Exploring Various Knowledge in Relation Extraction.

Conference Paper · January 2005

DOI: 10.3115/1219840.1219893 · Source: DBLP

4 authors, including:

Guodong Zhou
Soochow University (PRC)
238 PUBLICATIONS **4,624** CITATIONS
SEE PROFILE

Su Jian
Institute for Infocomm Research (I2R), Agency for Science, Technology and Research
96 PUBLICATIONS **4,178** CITATIONS
SEE PROFILE

Exploring Various Knowledge in Relation Extraction

ZHOU GuoDong SU Jian ZHANG Jie ZHANG Min
Institute for Infocomm research
21 Heng Mui Keng Terrace, Singapore 119613
Email: {zhougd, sujian, zhangjie, mzhang}@i2r.a-star.edu.sg

Abstract

Extracting semantic relationships between entities is challenging. This paper investigates the incorporation of diverse lexical, syntactic and semantic knowledge in feature-based relation extraction using SVM. Our study illustrates that the base phrase chunking information is very effective for relation extraction and contributes to most of the performance improvement from syntactic aspect while additional information from full parsing gives limited further enhancement. This suggests that most of useful information in full parse trees for relation extraction is shallow and can be captured by chunking. We also demonstrate how semantic information such as WordNet and Name List, can be used in feature-based relation extraction to further improve the performance. Evaluation on the ACE corpus shows that effective incorporation of diverse features enables our system outperform previously best-reported systems on the 24 ACE relation subtypes and significantly outperforms tree kernel-based systems by over 20 in F-measure on the 5 ACE relation types.

1 Introduction

With the dramatic increase in the amount of textual information available in digital archives and the WWW, there has been growing interest in techniques for automatically extracting information from text. Information Extraction (IE) systems are expected to identify relevant information (usually of pre-defined types) from text documents in a certain domain and put them in a structured format.

According to the scope of the NIST Automatic Content Extraction (ACE) program, current research in IE has three main objectives: Entity Detection and Tracking (EDT), Relation Detection and Characterization (RDC), and Event Detection and Characterization (EDC). The EDT task entails the detection of entity mentions and chaining them together by identifying their coreference. In ACE vocabulary, entities are objects, mentions are references to them, and relations are semantic relationships between entities. Entities can be of five types: persons, organizations, locations, facilities and geo-political entities (GPE: geographically defined regions that indicate a political boundary, e.g. countries, states, cities, etc.). Mentions have three levels: names, nomial expressions or pronouns. The RDC task detects and classifies implicit and explicit relations[1] between entities identified by the EDT task. For example, we want to determine whether a person is at a location, based on the evidence in the context. Extraction of semantic relationships between entities can be very useful for applications such as question answering, e.g. to answer the query "Who is the president of the United States?".

This paper focuses on the ACE RDC task and employs diverse lexical, syntactic and semantic knowledge in feature-based relation extraction using Support Vector Machines (SVMs). Our study illustrates that the base phrase chunking information contributes to most of the performance inprovement from syntactic aspect while additional full parsing information does not contribute much, largely due to the fact that most of relations defined in ACE corpus are within a very short distance. We also demonstrate how semantic information such as WordNet (Miller 1990) and Name List can be used in the feature-based framework. Evaluation shows that the incorporation of diverse features enables our system achieve best reported performance. It also shows that our fea-

[1] In ACE (http://www.ldc.upenn.edu/Projects/ACE), explicit relations occur in text with explicit evidence suggesting the relationships. Implicit relations need not have explicit supporting evidence in text, though they should be evident from a reading of the document.

Proceedings of the 43rd Annual Meeting of the ACL, pages 427–434,
Ann Arbor, June 2005. ©2005 Association for Computational Linguistics

ture-based approach outperforms tree kernel-based approaches by 11 F-measure in relation detection and more than 20 F-measure in relation detection and classification on the 5 ACE relation types.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 and Section 4 describe our approach and various features employed respectively. Finally, we present experimental setting and results in Section 5 and conclude with some general observations in relation extraction in Section 6.

2 Related Work

The relation extraction task was formulated at the 7th Message Understanding Conference (MUC-7 1998) and is starting to be addressed more and more within the natural language processing and machine learning communities.

