Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)

Domain-Specific Coreference Resolution with Lexicalized Features

Nathan Gilbert and **Ellen Riloff**
School of Computing
University of Utah
50 S. Central Campus Dr.
Salt Lake City, UT 84112
USA
{ngilbert,riloff}@cs.utah.edu

Abstract

Most coreference resolvers rely heavily on string matching, syntactic properties, and semantic attributes of words, but they lack the ability to make decisions based on individual words. In this paper, we explore the benefits of lexicalized features in the setting of domain-specific coreference resolution. We show that adding lexicalized features to off-the-shelf coreference resolvers yields significant performance gains on four domain-specific data sets and with two types of coreference resolution architectures.

1 Introduction

Coreference resolvers are typically evaluated on collections of news articles that cover a wide range of topics, such as the ACE (ACE03, 2003; ACE04, 2004; ACE05, 2005) and OntoNotes (Pradhan et al., 2007) data sets. Many NLP applications, however, involve text analysis for specialized domains, such as clinical medicine (Gooch and Roudsari, 2012; Glinos, 2011), legal text analysis (Bouayad-Agha et al., 2009), and biological literature (Batista-Navarro and Ananiadou, 2011; Castaño et al., 2002). Learning-based coreference resolvers can be easily retrained for a specialized domain given annotated training texts for that domain. However, we found that retraining an off-the-shelf coreference resolver with domain-specific texts showed little benefit.

This surprising result led us to question the nature of the feature sets used by noun phrase (NP) coreference resolvers. Nearly all of the features employed by recent systems fall into three categories: string match and word overlap, syntactic properties (e.g., appositives, predicate nominals, parse features, etc.), and semantic matching (e.g., gender agreement, WordNet similarity, named entity classes, etc.). Conspicuously absent from most systems are *lexical features* that allow the classifier to consider the specific words when making a coreference decision. A few researchers have experimented with lexical features, but they achieved mixed results in evaluations on broad-coverage corpora (Bengston and Roth, 2008; Björkelund and Nugues, 2011; Rahman and Ng, 2011a).

We hypothesized that lexicalized features can have a more substantial impact in domain-specific settings. Lexical features can capture domain-specific knowledge and subtle semantic distinctions that may be important within a domain. For example, based on the resolutions found in domain-specific training sets, our lexicalized features captured the knowledge that "tomcat" can be coreferent with "plane", "UAW" can be coreferent with "union", and "anthrax" can be coreferent with "diagnosis". Capturing these types of domain-specific information is often impossible using only general-purpose resources. For example, WordNet defines "tomcat" only as an animal, does not contain an entry for "UAW", and categorizes "anthrax" and "diagnosis" very differently.[1]

In this paper, we evaluate the impact of lexicalized features on 4 domains: management succession (MUC-6 data), vehicle launches (MUC-7 data), disease outbreaks (ProMed texts), and terrorism (MUC-4 data). We incorporate lexicalized feature sets into two different coreference architectures: Reconcile (Stoyanov et al., 2010), a pairwise coreference classifier, and Sieve (Raghunathan et al., 2010), a rule-based system. Our results show that lexicalized features significantly improve performance in all four domains and in both types of coreference architectures.

2 Related Work

We are not the first researchers to use lexicalized features for coreference resolution. However, pre-

[1] WordNet defines "anthrax" as a disease (condition/state) and "diagnosis" as an identification (discovery event).

Train \ Test	MUC-6			MUC-7			Promed			MUC-4		
	P	R	F	P	R	F	P	R	F	P	R	F
MUC-6	**80.79**	62.71	**70.61**	**84.33**	61.74	71.29	83.54	70.34	76.37	**80.22**	60.81	69.18
MUC-7	74.78	65.59	69.88	82.73	64.09	**72.23**	**85.29**	71.82	**77.98**	77.35	64.19	**70.16**
Promed	73.60	64.20	68.60	82.88	63.37	71.82	80.31	72.66	76.29	74.52	65.65	69.80
MUC-4	69.27	**65.66**	67.42	71.49	**67.22**	69.29	76.92	**74.25**	75.56	71.76	**67.37**	69.50

Table 1: Cross-domain B^3 (Bagga and Baldwin, 1998) results for Reconcile with its general feature set. The Paired Permutation test (Pesarin, 2001) was used for statistical significance testing and gray cells represent results that are not significantly different from the best result.

vious work has evaluated the benefit of lexical features only for broad-coverage data sets.

