the remaining decisions follow clear linguistic patterns, but a fairly large proportion is controlled by a highly complex interaction of constraints. Thus, a surface-oriented approach has a fair chance of getting up to a certain level of performance and will even get some of the hard cases right ("by chance", so to speak). Ideally, a more sophisticated approach should keep up the quality of the simple technique for the easier cases, but avoid some of the errors for the harder ones. However due to the complex interactions of factors, picking up certain valid deeper patterns may have the effect of breaking a favorable behavior in certain other cases, which may overall balance out the gain from deeper insights.[14]

[13] Note a final advantage of XLE: since GFs are not tied in LFG to particular structural configurations, the strategy used for their identification will work for other languages.

[14] For instance, a surface-based system will typically exclude person-shifts as shown for e_0 (Clinton) in (1). But a more sophisticated system may pick up circumstances under which they *are* possible. It is quite likely however that this pattern will overgenerate to some extent, thus leading to misclassifications in cases considered almost trivial with a surface-based system.

Data-set	train	test
BNEWS	216	51
NPAPER	76	17
NWIRE	130	29

Table 1: # of documents

Dataset	train	test
BNEWS	3740	950
NPAPER	2453	615
NWIRE	2724	608

Table 2: # of sentences

Dataset	train	test
BNEWS	10086	2608
NPAPER	11410	2504
NWIRE	10868	2630

Table 3: # of mentions

3.3 From the XLE output to learning features

How do we extract information from XLE output for creating our features? Among the various formats available, XLE outputs its analyses in Prolog, where c-structure subtrees and f-structure constraints are represented as lists of Prolog facts. (The mapping function ϕ from c- to f-structure is also captured this way.) This is illustrated in figure 2, which is the output for sentence (3).

More specifically, these representation encode: (i) the character offsets of each token, (ii) the c-structure projections for each token as well as the mapping from each subtree to its f-structure node, and finally (iii) the constraints associated with each f-structure node (i.e., a full description of the f-structure). The way we were able to map each mention to its corresponding f-structure was first unpacking the different Prolog facts into various data structures, then mapping the different tokens making up the mention to their corresponding surface forms in the XLE representation. Once identified, the different surface forms could be mapped to an actual f-structure node (and to the associated set of AVMs). In the case of multi-word mentions, the highest node in the graph, i.e., the node corresponding to the maximal projection, was used. Each mention is furthermore associated with a f-structure path from the main (i.e., ROOT) f-structure to its f-structure node.

4 Experimental setup

In order to evaluate the contribution of XLE for the coference task, we ran comparative experiments with various feature sets extracted from analyses provided by the three different "syntactic" analyzers with different depths of processing and degrees of robustness: (i) a part-of-speech tagger (we used OpenNLP Maxent POS tagger), (ii) a Penn Treebank trained phrase structure parser (namely, the Charniak parser), and (iii) XLE parser, which is a full-blown implementation of LFG.

4.1 Corpus and evaluation

For training and evaluation, we used the datasets from the ACE corpus (Phase 2). This corpus is composed of three parts, corresponding to different genres: broadcast news transcripts (BNEWS), newspaper texts (NPAPER), and newswire texts (NWIRE).[15] Each of these is split into a `train` part and a `devtest` part. We used the `devtest` material only once, namely for final testing. Progress during the development phase was estimated only by using cross-validation on the training set for the NPAPER section. Statistics for the different datasets are given in tables 1-3.