Miller et al (2000) augmented syntactic full parse trees with semantic information corresponding to entities and relations, and built generative models for the augmented trees. Zelenko et al (2003) proposed extracting relations by computing kernel functions between parse trees. Culotta et al (2004) extended this work to estimate kernel functions between augmented dependency trees and achieved 63.2 F-measure in relation detection and 45.8 F-measure in relation detection and classification on the 5 ACE relation types. Kambhatla (2004) employed Maximum Entropy models for relation extraction with features derived from word, entity type, mention level, overlap, dependency tree and parse tree. It achieves 52.8 F-measure on the 24 ACE relation subtypes. Zhang (2004) approached relation classification by combining various lexical and syntactic features with bootstrapping on top of Support Vector Machines.

Tree kernel-based approaches proposed by Zelenko et al (2003) and Culotta et al (2004) are able to explore the implicit feature space without much feature engineering. Yet further research work is still expected to make it effective with complicated relation extraction tasks such as the one defined in ACE. Complicated relation extraction tasks may also impose a big challenge to the modeling approach used by Miller et al (2000) which integrates various tasks such as part-of-speech tagging, named entity recognition, template element extraction and relation extraction, in a single model.

This paper will further explore the feature-based approach with a systematic study on the extensive incorporation of diverse lexical, syntactic and semantic information. Compared with Kambhatla (2004), we separately incorporate the base phrase chunking information, which contributes to most of the performance improvement from syntactic aspect. We also show how semantic information like WordNet and Name List can be equipped to further improve the performance. Evaluation on the ACE corpus shows that our system outperforms Kambhatla (2004) by about 3 F-measure on extracting 24 ACE relation subtypes. It also shows that our system outperforms tree kernel-based systems (Culotta et al 2004) by over 20 F-measure on extracting 5 ACE relation types.

3 Support Vector Machines

Support Vector Machines (SVMs) are a supervised machine learning technique motivated by the statistical learning theory (Vapnik 1998). Based on the structural risk minimization of the statistical learning theory, SVMs seek an optimal separating hyper-plane to divide the training examples into two classes and make decisions based on support vectors which are selected as the only effective instances in the training set.

Basically, SVMs are binary classifiers. Therefore, we must extend SVMs to multi-class (e.g. K) such as the ACE RDC task. For efficiency, we apply the *one vs. others* strategy, which builds K classifiers so as to separate one class from all others, instead of the *pairwise* strategy, which builds K*(K-1)/2 classifiers considering all pairs of classes. The final decision of an instance in the multiple binary classification is determined by the class which has the maximal SVM output. Moreover, we only apply the simple linear kernel, although other kernels can peform better.

The reason why we choose SVMs for this purpose is that SVMs represent the state-of–the-art in the machine learning research community, and there are good implementations of the algorithm available. In this paper, we use the binary-class SVMLight[2] deleveloped by Joachims (1998).

[2] Joachims has just released a new version of SVMLight for multi-class classification. However, this paper only uses the binary-class version. For details about SVMLight, please see http://svmlight.joachims.org/

4 Features

The semantic relation is determined between two mentions. In addition, we distinguish the argument order of the two mentions (M1 for the first mention and M2 for the second mention), e.g. M1-Parent-Of-M2 vs. M2-Parent-Of-M1. For each pair of mentions3, we compute various lexical, syntactic and semantic features.

4.1 Words

According to their positions, four categories of words are considered: 1) the words of both the mentions, 2) the words between the two mentions, 3) the words before M1, and 4) the words after M2. For the words of both the mentions, we also differentiate the head word[4] of a mention from other words since the head word is generally much more important. The words between the two mentions are classified into three bins: the first word in between, the last word in between and other words in between. Both the words before M1 and after M2 are classified into two bins: the first word next to the mention and the second word next to the mention. Since a pronominal mention (especially neutral pronoun such as 'it' and 'its') contains little information about the sense of the mention, the coreference chain is used to decide its sense. This is done by replacing the pronominal mention with the most recent non-pronominal antecedent when determining the word features, which include:

- WM1: bag-of-words in M1
- HM1: head word of M1
- WM2: bag-of-words in M2
- HM2: head word of M2
- HM12: combination of HM1 and HM2
- WBNULL: when no word in between
- WBFL: the only word in between when only one word in between
- WBF: first word in between when at least two words in between
- WBL: last word in between when at least two words in between
- WBO: other words in between except first and last words when at least three words in between
- BM1F: first word before M1
- BM1L: second word before M1
- AM2F: first word after M2
- AM2L: second word after M2

4.2 Entity Type

This feature concerns about the entity type of both the mentions, which can be PERSON, ORGANIZATION, FACILITY, LOCATION and Geo-Political Entity or GPE:

- ET12: combination of mention entity types

4.3 Mention Level

This feature considers the entity level of both the mentions, which can be NAME, NOMIAL and PRONOUN:

- ML12: combination of mention levels

4.4 Overlap

This category of features includes:

- #MB: number of other mentions in between
- #WB: number of words in between
- M1>M2 or M1<M2: flag indicating whether M2/M1is included in M1/M2.

