BART: A Multilingual Anaphora Resolution System

Samuel Broscheit*, Massimo Poesio‡, Simone Paolo Ponzetto*, Kepa Joseba Rodriguez‡, Lorenza Romano†, Olga Uryupina‡, Yannick Versley◇, Roberto Zanoli†

*Seminar für Computerlinguistik, University of Heidelberg
‡CiMeC, University of Trento
†Fondazione Bruno Kessler
◇SFB 833, University of Tübingen

broscheit@cl.uni-heidelberg.de, massimo.poesio@unitn.it,
ponzetto@cl.uni-heidelberg.de, kepa.rodriguez@unitn.it,
romano@fbk.eu, uryupina@gmail.com,
versley@sfs.uni-tuebingen.de, zanoli@fbk.eu

Abstract

BART (Versley et al., 2008) is a highly modular toolkit for coreference resolution that supports state-of-the-art statistical approaches and enables efficient feature engineering. For the SemEval task 1 on Coreference Resolution, BART runs have been submitted for German, English, and Italian.

BART relies on a maximum entropy-based classifier for pairs of mentions. A novel entity-mention approach based on Semantic Trees is at the moment only supported for English.

1 Introduction

This paper presents a multilingual coreference resolution system based on BART (Versley et al., 2008). BART is a modular toolkit for coreference resolution that supports state-of-the-art statistical approaches to the task and enables efficient feature engineering. BART has originally been created and tested for English, but its flexible modular architecture ensures its portability to other languages and domains. In SemEval-2010 task 1 on Coreference Resolution, BART has shown reliable performance for English, German and Italian.

In our SemEval experiments, we mainly focus on extending BART to cover multiple languages. Given a corpus in a new language, one can re-train BART to obtain baseline results. Such a language-agnostic system, however, is only used as a starting point: substantial improvements can be achieved by incorporating language-specific information with the help of the *Language Plugin*. This design provides effective separation between linguistic and machine learning aspects of the problem.

2 BART Architecture

The BART toolkit has five main components: preprocessing pipeline, mention factory, feature extraction module, decoder and encoder. In addition, an independent *LanguagePlugin* module handles all the language specific information and is accessible from any component. The architecture is shown on Figure 1. Each module can be accessed independently and thus adjusted to leverage the system's performance on a particular language or domain.

The preprocessing pipeline converts an input document into a set of lingustic layers, represented as separate XML files. The mention factory uses these layers to extract mentions and assign their basic properties (number, gender etc). The feature extraction module describes pairs of mentions $\{M_i, M_j\},\ i < j$ as a set of features.

The decoder generates training examples through a process of sample selection and learns a pairwise classifier. Finally, the encoder generates testing examples through a (possibly distinct) process of sample selection, runs the classifier and partitions the mentions into coreference chains.

3 Language-specific issues

Below we briefly describe our language-specific extensions to BART. These issues are addressed in more details in our recent papers (Broscheit et al., 2010; Poesio et al., 2010).

3.1 Mention Detection

Robust mention detection is an essential component of any coreference resolution system. BART supports different pipelines for mention detection. The

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 104–107,
Uppsala, Sweden, 15-16 July 2010.

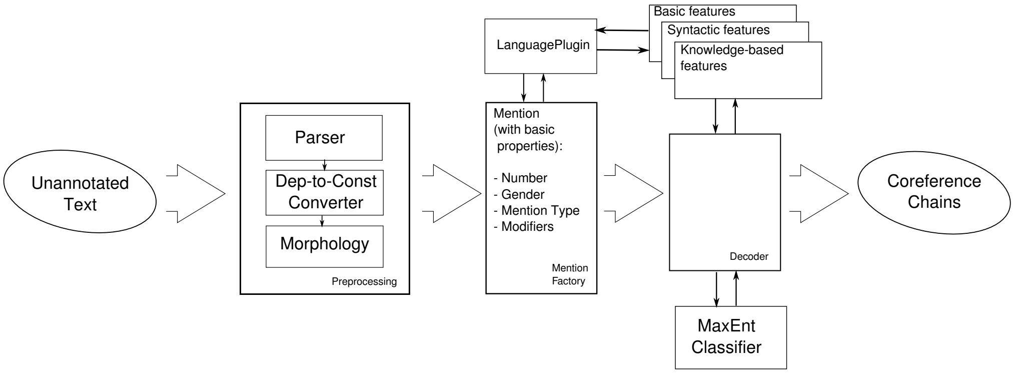

Figure 1: BART architecture

choice of a pipeline depends crucially on the availability of linguistic resources for a given language.

