Identification of Textual Contexts

Ovidiu Fortu and Dan Moldovan

Human Language Technology Research Institute,
University of Texas At Dallas,
Department of Computer Science University of Texas at Dallas,
Richardson, TX 75083-0688, (972) 883-4625
{fovidiu, moldovan}@hlt.utdallas.edu

Abstract. Contextual information plays a key role in the automatic interpretation of text. This paper is concerned with the identification of textual contexts. A context taxonomy is introduced first, followed by an algorithm for detecting context boundaries. Experiments on the detection of subjective contexts using a machine learning model were performed using a set of syntactic features.

1 Introduction

The identification of contexts in text documents becomes increasingly important due to recent interest in question answering, text inferences and some other natural language applications. Consider for example the question "Who was Kerry's running mate in the last US Presidential election?" Here one can identify a temporal context, i.e. last year, and a domain context namely US Presidential election. Proper identification of such contexts may help a Question Answering system locate the right documents, by using contextual indexing for example, and provide correct answers by using contextual reasoning.

Although contexts in natural language were studied for some time, not much work has been done on context taxonomy, boundary detection, and how to use them in language understanding. However, considerable work was done on contexts in AI in general. John McCarthy's work on formalizing the contexts ([1], [2]) provides a general framework for a context-based representation of information. His theory is general and flexible, and as a consequence it can be adapted to formalize some complex aspects of textual contexts.

Consider for example the sentence:

Mary said the dog ate the cake, but I don't believe it. The pronoun "it" refers to an entire sentence, and thus the representation of the phrase above in first order logic is complicated. The use of contexts can simplify the representation:

$\text{Ist}(C_M,$ "the dog ate the cake") $\wedge \neg\ \text{Ist}(C_S,$ "the dog ate the cake"),

where C_M is the context of Mary's perspective, while C_S is the context of the speaker's perspective. We used McCarthy's notation, i.e. $\text{Ist}(c, p)$ means that proposition p is true in context c.

A. Dey et al. (Eds.): CONTEXT 2005, LNAI 3554, pp. 169–182, 2005.

This paper proposes to represent textual contexts as objects, provides a concept taxonomy, and a method for the identification of contexts in text. Experiments were performed on the detection of subjective contexts using a machine learning model.

2 Textual Contexts

There is an important distinction between the concepts in McCarhty's theory and the ones discussed in this paper. While McCarthy's objective was to create a general and flexible theory for Artificial Intelligence applications, the objective of this paper is a lot more narrower. We deal with the problem of identifying contexts and determining the correct boundaries of these contexts. For example:

"John got a new job on Monday. He got up early, shaved, put on his best suit and went to the interview."

The word "Monday" appears only in the first sentence. Although the second sentence has absolutely no reference to any date, a human interprets the text considering that all actions in the second sentence happened on Monday as well. Thus, the temporal information given in the first sentence also applies to the second sentence. To cover both intra-sentence and inter-sentence contexts we refer in this paper to *textual contexts* as a set of clauses or sentences that have some implicit information in common. The commonality is the context we are trying to identify.

For the purpose of text understanding, we need this common information, because it helps in interpreting the sentences, like in the example above. The complexity of the common information shared by the sentences that belong to the same textual context makes the context detection a hard problem. Our solution for coping with this difficulty is to avoid direct detection of complex contexts. Instead, we define very simple classes of contexts (see section 3) and focus on them. Some of the different contexts that we find will overlap (contain

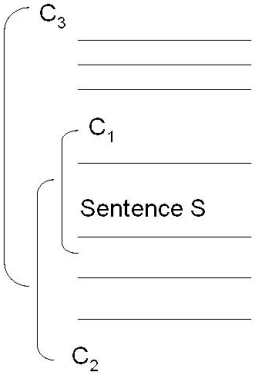

Fig. 1. Example of contexts layout

common sentences). For each sentence, we can use information from all contexts that contain it. The combination of all these pieces of information can be viewed as a more complex context. Thus, even if we do not detect complex contexts, we can reconstruct approximations from the simple ones. Figure 1 shows how a sentence can be contained in several textual contexts. The brackets show the boundaries of the contexts C_1, C_2 and C_3, all of which contain sentence S. For understanding S, we can use simultaneously the information of all the three contexts.