Bengston and Roth (2008) incorporated a *memorization* feature to learn which entities can refer to one another. They created a binary feature for every pair of head nouns, including pronouns. They reported no significant improvement from these features on the ACE 2004 data.

Rahman and Ng (2011a) also utilized lexical features, going beyond strict memorization with methods to combat data sparseness and incorporating semantic information. They created a feature for every ordered pair of head nouns (for pronouns and nominals) or full NPs (for proper nouns). *Semi-lexical features* were also used when one NP was a Named Entity, and *unseen features* were used when the NPs were not in the training set. Their features did yield improvements on both the ACE 2005 and OntoNotes-2 data, but the semi-lexical features included Named Entity classes as well as word-based features.

Rahman and Ng (2011b) explored the use of lexical features in greater detail and showed their benefit on the ACE05 corpus independent of, and combined with, a conventional set of coreference features. The ACE05 corpus is drawn from six sources (Newswire, Broadcast News, Broadcast Conversations, Conversational Telephone Speech, Webblogs, and Usenet). The authors experimented with utilizing lexical information drawn from different sources. The results showed that the best performance came from training and testing with lexical knowledge drawn from the same source. Although our approach is similar, this paper focuses on learning lexical information from different *domains* as opposed to the different genres found in the six sources of the ACE05 corpus.

Björkelund and Nugues (2011) used lexical word pairs for the 2011 CoNLL Shared Task, showing significant positive impact on performance. They used over 2000 annotated documents from the broad-coverage OntoNotes corpus for training. Our work aims to show the benefit of lexical features using much smaller training sets (< 50 documents) focused on specific domains.

Lexical features have also been used for slightly different purposes. Florian et al. (2004) utilized lexical information such as mention spelling and context for entity tracking in ACE. Ng (2007) used lexical information to assess the likelihood of a noun phrase being anaphoric, but this did not show clear improvements on ACE data.

There has been previous work on domain-specific coreference resolution for several domains, including biological literature (Castaño et al., 2002; Liang and Lin, 2005; Gasperin and Briscoe, 2008; Kim et al., 2011; Batista-Navarro and Ananiadou, 2011), clinical medicine (He, 2007; Zheng et al., 2011; Glinos, 2011; Gooch and Roudsari, 2012) and legal documents (Bouayad-Agha et al., 2009). In addition, BABAR (Bean and Riloff, 2004) used *contextual role knowledge* for coreference resolution in the domains of terrorism and natural disasters. But BABAR acquired and used lexical information to match the compatibility of contexts surrounding NPs, not the NPs themselves. To the best of our knowledge, our work is the first to examine the impact of lexicalized features for domain-specific coreference resolution.

3 Exploiting Lexicalized Features

Table 1 shows the performance of a learning-based coreference resolver, Reconcile (Stoyanov et al., 2010), with its default feature set using different combinations of training and testing data. Reconcile does not include any lexical features, but does contain over 60 general features covering semantic agreement, syntactic constraints, string match and recency.

Each row represents a training set, each column represents a test set, and each cell shows precision (P), recall (R), and F score results under the B^3 metric when using the corresponding training and test data. The best results for each test set appear