For English and German, we use the *Parsing Pipeline and Mention Factory* to extract mentions. The parse trees are used to identify minimal and maximal noun projections, as well as additional features such as number, gender, and semantic class.

For **English**, we use parses from a state-of-the-art constituent parser (Petrov et al., 2006) and extract all base noun phrases as mentions. For **German**, the SemEval dependency tree is transformed to a constituent representation and minimal and maximal phrases are extracted for all nominal elements (pronouns, common nouns, names), except when the noun phrase is in a non-referring syntactic position (for example, expletive “es”, predicates in copula constructions).

For **Italian**, we use the *EMD Pipeline and Mention Factory*. The Typhoon (Zanoli et al., 2009) and DEMention (Biggio et al., 2009) systems were used to recognize mentions in the test set. For each mention, its head and extension were considered. The extension was learned by using the mention annotation provided in the training set (13th column) whereas the head annotation was learned by exploiting the information produced by MaltParser (Nivre et al., 2007). In addition to the features extracted from the training set, such as prefixes and suffixes (1-4 characters) and orthographic information (capitalization and hyphenation), a number of features extracted by using external resources were used: mentions recognized by TextPro (http://textpro.fbk.eu), gazetteers of generic proper nouns extracted from the Italian phone-book and Wikipedia, and other features derived from WordNet. Each of these features was extracted in a local context of ± 2 words.

3.2 Features

We view coreference resolution as a binary classification problem. Each classification instance consists of two markables, i.e. an anaphor and potential antecedent. Instances are modeled as feature vectors (cf. Table 1) and are handed over to a binary classifier that decides, given the features, whether the anaphor and the candidate are coreferent or not. All the feature values are computed automatically, without any manual intervention.

Basic feature set. We use the same set of relatively language-independent features as a backbone of our system, extending it with a few language-specific features for each subtask. Most of them are used by virtually all the state-of-the-art coreference resolution systems. A detailed description can be found, for example, in (Soon et al., 2001).

English. Our English system is based on a novel model of coreference. The key concept of our model is a *Semantic Tree* – a filecard associated with each discourse entity containing the following fields:

- **Types:** the list of types for mentions of a given entity. For example, if an entity contains the mention “software from India”, the shallow predicate “software” is added to the types.
- **Attributes:** this field collects the premodifiers. For instance, if one of the mentions is “the expensive software” the shallow attribute “expensive” is added to the list of attributes.
- **Relations:** this field collects the prepositional postmodifiers. If an entity contains the mention “software from India”, the shallow relation “from(India)” is added to the list of relations.

For each mention BART creates such a filecard using syntactic information. If the classifier decides that both mentions are corefering, the filecard of the anaphora is merged into the filecard of the antecedent (cf. Section 3.3 below).

The `SemanticTreeCompatibility` feature extractor checks whether individual slots of the anaphor's filecard are compatible with those of the antecedent's.

The `StrudelRelatedness` feature relies on Strudel – a distributional semantic model (Baroni et al., 2010). We compute Strudel vectors for the sets of types of the anaphor and the antecedent. The relatedness value is determined as the cosine between the two.

German. We have tested extra features for German in our previous study (Broscheit et al., 2010).

The `NodeDistance` feature measures the number of clause nodes (SIMPX, R-SIMPX) and prepositional phrase nodes (PX) along the path between M_j and M_i in the parse tree.

The `PartialMorphMatch` feature is a substring match with a morphological extension for common nouns. In German the frequent use of noun composition makes a simple string match for common nouns unfeasible. The feature checks for a match between the noun stems of M_i and M_j. We extract the morphology with SMOR/Morphisto (Schmid et al., 2004).

The `GermanetRelatedness` feature uses the Pathfinder library for GermaNet (Finthammer and Cramer, 2008) that computes and discretizes raw scores into three categories of semantic relatedness. In our experiments we use the measure from Wu and Palmer (1994), which has been found to be the best performing on our development data.

Italian. We have designed a feature to cover Italian aliasing patterns. A list of company/person designators (e.g., "S.p.a" or "D.ssa") has been manually crafted. We have collected patterns of name variants for locations. Finally, we have relaxed abbreviation constraints, allowing for lower-case characters in the abbreviations. Our pilot experiments suggest that, although a universal aliasing algorithm is able to resolve some coreference links between NEs, creating a language-specific module boosts the system's performance for Italian substantially.

