BART: A Modular Toolkit for Coreference Resolution

Yannick Versley[*], **Simone Ponzetto**[†], **Massimo Poesio**[‡], **Vladimir Eidelman**[£], **Alan Jern**[§], **Jason Smith**[þ], **Xiaofeng Yang**[△], **Alessandro Moschitti**[◇]

[*] University of Tübingen, versley@sfs.uni-tuebingen.de [†] EML Research gGmbH, ponzetto@eml-research.de
[‡] University of Essex, poesio@essex.ac.uk [£] Columbia University, vae2101@columbia.edu
[§] University of California Los Angeles, ajern@ucla.edu [þ] Johns Hopkins University, jsmith@jhu.edu
[△] Inst. for Infocomm Research, xiaofengy@i2r.a-star.edu.sg [◇] University of Trento, moschitti@dit.unitn.it

Abstract

Developing a full coreference system able to run all the way from raw text to semantic interpretation is a considerable engineering effort, yet there is very limited availability of off-the shelf tools for researchers whose interests are not in coreference, or for researchers who want to concentrate on a specific aspect of the problem. We present BART, a highly modular toolkit for developing coreference applications. In the Johns Hopkins workshop on using lexical and encyclopedic knowledge for entity disambiguation, the toolkit was used to extend a reimplementation of the Soon et al. (2001) proposal with a variety of additional syntactic and knowledge-based features, and experiment with alternative resolution processes, preprocessing tools, and classifiers.

1. Introduction

Coreference resolution refers to the task of identifying noun phrases that refer to the same extralinguistic entity in a text. Using coreference information has been shown to be beneficial in a number of other tasks, including information extraction (McCarthy and Lehnert, 1995), question answering (Morton, 2000) and summarization (Steinberger et al., 2007). Developing a full coreference system, however, is a considerable engineering effort, which is why a large body of research concerned with feature engineering or learning methods (e.g. Culotta et al. 2007; Denis and Baldridge 2007) uses a simpler but non-realistic setting, using preidentified mentions, and the use of coreference information in summarization or question answering techniques is not as widespread as it could be. We believe that the availability of a modular toolkit for coreference will significantly lower the entrance barrier for researchers interested in coreference resolution, as well as provide a component that can be easily integrated into other NLP applications.

A number of systems that perform coreference resolution are publicly available, such as GUITAR (Steinberger et al., 2007), which handles the full coreference task, and JAVARAP (Qiu et al., 2004), which only resolves pronouns. However, literature on coreference resolution, if providing a baseline, usually uses the algorithm and feature set of Soon et al. (2001) for this purpose.

2. System Architecture

The BART toolkit has been developed as a tool to explore the integration of knowledge-rich features into a coreference system at the Johns Hopkins Summer Workshop 2007. It is based on code and ideas from the system of Ponzetto and Strube (2006), but also includes some ideas from GUITAR (Steinberger et al., 2007) and other coreference systems (Versley, 2006; Yang et al., 2006) [1].

The goal of bringing together state-of-the-art approaches to different aspects of coreference resolution, including specialized preprocessing and syntax-based features has led to a design that is very modular. This design provides effective separation of concerns across several several tasks/roles, including engineering *new features* that exploit different sources of knowledge, designing improved or sprecialized *preprocessing* methods, and improving the way that coreference resolution is mapped to a *machine learning* problem.

Preprocessing The first part of preprocessing is realized on top of the MMAX2 discourse API (Müller and Strube, 2006), a library for standoff annotation that is also the foundation of the MMAX2 annotation tool. Using a generic format for standoff annotation makes it possible to combine the coreference resolution with other independent components, for example in a question answering system. It also becomes very easy to use integrated MMAX2 functionality (annotation diff, visual display) to perform qualitative error analysis.

Generally, the preprocessing pipeline involves components to annotate part-of-speech tags, chunks, and named entities. A final component, the merger, then combines chunking and NER information into markables on the *markable* annotation layer that correspond to the system's notion of a textual entity that can enter a coreference relation. The system is easily extensible by writing new components or mixing or matching existing ones. Our exploration of possible designs yielded the following pipelines:

- The *chunking* pipeline uses a classical tagger/chunker combination, with the Stanford POS tagger (Toutanova et al., 2003), the YamCha chunker (Kudoh and Matsumoto, 2000) and the Stanford Named Entity Recognizer (Finkel et al., 2005).
- The *parsing* pipeline uses Charniak and Johnson's reranking parser (Charniak and Johnson, 2005) to assign POS tags and uses base NPs as chunk equivalents, while also providing syntactic trees that can be used by feature extractors.
- The *Carafe* pipeline uses the parser in conjunction with an ACE mention tagger provided by MITRE (Wellner and Vilain, 2006). A specialized merger then

[1] An open source version of BART can be downloaded from `http://www.sfs.uni-tuebingen.de/~versley/BART`.