	MUC-6			MUC-7			ProMED			MUC-4		
	P	R	F	P	R	F	P	R	F	P	R	F
Reconcile	80.79	62.71	70.61	82.73	64.09	72.23	80.31	72.66	76.29	71.76	67.37	69.50
+LexLookup	**87.01**	63.40	73.35	**87.39**	62.86	73.12	**86.66**	70.95	78.02	**82.89**	67.53	**74.42**
+LexSets	86.50	**63.76**	**73.41**	85.86	**64.35**	**73.56**	86.19	**72.14**	**78.54**	81.98	**67.73**	74.18
Sieve	**92.20**	61.70	73.90	**91.46**	59.59	72.16	**94.43**	67.25	78.55	**91.30**	59.84	72.30
+LexBegin	91.22	62.97	74.51	91.24	60.28	72.59	93.51	**69.15**	**79.51**	89.01	62.84	73.67
+LexEnd	90.59	**63.47**	**74.64**	91.17	**60.56**	**72.78**	93.99	68.87	79.49	89.04	**64.03**	**74.47**

Table 2: B^3 results for baselines and lexicalized feature sets across four domains.

in **boldface**.

We performed statistical significance testing using the Paired Permutation test (Pesarin, 2001) and the gray cells represent results where there was not significant difference from the best results in the same column. If just one cell is gray in a column, that indicates the result was significantly better than the other results in the same column with $p \leq 0.05$.

Table 1 does not show much benefit from training on the same domain as the test set. Three different training sets produce F scores that are not significantly different for both the MUC-6 and MUC-4 test data. For ProMed, training on the MUC-7 data yields significantly better results than training on all the other data sets, including ProMed texts! Based on these results, it would seem that training on the MUC-7 texts is likely to yield the best results no matter what domain you plan to use the coreference resolver for. The goal of our work is to investigate whether lexical features can extract additional knowledge from domain-specific training texts to help tailor a coreference resolver to perform better for a specific domain.

3.1 Extracting Coreferent Training Pairs

We adopt the terminology introduced by Stoyanov et al. (2009) to define a coreference element (CE) as a noun phrase that can participate in a coreference relation based on the task definition.

Each training document has manually annotated gold coreference chains corresponding to the sets of CEs that are coreferent. For each CE in a gold chain, we pair that CE with all of the other CEs in the same chain. We consider the coreference relation to be bi-directional, so we don't retain information about which CE was the antecedent. We do not extract CE pairs that share the same head noun because they are better handled with string match. For nominal NPs, we retain only the head noun, but we use the entire NP for proper names. We discard pairs that include a pronoun, and normalize strings to lower case for consistency.

3.2 Lexicalized Feature Sets

We explore two ways to capture lexicalized information as features. The first approach indicates whether two CEs have ever been coreferent in the training data. We create a single feature called LEXLOOKUP(X,Y) that receives a value of 1 when x and y have been coreferent at least twice, or a value of 0 otherwise.[2] LEXLOOKUP(X,Y) is a single feature that captures all CE pairs that were coreferent in the training data.

We also created *set-based* features that capture the set of terms that have been coreferent with a particular CE. The $CorefSet(x)$ is the set of CEs that have appeared in the same coreference chain as mention x at least twice.

We create a set of binary-valued features LEXSET(X,Y), one for each CE x in the training data. Given a pair of CEs, x and y, LEXSET(X,Y) = 1 if $y \in CorefSet(x)$, or 0 otherwise. The benefit of the set-based features over a single monolithic feature is that the classifier has one set-based feature for each mention found in the training data, so it can learn to handle individual terms differently.

We also tried encoding a separate feature for each distinct pair of words, analogous to the memorization feature in Bengston and Roth (2008). This did not improve performance as much as the other feature representations presented here.

4 Evaluation

4.1 Data Sets

We evaluated the performance of lexicalized features on 4 domain-specific corpora including two standard coreference benchmarks, the MUC-6 and MUC-7 data sets. The MUC-6 domain is management succession and consists of 30 training texts and 30 test texts. The MUC-7 domain is vehicle

[2]We require a frequency ≥ 2 to minimize overfitting because many cases occur only once in the training data.

launches and consists of 30 training texts and 20 test texts. We used these standard train/test splits to be consistent with previous work.