Basic feature set
MentionType(M_i),MentionType(M_j)
SemanticClass(M_i), SemanticClass(M_j)
GenderAgreement(M_i,M_j)
NumberAgreement(M_i,M_j)
AnimacyAgreement(M_i,M_j)
StringMatch(M_i,M_j)
Distance(M_i,M_j)
Basic features used for English and Italian
Alias(M_i,M_j)
Apposition(M_i,M_j)
FirstMention(M_i)
English
IsSubject(M_i)
SemanticTreeCompatibility(M_i,M_j)
StrudelRelatedness(M_i,M_j)
German
InQuotedSpeech(M_i), InQuotedSpeech(M_j)
NodeDistance(M_i,M_j)
PartialMorphMatch(M_i,M_j)
GermanetRelatedness(M_i,M_j)
Italian
AliasItalian(M_i,M_j)

Table 1: Features used by BART: each feature describes a pair of mentions $\{M_i, M_j\},\ i < j$, where M_i is a candidate antecedent and M_j is a candidate anaphor

3.3 Resolution Algorithm

The BART toolkit supports several models of coreference (pairwise modeling, rankers, semantic trees), as well as different machine learning algorithms. Our final setting relies on a pairwise maximum entropy classifier for Italian and German.

Our English system is based on an entity-mention model of coreference. The key concept of our model is a Semantic Tree - a filecard associated to each discourse entity (cf. Section 3.2). Semantic trees are used for both computing feature values and guiding the resolution process.

We start by creating a Semantic Tree for each mention. We process the document from left to right, trying to find an antecedent for each mention (candidate anaphor). When the antecedent is found, we extend its Semantic Tree with the types, attributes and relations of the anaphor, provided they are mutually compatible. Consider, for example, a list of mentions, containing, among others, "software from India", "the software" and "software from China". Initially, BART creates the following semantic trees: "(type: software) (relation: from(India))", "(type: software)" and "(type: software) (relation: from(China))". When the second mention gets resolved to the first one, their semantic trees are merged to "(type: software) (relation: from(India)". Therefore, when we attempt to resolve the third mention, both candidate antecedents are rejected, as their relation attributes are incompatible with "from(China)". This approach helps us avoid erroneous links (such as the link between the second and the third mentions in our example) by leveraging entity-level information.

4 Evaluation

The system was evaluated on the SemEval task 1 corpus by using the SemEval scorer.

First, we have evaluated our mention detection modules: the system's ability to recognize both the mention extensions and the heads in the *regular* setting. BART has achieved the best score for mention detection in German and has shown reliable figures for English. For Italian, the moderate performance level is due to the different algorithms for identifying the heads: the MaltParser (trained on TUT: http://www.di.unito.it/t̃utreeb) produces a more semantic representation, while the SemEval scorer seems to adopt a more syntactic approach.

Second, we have evaluated the quality of our coreference resolution modules. For German, BART has shown better performance than all the other systems on the *regular* track.

For English, the only language targeted by all systems, BART shows good performance over all metrics in the *regular* setting, usually only outperformed by systems that were tuned to a particular metric.

Finally, the Italian version of BART shows reliable figures for coreference resolution, given the mention alignment problem discussed above.

5 Conclusion

We have presented BART – a multilingual toolkit for coreference resolution. Due to its highly modular architecture, BART allows for efficient language-specific feature engineering. Our effort represents the first steps towards building a freely available coreference resolution system for many languages.

References

Marco Baroni, Brian Murphy, Eduard Barbu, and Massimo Poesio. 2010. Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, 34(2):222–254.

Silvana Marianela Bernaola Biggio, Claudio Giuliano, Massimo Poesio, Yannick Versley, Olga Uryupina, and Roberto Zanoli. 2009. Local entity detection and recognition task. In *Proc. of Evalita-09*.

Samuel Broscheit, Simone Paolo Ponzetto, Yannick Versley, and Massimo Poesio. 2010. Extending BART to provide a coreference resolution system for German. In *Proc. of LREC '10*.

Marc Finthammer and Irene Cramer. 2008. Exploring and navigating: Tools for GermaNet. In *Proc. of LREC '08*.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Slav Petrov, Leon Barett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of COLING-ACL-06*.

Massimo Poesio, Olga Uryupina, and Yannick Versley. 2010. Creating a coreference resolution system for Italian. In *Proc. of LREC '10*.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proc. of LREC '04*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics (Special Issue on Computational Anaphora Resolution)*, 27(4):521–544.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. BART: A modular toolkit for coreference resolution. In *Proceedings of the Linguistic Coreference Workshop at the International Conference on Language Resources and Evaluation (LREC-2008)*.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proc. of ACL-94*, pages 133–138.

Roberto Zanoli, Emiliano Pianta, and Claudio Giuliano. 2009. Named entity recognition through redundancy driven classifier. In *Proc. of Evalita-09*.