In our experiments, we restricted ourselves to the *true* ACE mentions, i.e., rather than trying to identify candidate phrases for coreference resolution automatically (task (a) addressed in section 2.1), we

[15]The mentions in ACE2 are restricted to 7 types of entities:FACility, GPE (geo-political entity), LOCation, ORGanization, PERson, VEHicle, WEApons.

```
fstructure('He knew Brosius was coming off a bad year, and
            he knew Brosius would be in line to make a decent salary.',
% Properties:
[
'xle_version'('XLE release of Aug 15, 2006 15:04.'),
...
'statistics'('15+30 solutions, 0.83 CPU seconds, 1119 subtrees unified'),
'rootcategory'('ROOT')
],
...
],
% Constraints:
[
cf(1,in_set(var(1),var(0))),
cf(1,in_set(var(22),var(0))),
cf(1,eq(attr(var(0),'COORD'),'+_')),
...
cf(1,eq(attr(var(1),'PRED'),semform('know',2,[var(19),var(3)],[]))),
cf(1,eq(attr(var(1),'SUBJ'),var(19))),
cf(1,eq(attr(var(1),'COMP'),var(3))),
...
cf(1,eq(attr(var(19),'PRED'),semform('he',0,□,[]))),
cf(1,eq(attr(var(19),'CASE'),'nom')),
cf(1,eq(attr(var(19),'GEND-SEM'),'male')),
cf(1,eq(attr(var(19),'HUMAN'),'+')),
cf(1,eq(attr(var(19),'NUM'),'sg')),
cf(1,eq(attr(var(19),'PERS'),'3')),
cf(1,eq(attr(var(19),'PRON-TYPE'),'pers')),
...
cf(1,eq(attr(var(3),'PRED'),semform('come',11,[var(15)],[]))),
cf(1,eq(attr(var(3),'SUBJ'),var(15))),
...
cf(1,eq(attr(var(15),'PRED'),semform('Brosius',3,[],□))),
cf(1,eq(attr(var(15),'CASE'),'nom')),
cf(1,eq(attr(var(15),'NUM'),'sg')),
cf(1,eq(attr(var(15),'PERS'),'3')),
...
],
% C-Structure:
[
cf(1,subtree(13969,'ROOT',15617,899)),
cf(1,phi(13969,var(0))),
...
cf(1,terminal(460,'he',[441])),
cf(1,phi(460,var(53))),
...
cf(1,terminal(77,'Brosius',[68])),
cf(1,phi(77,var(15))),
...
cf(1,terminal(38,'he',[21])),
cf(1,phi(38,var(19))),
...
cf(1,surfaceform(68,'Brosius',9,16)),
cf(1,surfaceform(42,'knew',4,8)),
cf(1,surfaceform(21,'^ he',1,3))
]).
```

Figure 2: XLE Prolog (abbreviated) output for sentence (3)

relied on the gold standard phrases/mentions marked manually in the corpus annotation. We made this decision because our focus is on comparing features between different knowledge sources, rather than on building a full-fledged resolution system. It is worth noting that previous work tends to be vague about mention detection: details on mention filtering or providing performance figures for identification are rarely given.

Following common practice in coreference resolution, we report our main results in terms of Recall-Precision at the level of chains partitioning the set of all mentions in the text. In particular, we use the model-theoretic metric proposed by (Vilain *et al.*, 1995). This method operates by comparing the equivalence classes defined by the resolutions produced by the system with the gold standard classes: these are the two "models". Roughly speaking, the scores are obtained by determining the minimal perturbations needed to transform one model into the other model. Recall is computed by trying to transform the predicted chains into the true chains, while precision is computed the other way around.

4.2 Feature sets

Overall, we actually used four systems, based on four different feature sets. In our baseline feature set, we used features obtainable from shallow processing; the corpus was preprocessed with the OpenNLP Toolkit[16], which includes a sentence detector, a tokenizer, and a POS tagger. These features include **NP type** features for the anaphor candidate and the antecedent candidate (i.e., whether the mention is a pronoun, a proper name, a definite description, etc.), **locality** features (encoded in the form of various distance features), **morpho-syntactic agreement** features (i.e., gender, number, and person compatibilities), **semantic compatibility** features (this is captured in terms of the named entity types), salience-based features (e.g., number of times a mention has been seen in the previous context), as well a number of **ad hoc features** for specific NP types (e.g., string matching, apposition and acronym). These features are summarized in table (4.2).