3 Context Taxonomy

As seen above, context structures in discourse can be used for text understanding. In the following, we give a taxonomy of contexts that defines the goals of the context detection task.

Let us consider a very simple abstract example. Let a, and b be predicates and the proposition: $a \wedge b$. One can take a context A such that $Ist(A, a)$, another one B such that $Ist(B, b)$, or AB such that $Ist(AB, a \wedge b)$.

By definition, the logic formulas above are valid, but there may be some other contexts in which a, b and $a \wedge b$ hold. In fact, we can take all combinations of predicates and define contexts - a very large number of contexts (exponential with the number of predicates). We can easily do the same with discourse elements, and thus obtain different, but valid representations of the same thing. The point is that contexts are not an inherent property of a text, like for example the subject of a sentence which is uniquely determined, but rather a way of representing the information in the text.

This is why the detection of arbitrary contexts, even if accurate, may not prove useful. Thus, the need to establish types of contexts which have useful meaning and properties. Once the context types are established, one can explore possibilities for representing information in a text using contexts. For the types of contexts that have been chosen, one can also find properties that allow the detection, and last but not least axioms that allow inferences with contexts.

Our purpose is to identify contexts that are useful for the analysis of text, and thus the types of contexts that we present here are intended for general use. Our criterion for the classification is the nature of the information they contain.

General Objective (Factual) Contexts. These contexts contain statements that are generally accepted as true (such as scientific facts, for example). One can identify them in various types of documents, from scientific papers to news articles. Example:

"Citrate ions form salts called citrates with many metal ions. An important one is calcium citrate or "sour salt", which is commonly used in the preservation and flavoring of food."

Note that even the sentences are general facts, they can not be taken separately without solving the coreferences (pronoun "one", which plays the role of subject in the second sentence, for example).

The sentences associated to these contexts are generally expressed using the present continuous tense. They are also marked by the presence of quantifiers (like "all", or "commonly" in our example). The presence of nouns with high generality (placed close to top in WordNet hierarchy) is also a fact that indicates objective contexts.

Subjective contexts. We introduce this type of contexts to handle feelings, beliefs, opinions, and so on. The truth value of sentences that express such facts is relative to the speaker's state of mind. From the point of view of the real world, the truth value of such sentences may vary from speaker to speaker and from situation to situation. One can identify some subclasses of this type of contexts:

1. Statement contexts, containing statements that people make.
2. Belief-type contexts, representing personal beliefs of the speaker
3. Fictive contexts, dealing with facts that are not real, like dreams and products of imagination.

It is necessary to treat the statements as contexts, because it is possible that different people say contradictory things about the same subject (in fact this is very common in debates or news articles about arguable things). By simply taking the stated facts without considering the contexts, one easily obtains contradictions, which puts an end to inferences.

Belief contexts are very similar to statement contexts, but in general it is less likely that facts that are true in a belief context are also true in the real world. In fact, in many cases people explicitly state that what they say is a belief in order to place a doubt over the truth of what is said. Example:

"John, when is this homework due?
I believe it's on Monday."

John thinks the homework is due on Monday, but he is not certain. He does not refuse to answer the question, but he avoids assuming the responsibility implied by a clear answer. In general, it is implied that people believe what they say (according to Grice's principles). This is why the explicit statement of belief is redundant, and thus an indication of possible falsity.

The main purpose of this classification is to offer support to establishing the relation between what is stated in the text and the reality. From this point of view, the three subclasses can be viewed as a finer grain ordering; the content of a statement is in general more likely to be true than a belief, which in turn is more likely to be true than a fiction.

Probability/Possibility/Uncertainty Contexts. Probability and probable or possible events are handled naturally in human inference and as a consequence, are present in human languages. This is why one can consider the manipulation of information related to possible events using contexts. Example:

"Perhaps Special Projects necessarily thinks along documentary lines. If so, it might be worth while to assign a future jazz show to a different department - one with enough confidence in the musical material to cut down on the number of performers and give them a little room to display their talents.