Normally, the above overlap features are too general to be effective alone. Therefore, they are also combined with other features: 1) ET12+M1>M2; 2) ET12+M1<M2; 3) HM12+M1>M2; 4) HM12+M1<M2.

4.5 Base Phrase Chunking

It is well known that chunking plays a critical role in the Template Relation task of the 7th Message Understanding Conference (MUC-7 1998). The related work mentioned in Section 2 extended to explore the information embedded in the full parse trees. In this paper, we separate the features of base

[3] In ACE, each mention has a head annotation and an extent annotation. In all our experimentation, we only consider the word string between the beginning point of the extent annotation and the end point of the head annotation. This has an effect of choosing the base phrase contained in the extent annotation. In addition, this also can reduce noises without losing much of information in the mention. For example, in the case where the noun phrase "the former CEO of McDonald" has the head annotation of "CEO" and the extent annotation of "the former CEO of McDonald", we only consider "the former CEO" in this paper.

[4] In this paper, the head word of a mention is normally set as the last word of the mention. However, when a preposition exists in the mention, its head word is set as the last word before the preposition. For example, the head word of the name mention "University of Michigan" is "University".

phrase chunking from those of full parsing. In this way, we can separately evaluate the contributions of base phrase chunking and full parsing. Here, the base phrase chunks are derived from full parse trees using the Perl script[5] written by Sabine Buchholz from Tilburg University and the Collins' parser (Collins 1999) is employed for full parsing. Most of the chunking features concern about the head words of the phrases between the two mentions. Similar to word features, three categories of phrase heads are considered: 1) the phrase heads in between are also classified into three bins: the first phrase head in between, the last phrase head in between and other phrase heads in between; 2) the phrase heads before M1 are classified into two bins: the first phrase head before and the second phrase head before; 3) the phrase heads after M2 are classified into two bins: the first phrase head after and the second phrase head after. Moreover, we also consider the phrase path in between.

- CPHBNULL when no phrase in between
- CPHBFL: the only phrase head when only one phrase in between
- CPHBF: first phrase head in between when at least two phrases in between
- CPHBL: last phrase head in between when at least two phrase heads in between
- CPHBO: other phrase heads in between except first and last phrase heads when at least three phrases in between
- CPHBM1F: first phrase head before M1
- CPHBM1L: second phrase head before M1
- CPHAM2F: first phrase head after M2
- CPHAM2F: second phrase head after M2
- CPP: path of phrase labels connecting the two mentions in the chunking
- CPPH: path of phrase labels connecting the two mentions in the chunking augmented with head words, if at most two phrases in between

4.6 Dependency Tree

This category of features includes information about the words, part-of-speeches and phrase labels of the words on which the mentions are dependent in the dependency tree derived from the syntactic full parse tree. The dependency tree is built by using the phrase head information returned by the Collins' parser and linking all the other fragments in a phrase to its head. It also includes flags indicating whether the two mentions are in the same NP/PP/VP.

- ET1DW1: combination of the entity type and the dependent word for M1
- H1DW1: combination of the head word and the dependent word for M1
- ET2DW2: combination of the entity type and the dependent word for M2
- H2DW2: combination of the head word and the dependent word for M2
- ET12SameNP: combination of ET12 and whether M1 and M2 included in the same NP
- ET12SamePP: combination of ET12 and whether M1 and M2 exist in the same PP
- ET12SameVP: combination of ET12 and whether M1 and M2 included in the same VP

4.7 Parse Tree

This category of features concerns about the information inherent only in the full parse tree.

- PTP: path of phrase labels (removing duplicates) connecting M1 and M2 in the parse tree
- PTPH: path of phrase labels (removing duplicates) connecting M1 and M2 in the parse tree augmented with the head word of the top phrase in the path.

4.8 Semantic Resources

Semantic information from various resources, such as WordNet, is used to classify important words into different semantic lists according to their indicating relationships.

