The Taming of Reconcile as a Biomedical Coreference Resolver

Youngjun Kim, Ellen Riloff, Nathan Gilbert
School of Computing
University of Utah
Salt Lake City, UT
{youngjun, riloff, ngilbert} @cs.utah.edu

Abstract

To participate in the Protein Coreference section of the BioNLP 2011 Shared Task, we use Reconcile, a coreference resolution engine, by replacing some pre-processing components and adding a new mention detector. We got some improvement from training two separate classifiers for detecting anaphora and antecedent mentions. Our system yielded the highest score in the task, F-score 34.05% in partial mention, protein links, and system recall mode. We witnessed that specialized mention detection is crucial for coreference resolution in the biomedical domain.

1 Introduction

Coreference resolution is a mechanism that groups entity mentions in a text into coreference chains based on whether they refer to the same real-world entity or concept. Like other NLP applications, which must meet the need for aggressive and sophisticated methods of detecting valuable information in emerging domains, numerous coreference resolvers have been developed, including JavaRap (Qiu et al., 2004), GuiTaR (Poesio and Kabadjov, 2004) and BART (Versley et al., 2008). Our research uses a recently released system, Reconcile (Stoyanov et al, 2009; 2010a; 2010b), which was designed as a general architecture for coreference resolution that can be used to easily create learning-based coreference resolvers. Reconcile is based on supervised learning approaches to coreference resolution and has showed relatively good performance compared with similar types of systems.

As a first step to adapting Reconcile for the biomedical domain, specifically the BioNLP Shared Task 2011 (Kim et al., 2011), we modified several subcomponents in Reconcile and revised the feature set for this task. Most importantly, we created a specialized mention detector trained for biomedical text. We trained separate classifiers for detecting anaphor and antecedent mentions, and experimented with several clustering techniques to discover the most suitable algorithm for producing coreference chains in this domain.

2 BioNLP 2011 Shared Task

Our system was developed to participate in a Protein Coreference (COREF) task (Nguyen et al., 2011), one of the supporting tasks in the BioNLP Shared Task 2011. The COREF task is to find all mentions participating in the coreference relation and to connect the anaphora-antecedent pairs. The corpus is based on the Genia-Medco coreference corpus. The Genia-Medco corpus was produced for the biomedical domain, and some comparative analysis with this corpus and other newswire domain data have been performed (Yang et al., 2004a; 2004b; Nguyen and Kim, 2008; Nguyen et al., 2008).

The COREF corpus consists of 800 text files for training, 150 for development, and 260 for testing, which all have gene/protein coreference annotations. The training set has 2,313 pairs of coreference links with 4,367 mentions. 2,117 mentions are antecedents, with an average of 4.21 tokens each (delimited by white space), and 2,301

mentions are anaphora, with an average of 1.28 tokens each. The anaphora are much shorter because many of them are pronouns. The five most frequent anaphora are *that* (686 times), *which* (526), *its* (270), *their* (130), and *it* (100).

3 Our Coreference Resolver

Reconcile was designed to be a research testbed capable of implementing the most current approaches to coreference resolution. Reconcile is written in Java, to be portable across platforms, and was designed to be easily reconfigurable with respect to sub-components, feature sets, parameter settings, etc. A mention detector and an anaphora-antecedent pairs generator are added for the COREF task.

3.1 Preprocessing

For pre-processing, we used the Genia Tagger (Tsuruoka and Tsujii. 2005) for sentence splitting, tokenizing, and part-of-speech (POS) tagging. For parsing, we used the Enju parser (Miyao and Tsujii, 2008).