Time/Space(dimensional) Contexts. Many aspects of our lives are related to time and space constraints. This makes temporal and spatial information frequently present in human reasoning and language. Since propositions that are true at a certain time or in a certain place are not necessarily true outside their time frame or place, it is useful to define time and space related contexts. From the point of view of the limitations they impose, the dimensional contexts can be further divided in:

- Time contexts - mainly temporal restrictions
- Space contexts - location/space restrictions
- Event contexts - combine time and space restrictions

Domain Contexts. These contexts deal with restrictions regarding the domain of applicability of the statement. Example:

John is very good at math, but he is slow at English.

Although these contexts are not as frequent as the time/space based contexts, they are still important for the correct interpretation of the text. The beginning of such a context is generally marked by the presence of constructions with prepositions like "in" and "at" and nouns that denote domains (like "math" above).

Necessity Contexts. These contexts deal with necessary conditions for something to happen. Example:

In order to achieve victory, the team must cooperate. The strikers must come back to help the defenders.

These contexts are marked by the presence of the modal verb "must" or some substitutes like "have to", "need to". Adverbs like "necessary" and nominalizations like "requirement" can also do the same job.

Planning/Wish Contexts. This type of context contains information about someone's plans (or wishes). Example:

After I cash my paycheck, I go to the casino and win a lot of money. I buy a nice car and a big house. Then I quit my job and go on vacation.

The taxonomy presented above covers a wide variety of examples. It is a starting point for solving specific problems. The classes presented here are not necessarily disjoint. In fact, it is possible to have a context that has characteristics from several classes. The issue here is not to accurately classify a given context; our purpose is to identify the textual contexts of types presented above and exploit the information they contain for text understanding. If several contexts of different types overlap (or even coincide), it is not a conflict, but richness in information.

4 Textual Context Identification

4.1 Approach

The atomic unit of text for our problem is a verb (or place-holder like a nominalization) together with their arguments. In general, this unit is a clause. For simplicity, we will use the term "clause" instead of "atomic unit" from now on. A textual context will then be a set of such atomic units (clauses).

The main property of discourse exploited by the context detection mechanism proposed in this paper is the simple fact: utterances that belong to the same context tend to "stick together" in clusters. A *cluster* in a textual context is a maximal set of adjacent clauses that belong to the same textual context. A textual context is composed of one or several clusters. Thus, a natural approach to the context identification problem is first to identify the clusters and then group them to construct full contexts.

The identification of such a cluster is equivalent to the identification of its boundaries. Each clause can be a border for a context, and the problem of identification of contexts is reduced to the problem of deciding whether a given clause is boundary or not. Let us consider the following example:

"We're getting more 'pro' letters than 'con' on horse race betting", said Ratcliff. *"But I believe if people were better informed on this question, most of them would oppose it also. I'm willing to stake my political career on it".*

All clauses of this text belong to the context of Ratcliff's statements, except for the one in non-italics, of course. Note that both the first clause (the border that marks the beginning of the textual context) and the last one have nothing special on their own. They could appear anywhere in text in the same form, without marking the beginning or the end of a context. Thus, the identification of context borders must depend mostly on the characteristics of discourse. However, the phrase in non-italics clearly indicates the presence of a subjective context, regardless of the surrounding clauses.

The flexibility of the natural language makes it possible for contexts to be introduced by countless syntactic patterns. Moreover, different types of contexts are introduced differently (more details in 4.2). Thus, for each type of context we must have a specific procedure for identifying the clues that indicate the presence of the context. For simplicity, these clues will be called *seeds* from now on.

Once we have identified the seeds, we can proceed to the identification of the clusters. The general method for this is to take each seed and try to add to the cluster associated to it the adjacent clauses through an iterative process. During this process, the text span that is associated to the context is growing, starting from the seed; this is the intuitive justification for the use of the term *seed* for designating the clue phrase that signals the existence of a context. Note that we have to consider both clauses that appear before the seed and after the seed to make sure we don't miss anything.