Country Name List

This is to differentiate the relation subtype "ROLE.Citizen-Of", which defines the relationship between a person and the country of the person's citizenship, from other subtypes, especially "ROLE.Residence", where defines the relationship between a person and the location in which the person lives. Two features are defined to include this information:

- ET1Country: the entity type of M1 when M2 is a country name
- CountryET2: the entity type of M2 when M1 is a country name

[5] http://ilk.kub.nl/~sabine/chunklink/

Personal Relative Trigger Word List

This is used to differentiate the six personal social relation subtypes in ACE: Parent, Grandparent, Spouse, Sibling, Other-Relative and Other-Personal. This trigger word list is first gathered from WordNet by checking whether a word has the semantic class "person|…|relative". Then, all the trigger words are semi-automatically[6] classified into different categories according to their related personal social relation subtypes. We also extend the list by collecting the trigger words from the head words of the mentions in the training data according to their indicating relationships. Two features are defined to include this information:

- ET1SC2: combination of the entity type of M1 and the semantic class of M2 when M2 triggers a personal social subtype.
- SC1ET2: combination of the entity type of M2 and the semantic class of M1 when the first mention triggers a personal social subtype.

5 Experimentation

This paper uses the ACE corpus provided by LDC to train and evaluate our feature-based relation extraction system. The ACE corpus is gathered from various newspapers, newswire and broadcasts. In this paper, we only model explicit relations because of poor inter-annotator agreement in the annotation of implicit relations and their limited number.

5.1 Experimental Setting

We use the official ACE corpus from LDC. The training set consists of 674 annotated text documents (~300k words) and 9683 instances of relations. During development, 155 of 674 documents in the training set are set aside for fine-tuning the system. The testing set is held out only for final evaluation. It consists of 97 documents (~50k words) and 1386 instances of relations. Table 1 lists the types and subtypes of relations for the ACE Relation Detection and Characterization (RDC) task, along with their frequency of occurrence in the ACE training set. It shows that the ACE corpus suffers from a small amount of annotated data for a few subtypes such as the subtype "Founder" under the type "ROLE". It also shows that the ACE RDC task defines some difficult subtypes such as the subtypes "Based-In", "Located" and "Residence" under the type "AT", which are difficult even for human experts to differentiate.

Type	Subtype	Freq
AT(2781)	Based-In	347
	Located	2126
	Residence	308
NEAR(201)	Relative-Location	201
PART(1298)	Part-Of	947
	Subsidiary	355
	Other	6
ROLE(4756)	Affiliate-Partner	204
	Citizen-Of	328
	Client	144
	Founder	26
	General-Staff	1331
	Management	1242
	Member	1091
	Owner	232
	Other	158
SOCIAL(827)	Associate	91
	Grandparent	12
	Other-Personal	85
	Other-Professional	339
	Other-Relative	78
	Parent	127
	Sibling	18
	Spouse	77

Table 1: Relation types and subtypes in the ACE training data

In this paper, we explicitly model the argument order of the two mentions involved. For example, when comparing mentions m1 and m2, we distinguish between m1-ROLE.Citizen-Of-m2 and m2-ROLE.Citizen-Of-m1. Note that only 6 of these 24 relation subtypes are symmetric: "Relative-Location", "Associate", "Other-Relative", "Other-Professional", "Sibling", and "Spouse". In this way, we model relation extraction as a multi-class classification problem with 43 classes, two for each relation subtype (except the above 6 symmetric subtypes) and a "NONE" class for the case where the two mentions are not related.

5.2 Experimental Results

In this paper, we only measure the performance of relation extraction on "true" mentions with "true" chaining of coreference (i.e. as annotated by the corpus annotators) in the ACE corpus. Table 2 measures the performance of our relation extrac-

[6] Those words that have the semantic classes "Parent", "GrandParent", "Spouse" and "Sibling" are automatically set with the same classes without change. However, The remaining words that do not have above four classes are manually classified.

tion system over the 43 ACE relation subtypes on the testing set. It shows that our system achieves best performance of 63.1%/49.5%/ 55.5 in precision/recall/F-measure when combining diverse lexical, syntactic and semantic features. Table 2 also measures the contributions of different features by gradually increasing the feature set. It shows that:

Features	P	R	F
Words	69.2	23.7	35.3
+Entity Type	67.1	32.1	43.4
+Mention Level	67.1	33.0	44.2
+Overlap	57.4	40.9	47.8
+Chunking	61.5	46.5	53.0
+Dependency Tree	62.1	47.2	53.6
+Parse Tree	62.3	47.6	54.0
+Semantic Resources	63.1	49.5	55.5

Table 2: Contribution of different features over 43 relation subtypes in the test data