We replaced Reconcile's mention detection module with new classifiers because of poor performance on the biomedical domain with the provided classifiers. We reformatted the training data with IOB tags and trained a sequential classifier using CRF++ (Kudoh, 2007). For this sequence tagging, we borrowed the features generally used for named entity recognition in the biomedical literature (Finkel et al., 2005; Zhou et al., 2005; McDonald and Pereira, 2005), including word, POS, affix, orthographic features and combinations of these features. We extracted features from the target word, as well as two words to its left and two words to its right. Two versions of mention detectors were developed. The first (MD-I) trained one model without differentiating between anaphora and antecedents. For this method, we chose the longest mentions when multiple mentions overlapped. The other detector (MD-II) used two different models for the antecedent and anaphor, classifying them separately. MD-II's classification result was used when generating the anaphora-antecedent pairs. Table 1 shows the performance of exact matching by these detectors compared with the performance of the Genia Noun Phrase (NP) chunker. Our classifiers did much better, 81.31% precision and 64.78% recall (MD-II), than the Genia chunker, 6.58% precision and 72.67% recall. Only an average of six mentions occurred in each text, while the Genia chunker detected 66.27 noun phrases on average. The Genia annotation scheme was not limited to specific types of concepts, so the Genia NP chunker identifies every possible concept. In contrast, the COREF shared task only involves a subset of the concepts. Mention boundaries were also frequently mismatched. For example, "its" was annotated as a mention in the COREF task when it appears as a possessive inside a noun phrase (e.g., "its activity"), but the Genia NP Chunker tags the entire noun phrase as a mention.

	Prec	Rec	F
Genia NP Chunker	6.58	72.67	12.07
Mention Detector-I	80.85	63.33	71.03
Mention Detector-II	81.31	64.78	**72.11**
Antecedent	65.48	41.35	50.69
Anaphor	91.72	85.07	88.27

Table 1:Mention Detection Results on Dev. Set

3.2 Feature Generation

We used the following four types of features:

Lexical: String-based comparisons of the two mentions, such as exact string matching and head noun matching.

Proximity: Sentence measures of the distance between two mentions.

Grammatical: A wide variety of syntactic properties of the mentions, either individually or in pairs. These features are based on part-of-speech tags, or parse trees.

Semantic: Semantic information about one or both mentions, such as tests for gender and animacy.

Due to the unavailability of paragraph information in our training data, we excluded Reconcile's paragraph features. Also, named entity and dependency parsing features were not used for training. Table 2 shows the complete feature set used for this task. In total, we excluded nine existing Reconcile features, mostly semantic features: WordNetClass, WordNetDist, WordNetSense, Subclass, ParNum, SameParagraph, IAntes, Prednom, WordOverlap. Full descriptions of these features can be found in Stoyanov (2010a).

Lexical	HeadMatch, PNStr, PNSubstr, ProStr, SoonStr, WordsStr, WordsSubstr
Proximity	ConsecutiveSentences, SentNum, SameSentence
Syntactic	Binding, BothEmbedded, BothInQuotes, BothPronouns, BothProperNouns, BothSubjects, ContainsPN, Contraindices, Definite1, Definite2, Demonstrative2, Embedded1, Embedded2, Indefinite, Indefinite1, InQuote1, InQuote2, MaximalNP, Modifier, PairType, Pronoun, Pronoun1, Pronoun2, ProperNoun, ProResolve, RuleResolve, Span, Subject1, Subject2, Syntax
Semantic	Agreement, Alias, AlwaysCompatible, Animacy, Appositive, ClosestComp, Constraints, Gender, instClass, Number, ProComp, ProperName, Quantity, WNSynonyms

Table 2: Feature Set for Coreference Resolution

3.3 Clustering

After Reconcile makes pairwise decisions linking each anaphor and antecedent, it produces a clustering of the mentions in a document to create coreference chains. Because the format of the COREF task submission was not chains but anaphora-antecedent pairs, it would have been possible to submit the direct results of Reconcile's pairwise decisions. However, it was easier to use Reconcile as a black-box and post-process the chains to reverse-engineer coreferent pairs from them. Reconcile supports three clustering algorithms:

Single-link Clustering (SL) (Transitive Closure) groups together all mentions that are connected by a path of coreferent links.

Best-first (BF) clustering uses the classifier's confidence value to cluster each noun phrase with its most confident antecedent.

Most Recent First (MRF) pairs each noun phrase with the single most recent antecedent that is labeled as coreferent.