We also created 2 new coreference data sets which we will make freely available. We manually annotated 45 ProMed-mail articles (www.promedmail.org) about disease outbreaks and 45 MUC-4 texts about terrorism, following the MUC guidelines (Hirschman, 1997). Inter-annotator agreement between two annotators was .77 (κ) on ProMed and .84 (MUC F Score)(Villain et al., 1995) on both ProMed and MUC-4.[3] We performed 5-fold cross-validation on both data sets and report the micro-averaged results.

Gold CE spans were used in all experiments to factor out issues with markable identification and anaphoricity across the different domains.

4.2 Coreference Resolution Models

We conducted experiments using two coreference resolution architectures. Reconcile[4] (Stoyanov et al., 2010) is a freely available pairwise mention classifier. For classification, we chose Weka's (Witten and Frank, 2005) Decision Tree learner inside Reconcile. Reconcile contains roughly 60 features (none lexical), largely modeled after Ng and Cardie (2002). We modified Reconcile's Single Link clustering scheme to enforce an additional rule that non-overlapping proper names cannot be merged into the same chain.

We also conducted experiments with the Sieve coreference resolver, which applies high precision heuristic rules to incrementally build coreference chains. We implemented the LEXLOOKUP(X,Y) feature as an additional heuristic rule. We tried inserting this heuristic before Sieve's other rules (LexBegin), and also after Sieve's other rules (LexEnd).

4.3 Experimental Results

Table 2 presents results for Reconcile trained with and without lexical features and when adding a lexical heuristic with data drawn from same-domain texts to Sieve.

The first row shows the results without the lexicalized features (from Table 1). All F scores for Reconcile with lexicalized features are significantly better than without these features based on the Paired Permutation test (Pesarin, 2001) with $p \leq 0.05$. MUC-4 showed the largest gain for Reconcile, with the F score increasing from 69.5 to over 74. For most domains, adding the lexical features to Reconcile substantially increased precision with comparable levels of recall.

The bottom half of Table 2 contains the results of adding a lexical heuristic to Sieve. The first row shows the default system with no lexical information. All F scores with the lexical heuristic are significantly better than without it. In Sieve's high-precision coreference architecture, the lexical heuristic yields additional recall gains without sacrificing much precision.

	ACE 2004		
	P	R	F
Reconcile	70.59	83.09	76.33
+LexLookup	71.32	82.93	76.69
+LexSets	**71.44**	**83.45**	**76.98**
Sieve	**90.09**	74.23	**81.39**
+LexBegin	86.54	75.43	80.61
+LexEnd	87.00	**75.45**	80.82

Table 3: B^3 results for baselines and lexicalized feature sets on the broad-coverage ACE 2004 data set.

Table 3 shows the results for Reconcile and Sieve when training and testing on the ACE 2004 data. Here, we see little improvement from adding lexical information. For Reconcile, the small differences in F scores are not statistically significant. For Sieve, the unlexicalized system yields a significantly higher F score than when adding the lexical heuristic. These results support our hypothesis that lexicalized information can be beneficial for capturing domain-specific word associations, but may not be as helpful in a broad-coverage setting where the language covers a diverse set of topics.

Table 4 shows a re-evaluation of the cross-domain experiments from Table 1 for Reconcile with the LexSet features added. The bottom half of the table shows cross-domain experiments for Sieve using the lexical heuristic at the end of its rule set (LexEnd). Results are presented using both the B^3 metric and the MUC Score (Villain et al., 1995).

Training and testing on the same domain always produced the highest recall scores for MUC-7, ProMed, and MUC-4 when utilizing lexical features. In all cases, lexical features acquired from same-domain texts yield results that are either clearly the best or not significantly different from the best.

[3] We also computed κ on MUC-4, but unfortunately the score and original data were lost.

