SUCRE: A Modular System for Coreference Resolution

Hamidreza Kobdani and Hinrich Schütze
Institute for Natural Language Processing
University of Stuttgart, Germany
kobdani@ims.uni-stuttgart.de

Abstract

This paper presents SUCRE, a new software tool for coreference resolution and its feature engineering. It is able to separately do noun, pronoun and full coreference resolution. SUCRE introduces a new approach to the feature engineering of coreference resolution based on a relational database model and a regular feature definition language. SUCRE successfully participated in SemEval-2010 Task 1 on Coreference Resolution in Multiple Languages (Recasens et al., 2010) for gold and regular closed annotation tracks of six languages. It obtained the best results in several categories, including the regular closed annotation tracks of English and German.

1 Introduction

In this paper, we introduce a new software tool for coreference resolution. Coreference resolution is the process of finding discourse entities (markables) referring to the same real-world entity or concept. In other words, this process groups the markables of a document into equivalence classes (coreference entities) so that all markables in an entity are coreferent.

There are various publicly available systems that perform coreference resolution, such as BART (Versley et al., 2008) and GUITAR (Steinberger et al., 2007). A considerable engineering effort is needed for the full coreference resolution task, and a significant part of this effort concerns feature engineering. Thus, a system which is able to extract the features based on a feature definition language can help the researcher reduce the implementation effort needed for feature extraction. Most methods of coreference resolution, if providing a baseline, usually use a feature set similar to (Soon et al., 2001) or (Ng and Cardie, 2002) and do the feature extraction in the preprocessing stage. SUCRE has been developed to provide a more flexible method for feature engineering of coreference resolution. It has a novel approach to model an unstructured text corpus in a structured framework by using a relational database model and a regular feature definition language to define and extract the features. Relational databases are a well-known technology for structured data modeling and are supported by a wide array of software and tools. Converting a text corpus to/from its equivalent relational database model is straightforward in our framework.

A regular language for feature definition is a very flexible method to extract different features from text. In addition to features defined directly in SUCRE, it accepts also externally extracted/generated features. Its modular architecture makes it possible to use any externally available classification method too. In addition to link features (features related to a markable pair), it is also possible to define other kinds of features: atomic word and markable features. This approach to feature engineering is suitable not only for knowledge-rich but also for knowledge-poor datasets. It is also language independent. The results of SUCRE in SemEval-2010 Task 1 show the promise of our framework.

2 Architecture

The architecture of SUCRE has two main parts: preprocessing and coreference resolution.

In preprocessing the text corpus is converted to a relational database model. These are the main functionalities in this stage:

1. Preliminary text conversion
2. Extracting atomic word features
3. Markable detection

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 92–95,
Uppsala, Sweden, 15-16 July 2010.

Column	Characteristic
Word Table	
Word-ID	Primary Key
Document-ID	Foreign Key
Paragraph-ID	Foreign Key
Sentence-ID	Foreign Key
Word-String	Attribute
Word-Feature-0	Attribute
Word-Feature-1	Attribute
...	Attribute
Word-Feature-N	Attribute
Markable Table	
Markable-ID	Primary Key
Begin-Word-ID	Foreign Key
End-Word-ID	Foreign Key
Head-Word-ID	Foreign Key
Markable-Feature-0	Attribute
Markable-Feature-1	Attribute
...	Attribute
Markable-Feature-N	Attribute
Links Table	
Link-ID	Primary Key
First-Markable-ID	Foreign Key
Second-Markable-ID	Foreign Key
Coreference-Status	Attribute
Status-Confidence-Level	Attribute

Table 1: Relational Database Model of Text Corpus

4. Extracting atomic markable features

After converting (modeling) the text corpus to the database, coreference resolution can be performed. Its functional components are:

1. Relational Database Model of Text Corpus
2. Link Generator
3. Link Feature Extractor
4. Learning (Applicable on Train Data)
5. Decoding (Applicable on Test Data)

2.1 Relational Database Model of Text Corpus

The Relational Database model of text thev corpus is an easy to generate format. Three tables are needed to have a minimum running system: Word, Markable and Link.

Table 1 presents the database model of the text corpus. In the word table, Word-ID is the index of the word, starting from the beginning of the corpus. It is used as the primary key to uniquely identify each token. Document-ID, Paragraph-ID and Sentence-ID are each counted from the beginning of the corpus, and also act as the foreign keys pointing to the primary keys of the document, paragraph and sentence tables, which are optional (the system can also work without them). It is obvious that the raw text as well as any other format of the corpus can be generated from the word table. Any word features (Word-Feature-#X columns) can be defined and will then be added to the word table in preprocessing. In the markable table, Markable-ID is the primary key. Begin-Word-ID, End-Word-ID and Head-Word-ID refer to the word table. Like the word features, the markable features are not mandatory and in the preprocessing we can decide which features are added to the table. In the link table, Link-ID is the primary key; First-Markable-ID and Second-Markable-ID refer to the markable table.