Of course, the process of growing a seeds is not the end. As mentioned above, it is possible that several text spans belonging to the same context are not adjacent. The simplest example is the dialogue; each participant says something, and the utterances of one of the speakers participating in the dialogue are scattered. For solving the problem, we need to compare the clusters obtained and group them as necessary.

In a nutshell, our algorithm for identification of contexts has three stages:

- Stage 1: Identify the seeds.
- Stage 2: Grow the seeds into clusters.
- Stage 3: Group the clusters obtained at the previous step to obtain the full contexts.

Next we provide a detailed description of these steps.

4.2 Seed Identification

The identification of the seeds is the first step towards the detection of contexts. As mentioned above, it is important to point out that different types of contexts are introduced by different seeds; for example, the fact that a certain clause contains the verb "say" has little if any relevance for the detection of a dimensional context. However, if we are trying to detect subjective contexts, the same clause becomes a very powerful indicator of a context presence. Therefore, we take the types of contexts one at a time, and adapt the search procedure for each one. We can not cast the problem as a multi-classification problem, having a class for each type of context and a null class for clauses that do not play the role of seeds, as it is possible that a single clause functions as seed for two different types of contexts simultaneously:

The Fulton County Grand Jury said Friday an investigation of Atlanta's recent primary election produced "no evidence" that any irregularities took place.

In this example, the clause in italics simultaneously introduces two contexts, namely a temporal context (the context of the last Friday before the text was written) and the context of the statements of the Fulton Jury.

Table 1 contains just a few examples of words that can signal a seed. Note that the property of being a seed applies to the whole clause, not to a single word. Thus, if a clause contains one of these words, it is not necessarily a seed. In general, we should use either a rule-based system or machine learning algorithm to decide if a given clause is a seed or not. These words are in general polysemous; obviously, not all their senses signal the presence of a seed. Thus, a word sense disambiguation system is needed for a better identification of seeds.

4.3 From Seeds to Clusters

The detection of all types of contexts at once is unlikely to produce good results, since the detection of each type of context boils down to solving of a more specific problem, and the problems for each type of context are somehow different in nature. For each type of context, there are different procedures for finding seeds and it is natural to have different procedures for boundary detection.

Table 1. Examples of words that may indicate a seed

	verb-seeds	noun-seeds	adverb-seeds	adjective-seeds	other
objective	NA	NA	commonly	abstract	in general
subjective	say, tell	statement, declaration	reportedly	aforesaid	according to
probability	can, might	chance, possibility	perhaps	likely	if
temporal	NA	day, year	now, yesterday	NA	during, before
domain	NA	NA	scientifically	NA	at, in
necessity	have to, need to	requirement	necessarily	compulsory	NA
planning	want, plan	plan	hopefully	desired	after

Markers. Since it is a set of adjacent clauses, a cluster is completely determined by its boundaries; all we need to do is to mark the first and the last clause of the cluster. For this purpose, we use a B (beginning) marker for the first clause and an E (end) marker for the last one.

Initially, we set a B and an E marker on each seed-clause. Thus, after the initialization, each cluster has only one clause, namely the seed. After that, the markers are moved, enlarging the cluster in both directions. This growth process is somehow similar to the growth of a seed, and this is the reason why the term of seed has been adopted for naming the clause that indicates the presence of a context.

The decision to move a marker (expand the boundary) is taken using a machine learning algorithm. The markers can move when there is cohesion between the cluster and the next clause. The general principle underlying the method of deciding the movement of the markers is that the text must be coherent; however, since coherence is very hard to detect, we must rely on cohesion, which is closely associated to coherence. The problem of text cohesiveness is also unsolved, but we can introduce features that relate to cohesiveness so that the learner may exploit them (more about that in 5.2).

A marker is considered stuck when it meets a marker that comes from the opposite direction (due to the assumption that contexts do not overlap). It is also stuck when a decision "not move" is taken, since the features are the same, and thus any new evaluation would yield the same result.