[4] http://www.cs.utah.edu/nlp/reconcile/

Train \ Test	MUC-6			MUC-7			Promed			MUC-4		
	P	R	F	P	R	F	P	R	F	P	R	F
Reconcile (B^3 Score)												
MUC-6	**86.50**	**63.76**	**73.41**	**90.44**	60.75	72.68	89.28	68.14	77.29	84.05	60.61	70.44
MUC-7	80.65	63.42	71.01	85.86	**64.46**	**73.56**	**89.41**	70.05	**78.55**	80.61	63.26	70.89
Promed	81.69	62.73	70.96	88.32	62.79	73.40	86.19	**72.14**	78.54	**84.81**	62.58	72.02
MUC-4	81.20	62.34	70.53	87.23	63.13	73.25	87.52	71.11	78.46	81.98	**67.73**	**74.18**
Reconcile (MUC Score)												
MUC-6	**89.56**	71.17	**79.32**	90.85	67.43	77.41	89.61	65.67	75.79	88.27	66.98	76.16
MUC-7	86.14	**72.22**	78.57	89.56	**72.01**	**79.83**	**89.34**	68.08	77.27	87.30	70.22	77.83
Promed	86.92	70.68	77.97	**90.93**	70.33	79.31	88.54	**69.55**	**77.90**	**88.83**	68.89	78.23
MUC-4	85.72	70.50	77.37	88.78	71.24	79.05	88.24	68.18	77.55	87.89	**74.18**	**80.45**
Sieve (B^3 Score)												
MUC-6	90.59	63.47	74.64	91.20	59.91	72.32	94.30	67.25	78.51	**91.30**	59.90	72.34
MUC-7	91.62	**63.67**	**75.13**	91.17	**60.56**	**72.78**	**94.43**	67.35	78.62	91.14	60.44	72.68
Promed	**92.14**	61.70	73.90	**91.46**	59.93	72.41	93.99	**68.87**	**79.49**	91.27	60.76	72.96
MUC-4	91.76	61.88	73.91	91.26	59.93	72.34	94.30	67.35	78.58	89.04	**64.03**	**74.47**
Sieve (MUC Score)												
MUC-6	91.80	**70.87**	**79.99**	91.38	65.52	76.32	92.08	64.71	76.01	90.38	66.98	77.10
MUC-7	**91.82**	69.70	79.25	**91.68**	**66.36**	**76.99**	**92.20**	64.86	76.15	90.71	67.09	77.13
Promed	91.99	69.15	78.95	**91.68**	65.52	76.42	91.70	**66.33**	**76.98**	**90.85**	67.09	77.18
MUC-4	91.79	69.39	79.03	91.48	65.52	76.36	92.00	64.86	76.08	90.31	**69.62**	**78.62**

Table 4: Cross-domain B^3 and MUC results for Reconcile and Sieve with lexical features. Gray cells represent results that are not significantly different from the best results in the column at the 0.05 p-level.

For MUC-6 and MUC-7, the highest F score results almost always come from training on same-domain texts, although in some cases these results are not significantly different from training on other domains. Lexical features can yield improvements when training on a different domain if there is overlap in the vocabulary across the domains. For the ProMed domain, the Sieve system performs significantly better, under both metrics, with same-domain lexical features than with lexical features acquired from a different domain. For Reconcile, there is not a significant difference in the F score for ProMed when training on ProMed, MUC-4, or MUC-7. In the MUC-4 domain, using same-domain lexical information *always* produces the best F score, under both metrics and in both coreference systems.

5 Conclusions

We explored the use of lexical information for domain-specific coreference resolution using 4 domain-specific data sets and 2 coreference resolvers. Lexicalized features consistently improved performance for all of the domains and in both coreference architectures. We see benefits from lexicalized features in cross-domain training, but the gains are often more substantial when utilizing same-domain lexical knowledge.

In the future, we plan to explore additional types of lexical information to benefit domain-specific coreference resolution.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1018314 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 reflect the view of the DARPA, AFRL, or the U.S. government.

References

ACE03. 2003. NIST ACE evaluation website. In *http://www.nist.gov/speech/tests/ace/2003*.

ACE04. 2004. NIST ACE evaluation website. In *http://www.nist.gov/speech/tests/ace/2004*.

ACE05. 2005. NIST ACE evaluation website. In *http://www.nist.gov/speech/tests/ace/2005*.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreference using the Vector Space Model. *Proceedings of the 17th international conference on Computational Linguistics (COLING)*.