In addition to the simple features described above, we used various composite features by "crossing" some of the basic features above. For the baseline feature set, we simply combined distance features with the type of the anaphor (e.g., pronoun, definite NP, proper names).

The second feature set expands on the baseline by encoding more linguistically-motivated features (mainly features appromixating GFs), but which are based solely on the outputs of the POS tagger. The third feature set incorporates features that use the output of Charniak, while the fourth feature set includes features derived from the XLE output. With both parsers, we used the unique most probable parse for each sentence. These new features fall into four main categories: **GF**, **GF transitions**, **Binding**, and **Syntactic context**. They are presented in detail in the form of templates in table 4.2.

In addition to these base features described above, we added composite features of the following types: (i) distances and GFs, (ii) distances and syntactic context of the antecedent candidate, (iii) distances and binding, (iv) anaphor type and syntactic embedding of the antecedent candidate, and (v) distances, anaphor type, and syntactic context of the antecedent.

[16] Available from `opennlp.sf.net`.

Feature type	Feature Name	Description
NP type	ANA_PRO	T if m_i is a pronoun; else F
	ANA_SPEECH_PRO	T if m_i is a speech pronoun; else F
	ANA_REFL_PRO	T if m_i is a refl. pron.; else F
	ANA_PN	T if m_i is a PN; else F
	ANA_DEF	T if m_i starts with *the*; else F
	ANTE_PRO	T if m_j is a pronoun; else F
	ANTE_PN	T if m_j is a PN; else F
	ANTE_DEF	T if m_j starts with *the*; else F
	ANTE_INDEF	T if m_j starts with *a(n)*; else F
Locality	S_DIST	binned values for S distance between m_i and m_j
	NP_DIST	binned values for NP distance between m_i and m_j
Morphosynt.	NUM_AGR	T if m_i and m_j agree in number; else F
Agreement	GEN_AGR	T if m_i and m_j agree in gender; else F;
Salience	ANA_M_CT	# of times m_i has been seen
String match	STR_MATCH	T if the strings of m_i and m_j match; else F
Semantic	NE_AGR	T if m_i and m_j correspond the same NE; else F
Agreement	ANTE_NE_&_ANA_GEN	the NE of m_j and the gender of m_i
Quotes	ANA_IN_QUOTES	T if m_i is within quotation marks; else F
	ANTE_IN_QUOTES	T if m_j is within quotation marks; else F
Acronym	ACRONYM	T if one NP is an acronym of the other; else F
Apposition	APPOSITION	T if m_i is an apposition of m_j; else F

Figure 3: Baseline feature set

5 Results and Analysis

This section presents the results of our various experiments, as well as some initial elements of analysis. Table 5 summarizes the results of our main experiment on the three ACE datasets.

The results tell us a number of things. First, the addition of the new features appears to yield a small drop in overall f-score; the differences are however not statistically significant (at $p < .01$) for any of the feature sets. Second, the actual pattern found for the different feature sets is that the addition of the new features produces a gain in recall, but this gain is accompanied by a corresponding drop in precision. From our statistical testing, we however found that although the decreases in precision were significant for all the features (at $p < .01$), the increase in recall is significant only with the XLE features. How do we interpret these results? One can start by considering more closely the different types of errors made by the new systems. One can break down the types of mistakes made by a CR system into three categories: (i) *missing* mentions (i.e., mentions that are not treated as anaphoric when they should), *spurious* mentions (i.e., mentions that treated as anaphoric when they should not), (ii) (correctly identified anaphoric) mentions that are *wrongly resolved*. The first two categories concern the (non-)anaphoricity of a mention, while the third one concerns the resolution *per se*. Also note that the first category only affects recall (these are the false negatives), the second category only affects precision (these are the false positives), while the latter affects both. Looking first at the distributions of the different types of mistakes in the baseline, one first finds that almost 2/3 of the recall mistakes are due to missing anaphoric mentions (the other third is due to wrong resolutions). On the precision side, one finds the opposite pattern: only 1/3 of errors are due to spurious anaphora. As for the effect of the new features, one finds that 2/3 of the recall error reduction comes from a reduction of the missing anaphora; that is, only 1/3 comes from rectifying wrong resolutions.