For example, let us see the steps of the algorithm applied to the example from 4.1:

Table 2 shows the evolution of the cluster boundaries (assuming that the decision of moving the markers is perfect). At the end of the algorithm, the clause that has the B marker is the beginning clause, and the one that has the E marker is the last one in the cluster.

Marker Movement. The decision to move a marker can easily be cast as a binary classification problem. There are two possible decisions, namely to move

Table 2. Marker movement

No.	clauses	markers' positions					
		initial	step 1	step 2	step 3	step 4	step 5
1	"We're getting more 'pro' letters than 'con' on horse race betting",		B	B	B	B	B
2	said Ratcliff	B,E					
3	"But I believe		E				
4	if people were better informed on this question,			E			
5	most of them would oppose it also.				E		
6	I'm willing					E	
7	to stake my political career on it".						E

or to stop. This decision can be taken using a classifier automatically constructed by a machine learning algorithm.

In a nutshell, the second stage of context detection is as follows:

1. Initialize markers placing one B (from beginning) marker and one E (ending) marked at each seed
2. While we ca still move any marker, do the following:
   - for each beginning marker B, if decision(cluster(B), previous clause) = yes, move B upwards
   - for each ending marker E, if decision(cluster(E), next clause) move E downwards
3. Output the clusters obtained.

where decision(cluster(marker), clause) is the procedure that tells if the marker must be moved, machine learning or manually constructed. (Cluster(marker) is the cluster that is bounded by the marker).

4.4 Cluster Grouping

The final stage in context detection is the grouping of the clusters obtained at stage 2. This process will also depend on the type of context. It is a very simple task for subjective contexts, because all we need to do is group all statements, beliefs, etc. by the speaker to whom they belong.

For a dimensional context, the problem is also simple if the context is associated to a fixed, known temporal interval, like a day or an year. However, when we deal with spans of several days, for example, things get more complicated.

In general, the contexts considered in this paper are rich in information, like the situations in situation calculus. The need for efficiency in communication has made the natural language very compressed; definitely, a text is not built on the principle "what you see is what you get". What we find in the text is merely a projection of the real context, which resides in the speaker's mind. As a consequence, it may become difficult to determine to which context a certain cluster belongs. Note that failure to add some cluster to a context is preferable to

addition of a cluster to the wrong context. If a cluster that belongs to a context is wrongfully declared a new context, the impact on interpretation of sentences is low, provided that the information defining the context is correctly extracted. Incorrect introduction of a cluster in a context is, however, leading to falsity, because the information of the context will by assumed to be true for the cluster as well.

5 Experimental Implementation: Detection of Subjective Contexts

Since different procedures need to be developed for each class of contexts, a context detection system will be very large. The lack of a corpus compelled us to test the strategy of detection on a single class, namely the subjective contexts. The detection of subjective contexts follows the general algorithm described above. To complete the description, we only need to specify two aspects:

1. The procedure for the detection of the seeds
2. The way the decision for marker movement is taken

The most difficult of the two is the decision for marker movement. The rest of this section describes the details of this second problem and the experimental results.

5.1 Subjective Seed Detection

The class of subjective contexts has three subclasses, and for each of them there are different types of seeds. The presence of the statement contexts is indicated by the semantic relation "topic", which appears with communication verbs or their place-holders:

- communication verbs: say, state, tell, reply, answer, inform, declare, announce, etc.
- non-communication verbs used with a sense that implies communication: add, estimate, notify, etc. Example:
  John said he had seen Paul at school. In the detention room, *he added.*
- expressions that replace the communication verbs: *to have an answer for, one's explanation is*, etc.

The situation is similar for the belief contexts, the only difference coming from the fact that we have another set of verbs:

- verbs that express beliefs: believe, think, consider, etc.
- expressions that introduce the idea of belief: from one's point of view, for him, etc.