Table 3 shows the MUC scores of each clustering method with gold standard mentions and with the mentions automatically detected by each of our two mention detectors. Not surprisingly, using gold mentions produced the highest score of 87.32%. Automatically detected mentions yielded much lower performance. MD-I performed best, in this evaluation, achieving 49.65%. The *most recent first* clustering algorithm produced the best results for both gold mentions and MD-I. The *single link* clustering algorithm, which is the default method used by Reconcile, produced the lowest results for both gold mentions and MD-I.

	SL	BF	MRF
Gold Mention	85.34	86.87	**87.32**
Mention Detector-I	48.64	48.82	**49.65**
Mention Detector-II	48.31	**48.62**	48.07

Table 3: MUC Scores of Dev. Set by Three Different Clustering Methods (SL: *Single-link*, BF: *Best-first*, MRF: *Most recent first*)

3.4 Pair Generation from Chains

Reconcile generates coreference chains, but the output for the shared task required anaphora-antecedent pairs. Therefore, we needed to extract individual pairs from the chains. We used the chains produced by the *most recent first* clustering algorithm for pair generation. When using MD-I output, we took the earliest mention (i.e., the one occurring first in the source document) in the chain and paired it with each of the subsequent mentions in the same chain. Thus, each chain of size N produced N-1 pairs. When using the MD-II predictions, the classifiers gave us two separate lists of antecedent and anaphora mentions. In this case, we paired each anaphor in the chain with every antecedent in the same chain that preceded it in the source document.

3.5 Evaluation and Analysis

The mention linking can be evaluated using three different scores: *atom* coreference links, *protein* coreference links, and *surface* coreference links. In the *atom* link option, only links containing given gene/protein annotations are considered while in the *surface* link option, every link is a target for the evaluation. *Protein* links are similar to *atom* links but loosen the boundary of gene/protein annotations. There were 202 protein links out of 469 surface links in development set.

For mention detection, *exact* match and *partial* match are supported in the task evaluation. Recall is measured in two modes. In *system* mode, every link is calculated for the linking evaluation. In *algorithm* mode, only links with correctly detected mentions are considered for evaluation. For

detailed information refer to Nguyen et al. (2011) or the task web site.[1] Table 4 shows the mention linking results (F-score) for the COREF task evaluation using *partial* match and *system* recall. The *surface* link score on gold mentions reached 90.06%. For automatic mention detection, MD-I achieved a score of 45.38% score, but MD-II produced a substantially better score of 50.41%. MD-II, which was trained separately for antecedent and anaphora detection, performed about 5% higher than MD-I in every link mode.

	Atom	Protein	Surface
Gold Mention	84.09	**84.09**	90.06
Mention Detector-I	28.67	**34.41**	45.38
Mention Detector-II	33.45	**39.27**	50.41

Table 4: Dev. Set Results by Three Different Evaluation Options

Table 5 shows the recall and precision breakdown for the *protein* evaluation results. Looking behind the composite F-score reveals that our system produced higher precision than recall. Looking back at Table 1, we saw that our anaphor detector performed much better than our antecedent detector. Since every coreference link requires one of each, the relatively poor performance of antecedent detection (especially in terms of recall) is a substantial bottleneck.

	Prec	Rec	F
Gold Mention	98.67	73.27	**84.09**
Mention Detector-I	62.34	23.76	**34.41**
Mention Detector-II	73.97	26.73	**39.27**

Table 5: Precision and Recall Breakdown for *Protein* Evaluation Coreference Links

3.6 Results: Submission for COREF Task

We merged the training and development sets to use as training data for Reconcile. We used MD-II for mention detection and the *most recent first* algorithm for clustering to submit the final output on the test data. Table 6 shows the results of our final submission along with the five other participating teams for the *protein* evaluation coreference links (Nguyen et al., 2011). Our system produced a 34.05% F-score (73.26% precision and 22.18% recall) in *protein* coreference links and 25.41% F-score in *atom* links.