Looking at the actual predictions, one finds that the XLE features allow the system to identify new,

Feature Type	Feature Name	Description
GFs	ANA_SUBJ	m_i has subject POS context/tree config./SUBJ attr.
	ANA_OBJ	m_i has object POS context/config./OBJ attrib.
	ANA_OBL	m_i has oblique POS context/tree config./OBL attr.
	ANTE_SUBJ	m_j has subject POS context/tree config./SUBJ attr.
	ANTE_OBJ	m_j has object POS context/tree config./OBJ attr.
	ANTE_OBL	m_j has oblique POS context/tree config./OBL attr.
GF	BOTH_SUBJ	m_i and m_j are both subjects
Transitions	SAME_GR	m_i and m_j have the same GF
Binding	C-/F-COMMAND	m_j c-/f-commands m_i
Context	ANTE_PATH_SUFFIX_N	last n nodes (n in {1,2,3}) in m_j's FS/tree path[17]
	ANA_PATH_SUFFIX_N	last n nodes (n in {1,2,3}) in m_i's FS/tree path
	ANTE_PATH_LN	binned value for number of nodes in m_j's FS/tree path

Figure 4: New feature templates

	BNEWS			NPAPER			NWIRE			Overall		
Feature Set	R	P	F	R	P	F	R	P	F	R	P	F
Baseline	53.4	**84.0**	65.3	55.8	**84.3**	**67.1**	51.6	**80.5**	62.9	51.6	**80.5**	62.9
Tagger	54.5	80.6	65.1	56.6	81.0	66.6	53.2	78.7	63.5	53.2	78.7	63.5
Charniak	55.6	79.8	65.5	56.1	80.3	66.1	54.8	80.0	**65.0**	54.8	80.0	**65.0**
XLE	56.2	76.8	64.9	**58.8**	77.1	66.8	55.2	76.0	63.9	55.2	76.0	63.9
All	**57.6**	76.4	**65.7**	57.9	76.9	66.0	**56.3**	76.1	64.8	**56.3**	76.1	64.8

Figure 5: Results for the 3 ACE datasets

more subtle coreferential configurations, but these features tend to be unreliable. To give an illustration of this tendency, note for instance that one finds more correct long distance resolutions, but at the same time one also finds errors showing number and gender mismatches (e.g., $\langle he, she \rangle$, $\langle he, Mrs.Anderson \rangle$).

Although the performances are fairly similar for all the new systems, there is however one dataset where XLE seems to be better than the baseline, namely the NPAPER dataset.[18] Interestingly, it is also on this the corpus that XLE shows the best parsing performances (especially in training), with only 3% (against an average is 4.3%) of parses missing and 18.7% of fragment parses (against an average is 24.4%) for training, and 2% (against an average is 5.1%) of parses missing and 23% of fragment parses (against an average is 24.7%) for test. This would suggest that there is a correlation between the amount of sentences given a full parse and the coreference performances.

A similar conclusion emerges from looking at learning curves for this dataset. These are given for the three feature sets on the in figure 6. These curves are encouraging for XLE in suggesting that this system would benefit the most from additional training data; indeed, it is the only curve among the three that does not appear to converge. This indicates that as speculated in section 3.2, the deeper approaches may benefit more from larger training sets than the surface-oriented approaches.

A final, interesting question is whether the systems did differently for different types of mention. Here, we consider three main types, namely mentions that are headed by a pronoun, a proper name (PN), or a common noun (CN). The results below are given in terms of a slightly different evaluation

[18]The difference is not statistically significant however.