2.2 Link Generator

For training, the system generates a positive training instance for each adjacent coreferent markable pair and negative training instances for a markable m and all markables disreferent with m that occur before m (Soon et al., 2001). For decoding it generates all the possible links inside a window of 100 markables.

2.3 Link Feature Extractor

There are two main categories of features in SUCRE: Atomic Features and Link Features

We first explain atomic features in detail and then turn to link features and the extraction method we use.

Atomic Features: The current version of SUCRE supports the atomic features of words and markables but in the next versions we are going to extend it to sentences, paragraphs and documents. An atomic feature is an attribute. For example the position of the word in the corpus is an atomic word feature. Atomic word features are stored in the columns of the word table called Word-Feature-X.

In addition to word position in the corpus, document number, paragraph number and sentence number, the following are examples of atomic word features which can be extracted in preprocessing: Part of speech tag, Grammatical Gender (male, female or neutral), Natural Gender (male or female), Number (e.g. singular, plural or both), Semantic Class, Type (e.g. pronoun types: personal, reflexive, demonstrative ...), Case (e.g. nominative, accusative, dative or genitive in German) and Pronoun Person (first, second or third). Other possible atomic markable features include:

number of words in markable, named entity, alias, syntactic role and semantic class.

For sentences, the following could be extracted: number of words in the sentence and sentence type (e.g. simple, compound or complex). For paragraphs these features are possible: number of words and number of sentences in the paragraph. Finally, examples of document features include document type (e.g. news, article or book), number of words, sentences and paragraphs in the document.

Link Features: Link features are defined over a pair of markables. For link feature extraction, the head words of the markables are usually used, but in some cases the head word may not be a suitable choice. For example, consider the two markables *the books* and *a book*. In both cases *book* is the head word, but to distinguish which markable is definite and which indefinite, the article must be taken into account. Now consider the two markables *the university student from Germany* and *the university student from France*. In this case, the head words and the first four words of each markable are the same but they can not be coreferent; this can be detected only by looking at the last words. Sometimes we need to consider all words in the two markables, or even define a feature for a markable as a unit. To cover all such cases we need a regular feature definition language with some keywords to select different word combinations of two markables. For this purpose, we define the following variables. **m1** is the first markable in the pair. **m1b**, **m1e** and **m1h** are the first, last and head words of the first markable in the pair. **m1a** refers to all words of the first markable in the pair. **m2, m2b, m2e, m2h** and **m2a** have the same definitions as above but for the second markable in the pair.

In addition to the above keywords there are some other keywords that this paper does not have enough space to mention (e.g. for accessing the constant values, syntax relations or roles). The currently available functions are: exact- and substring matching (in two forms: case-sensitive and case-insensitive), edit distance, alias, word relation, markable parse tree path, absolute value.

Two examples of link features are as follows:

- $(seqmatch(m1a, m2a) > 0)$
  $\&\&\ (m1h.f0 == f0.N\ )$
  $\&\&\ (m2h.f0 == f0.N\ )$
  means that there is at least one exact match between the words of the markables and that the head words of both are nouns (f0 means Word-Feature-0, which is part of speech in our system).

- $(abs(m2b.stcnum - m1b.stcnum) == 0)$
  $\&\&\ (m2h.f3 == f3.reflexive)$
  means that two markables are in the same sentence and that the type of the second markable head word is reflexive (f3 means Word-Feature-3, which is morphological type in our system).

2.4 Learning

There are four classifiers integrated in SUCRE: Decision-Tree, Naive-Bayes, Support Vector Machine (Joachims, 2002) and Maximum-Entropy (Tsuruoka, 2006).

When we compared these classifiers, the best results, which are reported in Section 3, were achieved with the Decision-Tree.

2.5 Decoding

In decoding, the coreference chains are created. SUCRE uses best-first clustering for this purpose. It searches for the best predicted antecedent from right-to-left starting from the end of the document.

3 Results

Table 2 shows the results of SUCRE and the best competitor system on the test portions of the six languages from SemEval-2010 Task 1. Four different evaluation metrics were used to rank the participating systems: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF (Luo, 2005) and BLANC (Recasens and Hovy, in prep).