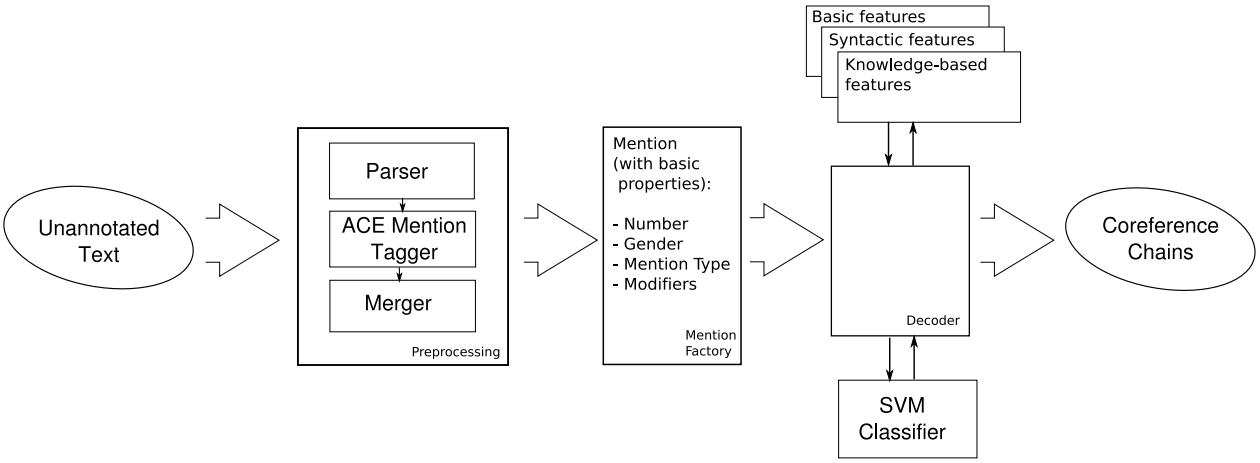

Figure 1: Example system configuration

discards any base NP that was not detected to be an ACE mention.

In a second step, the mention-building module uses the markables from this layer to create mention objects. These mention objects are grouped into equivalence classes by the resolution process and a coreference layer is written into the document, which can be used for detailed error analysis.

Feature Extraction BART's default resolver goes through all mentions and looks for possible antecedents in previous mentions as described by Soon et al. (2001). Each pair of anaphor and candidate is represented as a `PairInstance` object, which is enriched with classification features by feature extractors, and then handed over to a machine learning-based classifier that decides, given the features, whether anaphor and candidate are coreferent or not. Feature extractors are realized as separate classes, allowing for their independent development. The set of feature extractors that the system uses is set in an XML description file, which allows for straightforward prototyping and experimentation with different feature sets.

Learning Interfaces to several machine learning libraries have been realized:

- The *WEKA* machine learning toolkit (Witten and Frank, 2005); all classifiers from WEKA can be used.
- *SVMLight* (Joachims, 1999), or SVMLight/TK (Moschitti, 2006), a modified version of SVMLight that can be used with tree-valued features. Classification uses a Java Native Interface-based wrapper replacing SVMLight/TK's `svm_classify` program to improve the classification speed.
- A *Maximum entropy* classifier that is based on Robert Dodier's translation[2] of Liu and Nocedal's (1989) L-BFGS optimization code, with a function for programmatic feature combination.

Training/Testing The training and testing phases slightly differ from each other. In the training phase, the pairs that are to be used as training examples have to be selected in a process of sample selection, whereas in the testing phase, it has to be decided which pairs are to be given to the decision function and how to group mentions into equivalence relations given the classifier decisions.
This functionality is factored out into the *encoder/decoder* component, which is separate from feature extraction and machine learning itself. It is possible to completely change the basic behavior of the coreference system by providing new encoders/decoders, and still rely on the surrounding infrastructure for feature extraction and machine learning components.

3. Evaluation

Although BART is primarily meant as a platform for experimentation, it can be used simply as a coreference resolver, with a performance close to state of the art. Among the other publicly available systems for coreference resolution, GUITAR has only been evaluated on the Gnome corpus and a direct comparison is not necessary meaningful. For JAVARAP, Qiu et al. give figures for pronoun resolution on MUC6 that we can directly compare to; they give an accuracy of 61% for pronouns, whereas we get 64.3% recall and 63.1% precision on the same task for the basic feature set, whereas performance using the extended feature set with tree kernels gives 73.4% recall on MUC, coming near specialized pronoun resolution systems such as (Denis and Baldridge, 2007). As in Uryupina (2006), we can compare the performance using different learners. Using decision trees, we get results that are slightly below hers, whereas our MaxEnt results are slightly better. With a discretized sentence distance, we are able to efficiently use feature conjunctions; the corresponding results indicate that this is beneficial for system performance.

Lexical and Encyclopedic Knowledge As the goal of the workshop was using lexical and encyclopedic knowledge, we created an extended feature set including more information than the simple baseline. This includes syntactic features (e.g. using tree kernels to represent the syntactic relation between anaphor and antecedent, cf. Yang et al. 2006), as well as features based on knowledge extracted from Wikipedia (cf. Ponzetto and Smith, in preparation).