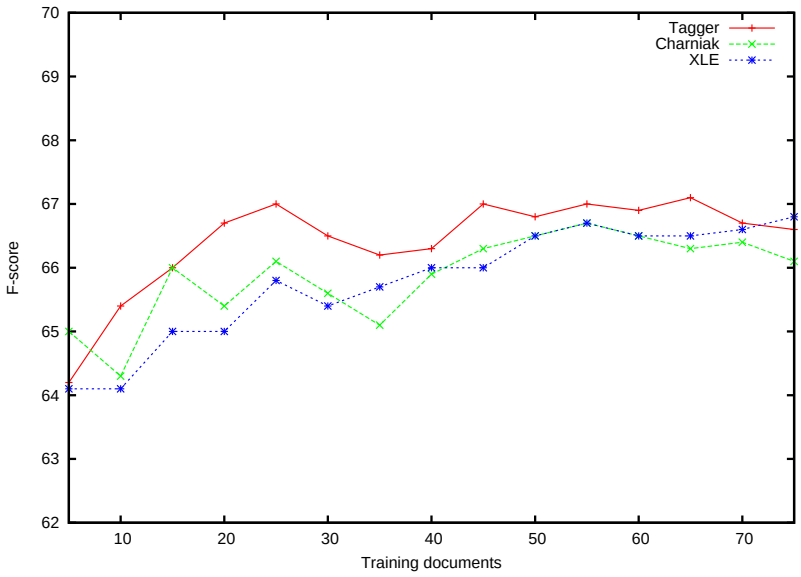

Figure 6: Learning curves for the NPAPER dataset

scheme, namely these are *anaphora resolution* scores. Roughly, one looks at individual links rather than comparing the entire chains.[19] Under this metric, recall is the number of mentions (of the given type) that are correctly resolved divided by the total number of anaphoric mentions (of that same type). And precision is the number of mentions (of the given type) that are correctly resolved divided by the number of mentions that are resolved. The results for the different mention types are given in table 7.

These results are rather inconclusive: the Charniak features seem to make a stronger contribution with pronouns, while the XLE features yield improvements with proper and common nouns. Note that the general low scores for the latter type is explained by the fact that our system does not incorporate a lot of lexical semantic information, which is so critical for these (e.g., definite descriptions).

6 Conclusions and Future Work

By way of various experiments, this study has compared the use of feature sets encoding various depths of linguistic processing for the task of robust coreference resolution. We have in particular compared three main feature sets, extracted from a simple POS tagger, Charniak parser, and the XLE parser. The main conclusions are as follows. The addition of the new features gives rise to an increase in Recall, but don't lead to an overall increase in f-score. We take this to indicate that the new features permit the detection of more coreference configurations, but the extra information is not reliable yet. XLE seems to offer a better improvement potential than Charniak or the POS tagger, but only when it

[19]This type of evaluation is coarser than Vilain's metric in that it misses potential "implicit" links (cf. coreference is an equivalence relation), but it makes it easier to compare different NP types.

	BNEWS			NPAPER			NWIRE		
NP type	R	P	F	R	P	F	R	P	F
Pronouns									
Baseline	67.8	77.0	72.1	**67.2**	**74.8**	**70.8**	60.6	70.8	65.3
Tagger	68.0	76.9	72.1	65.7	72.8	69.1	61.5	71.3	66.0
Charniak	**69.2**	**77.1**	**72.9**	65.2	72.0	68.4	**67.4**	**74.2**	**70.6**
XLE	67.2	75.6	71.1	65.2	66.9	66.1	63.0	69.1	65.9
PNs									
Baseline	47.6	84.6	60.9	56.6	**87.6**	68.8	58.2	87.8	70.0
Tagger	48.4	**84.8**	61.6	56.5	**87.6**	68.7	58.6	**87.9**	**70.3**
Charniak	49.3	84.7	62.3	56.5	87.3	68.6	58.6	87.7	70.2
XLE	**50.5**	82.7	**62.7**	**57.6**	86.6	**69.1**	**59.6**	83.5	69.6
CNs									
Baseline	27.8	**86.0**	**42.0**	25.6	**89.3**	**39.8**	27.4	**75.9**	40.2
Tagger	30.9	64.5	41.8	27.3	61.6	37.8	30.7	63.8	41.5
Charniak	30.3	57.5	39.7	27.0	62.6	37.7	30.2	65.9	41.4
XLE	**33.7**	51.7	40.8	**30.4**	54.0	38.9	**33.2**	61.0	**43.0**

Figure 7: Results per mention types

achieves good parsing performances. XLE also seems more likely to benefit from additional training data.