At last, fictive contexts are generally marked by the presence of nouns/verbs that express fictiveness:

- dream, novel, play, story, tale
- to imagine, to picture, to fancy

Example: *H.E.Bates has scribbled a farce called "Hark, Hark, the Lark"! It is one of the most entertaining and irresponsible novels of the season. If there is a moral lurking among the shenanigans, it is hard to find. Perhaps the lesson we should take from these pages is that the welfare state in England still allows wild scope for all kinds of rugged eccentrics . Anyway, a number of them meet here in devastating collisions. One is an imperial London stockbroker called Jerebohm. Another is a wily countryman called Larkin, [...]*

The author of this article is introducing us to the world of a literary work. The seed clause in this case is not the noun "farce" itself, but a reference to it (the adverb "here").

Practically, we need to detect the idea of communication (belief, fiction) in text; this can be achieved using a semantic relation detector that searches for the corresponding semantic relations, or even with the help of an word-sense disambiguation system, by selecting the right senses of the verbs. This is because, as seen in 4.2, not all senses of these words are signaling seeds; for example, from the 8 senses of verb "tell" in WordNet, only the first 7 imply communication.

For subjective contexts, the communication verbs are by far the most frequent manner of introducing a context (manual evaluation showed that more than 90% of the seed clauses contain such verbs). Moreover, for these verbs most of the senses (and also the most frequently used) imply communication.

For our experiments, we used gold standard seeds (we did this part of the task manually).

5.2 Machine Learning for Marker Movement

The decision for marker movement is taken using a classifier automatically constructed using Support Vector Machines. The features that we considered for the encoding of the training examples are listed below:

1. first connector
2. second connector - the first two features refer to the punctuation signs or conjuncts that make the connection between the current clause and the next one; if there is no such connector, we use the "null" value
3. direction of movement (up, down)
4. distance (number of clauses) from the seed
5. seed verb (the WordNet synset of the verb of the seed clause)
6. connector path from the seed - the sequence of connectors from the seed to the current clause; in this sequence, we eliminate the duplicate null values
7. verb tense of previous clause
8. verb tense of current clause
9. end of sentence encountered (the ".", "?" etc)
10. number of coreferences - this feature is computed by counting the coreference chains between words in the current context cluster (as much as it has grown at this point) and the next two clauses; the number three was arbitrary chosen; the reason we don't take it to be 1 is that strong cohesion between

the clauses of the context and the ones following the current clause is an indication that we should include the current clause in the context

11. marker clash (true if another marker of the same type, coming from opposite direction is encountered)

Example:

1. The Fulton County Grand Jury said Friday
2. an investigation of Atlanta 's recent primary election produced “no evidence”
3. that any irregularities took place .
4. The jury further said in term end presentments [...]

Clause number 1 is the seed, and the three following clauses have the feature vectors given below:

Table 3. Example of feature codification

1	2	3	4	5	6	7	8	9	10	11	target
null	null	down	1	say	null	past	past	f	0	f	y
null	that	down	2	say	null+that	past	past	f	0	f	y
par	further	down	3	say	null+that+par+further	past	past	t	1	t	n

The value “par” means that a paragraph end has been encountered. Note that the connectors are not necessarily placed at the beginning or ending of the clauses; in clause 4 of the example, the word “further” acts as connector, as it establishes the relation between the text segments. In general, all discourse markers (cue phrases - see [3]) must be considered.

Features 1, 2, 3, 4, and 6 are designed to characterize the structure of the text. The following example shows the rationale behind these features:

Mary said the dog ate the cake and *I don't believe it.* In this situation we have only one connector. The phrase has a low degree of ambiguity, but the addition of a comma before the conjunction “and” would make it even better. The interpretation is that the second phrase does not belong to the context of Mary's statements. However, this changes if we add another connector:

Mary said the dog ate the cake and that *I don't believe it.* Now, the second clause is clearly a part of the context of Mary's beliefs. The example considered here is as simple as can be, since it spans over a single short, clearly structured sentence. In spite of this simplicity, the detection of its boundaries is not straight forward, since we have to be able to correctly interpret the relation between the two clauses. Semantic information is also needed, for example:

Mary said the dog ate the cake and John was laughing. By replacing the second clause we have obtained an ambiguous phrase; it is not clear whether Mary said that John was laughing or if John was laughing while Mary was speaking. Syntactic information alone is not enough to solve such a problem.