Riza Theresa Batista-Navarro and Sophia Ananiadou. 2011. Building a coreference-annotated corpus from the domain of biochemistry. In *Proceedings of BioNLP 2011 Workshop*, BioNLP '11, pages 83–91.

David Bean and Ellen Riloff. 2004. Unsupervised learning of Contextual Role Knowledge for coreference resolution. *Proceedings of the HLT/NAACL 2004*.

Eric Bengston and Dan Roth. 2008. Understanding the value of features for coreference resolution. *Empirical Methods in Natural Language Processing*.

Anders Björkelund and Pierre Nugues. 2011. Exploring lexicalized features for coreference resolution. *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 45–50.

Nadjet Bouayad-Agha, Gerard Casamayor, Gabriela Ferraro, Simon Mille, Vanesa Vidal, and Leo Wanner. 2009. Improving the comprehension of legal documentation: the case of patent claims. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 78–87.

José Castaño, Jason Zhang, and James Pustejovsky. 2002. Anaphora resolution in biomedical literature. *International Symposium on Reference Resolution*.

Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov, Salim Roukos, and T Zhang. 2004. A statistical model for multilingual entity detection and tracking. *HLT-NAACL*.

Caroline Gasperin and Ted Briscoe. 2008. Statistical anaphora resolution in biomedical texts. *Proceedings of the 22nd Annual Conference on Computational Linguistics*, pages 257–264.

Demetrios G. Glinos. 2011. A search based method for clinical text coreference resolution. *In Proceedings of the Fifth i2b2/VA Track on Challenges in Natural Language Processing for Clinical Data (i2b2 2011)*.

Phil Gooch and Abdul Roudsari. 2012. Lexical patterns, features and knowledge resources for coreference resolution in clinical notes. *Journal of Biomedical Informatics*, 45.

Tian Ye He. 2007. *Coreference resolution on entities and events for hospital discharge summaries*. Ph.D. thesis, Massachusetts Institute of Technology.

Lynette Hirschman. 1997. MUC-7 task definition. *Proceedings of MUC-7*.

Youngjun Kim, Ellen Riloff, and Nathan Gilbert. 2011. The taming of Reconcile as a Biomedical coreference resolver. *ACL/HLT 2011 Workshop on Biomedical Natural Language Processing (BioNLP 2011) Shared Task Paper*.

Tyne Liang and Yu-Hsiang Lin. 2005. Anaphora resolution for biomedical literature by exploiting multiple resources. *Natural Language Processing–IJCNLP 2005*, pages 742–753.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. *Proceedings of the 40th Annual Meeting of the ACL*, pages 104–111.

Vincent Ng. 2007. Shallow semantics for coreference resolution. *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1689–1694.

Fortunato Pesarin. 2001. *Multivariate permutation tests: with applications in biostatistics*, volume 240. Wiley Chichester.

Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessice MacBride, and Linnea Micciulla. 2007. Unrestricted coreference: Identifying entities and events in ontonotes. *In Proceedings of the International Conference on Semantic Computing*.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A Multi-Pass Sieve for coreference resolution. *Empirical Methods in Natural Langugage Processing 2010*.

Altaf Rahman and Vincent Ng. 2011a. Coreference resolution with world knowledge. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics and Human Language Technologies (ACL-HLT)*, pages 814–824.

Altaf Rahman and Vincent Ng. 2011b. Narrowing the modelling gap: A cluster-ranking approach to coreference resolution. *Journal of Artificial Intelligence Research*.

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 the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP (ACL-IJCNLP 2009)*.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2010. Coreference resolution with Reconcile. *Proceedings of the Joint Conference of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*.

Marc Villain, John Aberdeen, John Berger, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. *Proceedings of the 6th conference on Message understanding*.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition.

Jiaping Zheng, Wendy Chapman, Rebecca Crowley, and Guergana Savova. 2011. Coreference resolution: A review of general methodologies and applications in the clinical domain. *Journal of Biomedical Informatics*, 44:1113–1122.