In section 3.2, we speculated about a lot of potential issues that may preclude a straightforward improvement of the surface-oriented CR techniques by simply adding more linguistically sophisticated knowledge sources. Presumably several of them do hold true. By setting up a flexible system for integrating linguistic resources, we established a basis for further explorations of the interactions.

There are various natural ways to extend this work. First, by using the unique most probable parses, our experiments have not used the two parsing systems to their full potentials. For instance, one would like to take advantage of the "packed" representations provided by XLE, instead of just using a single parse. Second, a lot of extensions are possible regarding feature design: we only scratched the surface in considering only GFs and GF paths. Third, there are more effective ways of combining the different feature sets, instead of just adding them together in a unique model; a better alternative would be to use ensemble models. Finally, there is also the possibility that more "global" models and less greedy search strategies will make better use of the rich features extracted from the deep parses.

References

Berger, A., Pietra, S. D., and Pietra, V. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, **22**(1), 39–71.

Byron, D. K. (2002). Resolving pronominal reference to abstract entities. In *Proceedings of the ACL '02*, pages 80–87.

Denis, P. and Baldridge, J. (2007). A ranking approach to pronoun resolution. In *Proceedings of IJCAI-07*.

Goldwater, S. and Johnson, M. (2003). Learning OT constraint rankings using a maximum entropy model. In J. Spenader, A. Eriksson, , and Ö. Dahl, editors, *Proceedings of the Stockholm Workshop*

on 'Variation within Optimality Theory. April 26-27, 2003 at Stockholm Univ. Sweden, pages 111–120.

Grosz, B., Joshi, A., and Weinstein, S. (1995). Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, **2**(21).

Johnson, M. (1998). Optimality-theoretic Lexical Functional Grammar. In *Proceedings of the 11th Annual CUNY Conference on Human Sentence Processing, Rutgers University*.

Johnson, M., Geman, S., Canon, S., Chi, Z., and Riezler, S. (1999). Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99), College Park, MD*, pages 535–541.

Kehler, A. (1997). Probabilistic coreference in information extraction. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 163–173.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Workshop on Natural Language Learning*, pages 49–55, Taipei, Taiwan.

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

McCarthy, J. F. and Lehnert, W. G. (1995). Using decision trees for coreference resolution. In *IJCAI*, pages 1050–1055.

Mitkov, R. (2002). *Anaphora Resolution*. Longman, Harlow, UK.

Morton, T. (1999). Using coreference for question answering. In *Proceedings of ACL Workshop on Coreference and Its Applications*.

Morton, T. (2000). Coreference for NLP applications. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, Hong Kong.

Ng, V. and Cardie, C. (2002a). Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*.

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

Preiss, J. (2002). Choosing a parser for anaphora resolution. In *Proceedings of DAARC 2002*, pages 175–180.

Riezler, S., Crouch, D., Kaplan, R., King, T., Maxwell, J., and Johnson, M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), Pennsylvania, Philadelphia*.

Soon, W., Ng, H., and Lim, D. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, **27**(4), 521–544.

Uryupina, O. (2006). Coreference resolution with and without linguistic knowledge. In *Proceedings of LREC 2006*, pages 893–898.

van Deemter, K. and Kibble, R. (2000). On coreferring: Coreference in MUC and related annotation schemes. *Computational Linguistics*, **26**(2), 629–637.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In *Proceedings fo the 6th Message Understanding Conference (MUC-6)*, pages 45–52, San Mateo, CA. Morgan Kaufmann.