We introduced features 7 and 8 to take advantage of the fact that there is a correlation between the change of the verbal tense and the structure of discourse. Feature number 9, though different in nature, has the same rationale as 7 and 8.

Finally, the number of coreferences (feature 10) was introduced as an additional measure of the cohesion of the text.

Our model is based on syntactic features only; we expect that the addition of semantic information will improve the results.

5.3 Results

We used a set of news articles collected from SemCor corpus for our experiments. We generated feature vectors (like in table 3) for the subjective contexts in these articles. We have obtained roughly 1000 training instances that we used for the evaluation of the model. The machine learning we used was Support Vector Machines ([4]), and the method of evaluation was 20 fold cross validation. Since the features are discrete, we encoded them in numerical vectors. The encoding procedure is widely used in N.L.P. applications: assign each feature a number of dimensions equal to the number of values; this way each value for each feature has a unique entry in the vector, which takes value 1 if the corresponding feature valued is encountered, and 0 otherwise. The kernel that we used was the RBF kernel. The accuracy of the decision to move or stop markers was 90.4%.

Table 4. Experimental results

	1	2	3	4	5	6	7	8	9	10
1	90.30									
2	88.69	88.83								
3	88.54	85.16	87.95							
4	90.01	88.69	88.25	89.86						
5	90.60	88.10	87.51	89.72	90.45					
6	89.42	84.28	88.10	90.01	89.72	90.01				
7	89.72	88.10	88.39	90.30	90.60	90.45	90.30			
8	90.89	87.81	87.95	89.86	89.28	90.01	90.60	90.30		
9	88.69	81.93	87.66	88.54	88.69	88.83	89.13	89.28	89.28	
10	90.45	88.39	88.69	90.16	89.72	89.72	90.01	90.01	89.57	89.72

In table 4, the field (i,j) contains the accuracy obtained if we remove the features i and j; (i, i) is the accuracy obtained by removing only feature number i. In the evaluation of the performance of the learning, we dropped feature 11 (marker clash). We did so because in the annotated data a marker clash can only happen at the boundaries of the clusters, while in real life this is obviously not true. By using this feature, the accuracy grows to nearly 95%. This proves that the learner deduces the rule that markers coming from opposite directions must stop when they meet.

We notice that the contribution of the features is rather uniform (none of the features has a major impact on its own); the coreference seems to have the lowest contribution.

6 Discussion

Table 4 shows that the feature no. 6 (connector path) does not have the impact that we expected. Intuitively, this feature carries a considerable informational baggage. In many instances, the human annotator can correctly guess the decision (if the marker must move) solely based on this feature. The conclusion is that the learner, as we use it now, can not fully exploit this feature.

The cost of an error in the decision of moving a marker is not constant; it depends on the distance between the clause where the marker stops and the one where it should stop. The further the marker stops, the worse the error is. Currently, our system does not take this fact into account.

References

1. John McCarthy. Notes on formalizing contexts. In Tom Kehler and Stan Rosenschein, editors, *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 555–560, Los Altos, California, 1986. Morgan Kaufmann.
2. Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project.* Addison-Wesley, Reading, Massachusetts, 1990.
3. Diane J. Litman. Classifying cue phrases in text and speech using machine learning. In *National Conference on Artificial Intelligence*, pages 806–813, 1994.
4. Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
5. Diane Litman and Julia Hirschberg. Disambiguating cue phrases in text and speech. In *13th. International Conference on Computational Linguistics (COLING-90)*, Helsinki, Finland, 1990.
6. Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Comput. Linguist.*, 12(3):175–204, 1986.
7. William C. Mann. Discourse structures for text generation. In *Proceedings of the 22nd conference on Association for Computational Linguistics*, pages 367–375. Association for Computational Linguistics, 1984.
8. Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.