[2] `http://riso.sourceforge.net`

	Recall	Precision	F	train(sec.)	test(sec.)
J48	55.0	72.6	62.6	30	76
SVMlight (linear)	51.0	74.1	60.4	44	90
MaxEnt (plain)	52.4	73.4	61.2	31	75
SVMlight (polynomial d=2)	51.5	73.8	60.6	221	360
MaxEnt (combination d=2)	56.3	71.2	62.9	51	151
Soon et al (C5.0)	56.1	65.5	60.4		

Timing was measured on a 2GHz dual Opteron.

Table 1: Performance and time consumption (without preprocessing) for different classifiers on MUC7, Soon et al's feature set

	BNews			NPaper			NWire		
	Recl	Prec	F	Recl	Prec	F	Recl	Prec	F
basic feature set	0.594	0.522	0.556	0.663	0.526	0.586	0.608	0.474	0.533
extended feature set	0.607	0.654	0.630	0.641	0.677	0.658	0.604	0.652	0.627
Ng 2007*	0.561	0.763	0.647	0.544	0.797	0.646	0.535	0.775	0.633

*: "expanded feature set" in Ng 2007; Ng trains on the entire ACE training corpus.

Table 2: Performance on ACE-2 corpora, basic vs. extended feature set

Table 2 compares our results, obtained using this extended feature set, with results from Ng (2007).

4. Conclusions

We presented BART, a modular toolkit for coreference resolution that will be made available as open source, which provides an easy to use implementation of the Soon et al. algorithm. BART includes an extended feature set that uses syntactic and knowledge-based features to achieve state-of-the-art performance. We are currently investigating alternative resolution algorithms such as ranking-based resolution, either with a maximum entropy model as proposed by Luo et al. (2004), Versley (2006) or with the tournament-based ranking algorithm of Yang et al. (2005), as well as methods that incorporate more linguistic assumptions, such as those used in GUITAR. Other future work would include improvements to mention detection algorithms and a more comprehensive evaluation of features including those recently proposed by other researchers (e.g. Uryupina 2006; Ng 2007).

Achnowledgements We thank the CLSP at Johns Hopkins, NSF and the Department of Defense for ensuring funding for the workshop and to EML Research, MITRE, the Center for Excellence in HLT, and FBK-IRST, that provided partial support. Yannick Versley was supported by the Deutsche Forschungsgesellschaft as part of Collaborative Research Centre 441 "Linguistic Data Structures"; Simone Ponzetto has been supported by a grant of the Klaus Tschira Foundation (09.003.2004).

References

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc. ACL 2005*.

Culotta, A., Wick, M., and McCallum, A. (2007). First-order probabilistic models for coreference resolution. In *Proc. HLT/NAACL 2007*.

Denis, P. and Baldridge, J. (2007). A ranking approach to pronoun resolution. In *Proc. IJCAI 2007*.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. ACL 2005*, pages 363–370.

Joachims, T. (1999). Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*.

Kudoh, T. and Matsumoto, Y. (2000). Use of Support Vector Machines for chunk identification. In *Proc. CoNLL 2000*.

Liu, D. C. and Nocedal, J. (1989). On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., and Roukos, S. (2004). A mention-synchronous coreference resolution algorithm based on the bell tree. In *ACL 2004*.

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

Morton, T. S. (2000). Coreference for NLP applications. In *Proc. ACL 2000*.

Moschitti, A. (2006). Making tree kernels practical for natural language learning. In *Proc. EACL 2006*.

Müller, C. and Strube, M. (2006). Multi-level annotation of linguistic data with MMAX2. In Braun, S., Kohn, K., and Mukherjee, J., editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*. Peter Lang, Frankfurt a.M., Germany.

Ng, V. (2007). Shallow semantics for coreference resolution. In *Proc. IJCAI 2007*.

Ponzetto, S. P. and Strube, M. (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. HLT/NAACL 2006*.

Qiu, L., Kan, M.-Y., and Chua, T.-S. (2004). A public reference implementation of the RAP anaphora resolution algorithm. In *Proc. LREC 2004*.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of

noun phrases. *Computational Linguistics*, 27(4):521–544.

Steinberger, J., Poesio, M., Kabadjov, M., and Jezek, K. (2007). Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43:1663–1680. Special issue on Summarization.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. NAACL 2003*, pages 252–259.

Uryupina, O. (2006). Coreference resolution with and without linguistic knowledge. In *Proc. LREC 2006*.

Versley, Y. (2006). A constraint-based approach to noun phrase coreference resolution in German newspaper text. In *Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS 2006)*.

Wellner, B. and Vilain, M. (2006). Leveraging machine readable dictionaries in discriminative sequence models. In *Proc. LREC 2006*.

Witten, I. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yang, X., Su, J., and Tan, C. L. (2005). A twin-candidate model of coreference resolution with non-anaphor identification capability. In *Second International Joint Conference on Natural Language Processing (IJCNLP 2005)*.

Yang, X., Su, J., and Tan, C. L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. In *Proc. CoLing/ACL-2006*.