Team	Prec	Rec	F
University of Utah	73.26	22.18	**34.05**
University of Zurich	55.45	21.48	**30.96**
Concordia University	63.22	19.37	**29.65**
University of Turku	67.21	14.44	**23.77**
University of Szeged	3.47	3.17	**3.31**
University College Dublin	0.25	0.70	**0.37**

Table 6: Evaluation Results of Final Submissions (*Protein* Coreference Links)

4 Conclusions

The effort to tame Reconcile as a coreference engine for the biomedical domain was successful and our team's submission obtained satisfactory results. However, there is ample room for improvement in coreference resolution. We observed that mention detection is crucial - the MUC score reached 87.32% with gold mentions on the development set but only 49.65% with automatically detected mentions (Table 3). One possible avenue for future work is to develop domain-specific features to better identify mentions in biomedical domains.

Acknowledgments

We thank the BioNLP Shared Task 2011 organizers for their efforts, and gratefully acknowledge the support of the National Science Foundation under grants IIS-1018314 and DBI-0849977 and the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily express the view of the DARPA, AFRL, NSF, or the U.S. government.

References

Jenny Finkel, Shipra Dingare, Christopher D Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. 2005. Exploring the Boundaries: Gene and Protein Identification in Biomedical Text. BMC Bioinformatics. 6:S5.

[1] http://sites.google.com/site/bionlpst/home/protein-gene-coreference-task

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Jun'ichi Tsujii. 2011. Overview of BioNLP Shared Task 2011. Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task, Portland, Oregon, June. ACL 2011.

Taku Kudoh. 2007. CRF++. http://crfpp.sourceforge.net/.

Ryan McDonald and Fernando Pereira. 2005. Identifying Gene and Protein Mentions in Text Using Conditional Random Fields. BMC Bioinformatics. 6:S6.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature Forest Models for Probabilistic HPSG Parsing. Computational Linguistics, 34(1):35–80.

Ngan L. T. Nguyen and Jin-Dong Kim. 2008. Exploring Domain Differences for the Design of Pronoun Resolution Systems for Biomedical Text. Proceedings of COLING 2008:625-632

Ngan L. T. Nguyen, Jin-Dong Kim, and Jun'ichi Tsujii. 2008. Challenges in Pronoun Resolution System for Biomedical Text. Proceedings of LREC 2008.

Ngan L. T. Nguyen, Jin-Dong Kim, and Jun'ichi Tsujii. 2011. Overview of the Protein Coreference task in BioNLP Shared Task 2011. Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task, Portland, Oregon, June. ACL 2011.

Massimo Poesio and Mijail A. Kabadjov. 2004. A general-purpose, off-the-shelf anaphora resolution module: implementation and preliminary evaluation. Proceedings of LREC 2004.

Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2004. A Public Reference Implementation of the Rap Anaphora Resolution Algorithm. Proceedings of LREC 2004.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010a. Reconcile: A Coreference Resolution Platform. Tech Report. Cornell University.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010b. Coreference Resolution with Reconcile. Proceedings of ACL 2010.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in Noun Phrase Coreference Resolution: Making Sense of the State-of-the-Art. Proceedings of ACL-IJCNLP 2009.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. Proceedings of HLT/EMNLP 2005:467-474.

Yannick Versley, Simone P. Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. BART: A modular toolkit for coreference resolution. Proceedings of LREC 2008.

Xiaofeng Yang, Jian Su, Guodong Zhou, and Chew Lim Tan. 2004a. A NP-Cluster Based Approach to Coreference Resolution. Proceedings of COLING 2004:226-232.

XiaoFeng Yang, GuoDong Zhou, Jian Su, and Chew Lim Tan. 2004b. Improving Noun Phrase Coreference Resolution by Matching Strings. Proceedings of IJCNLP 2004:226-333.

GuoDong Zhou, Dan Shen, Jie Zhang, Jian Su, and SoonHeng Tan. 2005. Recognition of Protein/Gene Names from Text Using an Ensemble of Classifiers. BMC Bioinformatics. 6:S7.