SUCRE has the best results in regular closed annotation track of English and German (for all metrics). Its results for gold closed annotation track of both English and German are the best in MUC and BLANC scoring metrics (MUC: English +27.1 German +32.5, BLANC: English +9.5 German +9.0) and for CEAF and B^3 (CEAF: English -1.3 German -4.8, B^3: English -2.1 German -4.8); in comparison to the second ranked system, the performance is clearly better in the first case and slightly better in the second. This result shows that SUCRE has been optimized in a way that achieves good results on the four different scoring metrics. We view this good performance as a demonstration of the strength of SUCRE: our

method of feature extraction, definition and tuning is uniform and can be optimized and applied to all languages and tracks.

Results of SUCRE show a correlation between the MUC and BLANC scores (the best MUC scores of all tracks and the best BLANC scores in 11 tracks of a total 12), in our opinion this correlation is not because of the high similarity between MUC and BLANC, but it is because of the balanced scores.

Language	ca	de	en	es	it	nl
System	SUCRE (Gold Annotation)					
MD-F1	100	100	100	100	98.4	100
CEAF-F1	68.7	72.9	74.3	69.8	66.0	58.8
MUC-F1	**56.2**	**58.4**	**60.8**	**55.3**	**45.0**	**69.8**
B^3-F1	77.0	81.1	82.4	77.4	**76.8**	**67.0**
BLANC	**63.6**	**66.4**	**70.8**	**64.5**	**56.9**	**65.3**
System	SUCRE (Regular Annotation)					
MD-F1	69.7	**78.4**	**80.7**	70.3	**90.8**	**42.3**
CEAF-F1	47.2	**59.9**	**62.7**	52.9	**61.3**	15.9
MUC-F1	**37.3**	**40.9**	**52.5**	**36.3**	**50.4**	**29.7**
B^3-F1	51.1	**64.3**	**67.1**	55.6	**70.6**	11.7
BLANC	**54.2**	**53.6**	**61.2**	**51.4**	57.7	**46.9**
System	Best Competitor (Gold Annotation)					
MD-F1	100	100	100	100	N/A	N/A
CEAF-F1	**70.5**	**77.7**	**75.6**	66.6	N/A	N/A
MUC-F1	42.5	25.9	33.7	24.7	N/A	N/A
B^3-F1	**79.9**	**85.9**	**84.5**	**78.2**	N/A	N/A
BLANC	59.7	57.4	61.3	55.6	N/A	N/A
System	Best Competitor (Regular Annotation)					
MD-F1	**82.7**	59.2	73.9	**83.1**	55.9	34.7
CEAF-F1	**57.1**	49.5	57.3	**59.3**	45.8	**17.0**
MUC-F1	22.9	15.4	24.6	21.7	42.7	8.3
B^3-F1	**64.6**	50.7	61.3	**66.0**	46.4	**17.0**
BLANC	51.0	44.7	49.3	51.4	**59.6**	32.3

Table 2: Results of SUCRE and the best competitor system. Bold F1 scores indicate that the result is the best SemEval result. MD: Markable Detection, ca: Catalan, de: German, en:English, es: Spanish, it: Italian, nl: Dutch

4 Conclusion

In this paper, we have presented a new modular system for coreference resolution. In comparison with the existing systems the most important advantage of our system is its flexible method of feature engineering based on relational database and a regular feature definition language. There are four classifiers integrated in SUCRE: Decision-Tree, Naive-Bayes, SVM and Maximum-Entropy. The system is able to separately do noun, pronoun and full coreference resolution. The system uses best-first clustering. It searches for the best predicted antecedent from right-to-left starting from the end of the document.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Thorsten Joachims. 2002. *Learning to Classify Text Using Support Vector Machines, Methods, Theory, and Algorithms*. Kluwer/Springer.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the ACL*, pages 104–111.

Marta Recasens and Eduard Hovy. in prep. BLANC: Implementing the Rand Index for Coreference Evaluation.

Marta Recasens, Lluís Màrquez, Emili Sapena, M.Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. SemEval-2010 Task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluations (SemEval-2010)*, Uppsala, Sweden.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. In *Computational Linguistics*, pages 521–544.

Josef Steinberger, Massimo Poesio, Mijail A. Kabadjovb, and Karel Jezek. 2007. Two uses of anaphora resolution in summarization. In *Information Processing and Management, Special issue on Summarization*, pages 1663–1680.

Yoshimasa Tsuruoka. 2006. A simple c++ library for maximum entropy classification. *Tsujii laboratory, Department of Computer Science, University of Tokyo*.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, and Xiaofeng Yang. 2008. Bart: A modular toolkit for coreference resolution. In *Proceedings of the 46nd Annual Meeting of the Association for Computational Linguistics*, pages 9–12.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 45–52, Morristown, NJ, USA. Association for Computational Linguistics.