- Using word features only achieves the performance of 69.2%/23.7%/35.3 in precision/recall/F-measure.
- Entity type features are very useful and improve the F-measure by 8.1 largely due to the recall increase.
- The usefulness of mention level features is quite limited. It only improves the F-measure by 0.8 due to the recall increase.
- Incorporating the overlap features gives some balance between precision and recall. It increases the F-measure by 3.6 with a big precision decrease and a big recall increase.
- Chunking features are very useful. It increases the precision/recall/F-measure by 4.1%/5.6%/ 5.2 respectively.
- To our surprise, incorporating the dependency tree and parse tree features only improve the F-measure by 0.6 and 0.4 respectively. This may be due to the fact that most of relations in the ACE corpus are quite local. Table 3 shows that about 70% of relations exist where two mentions are embedded in each other or separated by at most one word. While short-distance relations dominate and can be resolved by above simple features, the dependency tree and parse tree features can only take effect in the remaining much less long-distance relations. However, full parsing is always prone to long distance errors although the Collins' parser used in our system represents the state-of-the-art in full parsing.
- Incorporating semantic resources such as the country name list and the personal relative trigger word list further increases the F-measure by 1.5 largely due to the differentiation of the relation subtype "ROLE.Citizen-Of" from "ROLE. Residence" by distinguishing country GPEs from other GPEs. The effect of personal relative trigger words is very limited due to the limited number of testing instances over personal social relation subtypes.

Table 4 separately measures the performance of different relation types and major subtypes. It also indicates the number of testing instances, the number of correctly classified instances and the number of wrongly classified instances for each type or subtype. It is not surprising that the performance on the relation type "NEAR" is low because it occurs rarely in both the training and testing data. Others like "PART.Subsidary" and "SOCIAL. Other-Professional" also suffer from their low occurrences. It also shows that our system performs best on the subtype "SOCIAL.Parent" and "ROLE. Citizen-Of". This is largely due to incorporation of two semantic resources, i.e. the country name list and the personal relative trigger word list. Table 4 also indicates the low performance on the relation type "AT" although it frequently occurs in both the training and testing data. This suggests the difficulty of detecting and classifying the relation type "AT" and its subtypes.

Table 5 separates the performance of relation detection from overall performance on the testing set. It shows that our system achieves the performance of 84.8%/66.7%/74.7 in precision/recall/F-measure on relation detection. It also shows that our system achieves overall performance of 77.2%/60.7%/68.0 and 63.1%/49.5%/55.5 in precision/recall/F-measure on the 5 ACE relation types and the best-reported systems on the ACE corpus. It shows that our system achieves better performance by ~3 F-measure largely due to its gain in recall. It also shows that feature-based methods dramatically outperform kernel methods. This suggests that feature-based methods can effectively combine different features from a variety of sources (e.g. WordNet and gazetteers) that can be brought to bear on relation extraction. The tree kernels developed in Culotta et al (2004) are yet to be effective on the ACE RDC task.

Finally, Table 6 shows the distributions of errors. It shows that 73% (627/864) of errors results from relation detection and 27% (237/864) of errors results from relation characterization, among which 17.8% (154/864) of errors are from misclassification across relation types and 9.6% (83/864) of errors are from misclassification of relation subtypes inside the same relation types. This suggests that relation detection is critical for relation extraction.

# of relations		# of other mentions in between					
		0	1	2	3	>=4	Overall
# of the words in between	0	3991	161	11	0	0	4163
	1	2350	315	26	2	0	2693
	2	465	95	7	2	0	569
	3	311	234	14	0	0	559
	4	204	225	29	2	3	463
	5	111	113	38	2	1	265
	>=6	262	297	277	148	134	1118
	Overall	7694	1440	402	156	138	**9830**

Table 3: Distribution of relations over #words and #other mentions in between in the training data

Type	Subtype	#Testing Instances	#Correct	#Error	P	R	F
AT		**392**	**224**	**105**	**68.1**	**57.1**	**62.1**
	Based-In	85	39	10	79.6	45.9	58.2
	Located	241	132	120	52.4	54.8	53.5
	Residence	66	19	9	67.9	28.8	40.4
NEAR		**35**	**8**	**1**	**88.9**	**22.9**	**36.4**
	Relative-Location	35	8	1	88.9	22.9	36.4
PART		**164**	**106**	**39**	**73.1**	**64.6**	**68.6**
	Part-Of	136	76	32	70.4	55.9	62.3
	Subsidiary	27	14	23	37.8	51.9	43.8
ROLE		**699**	**443**	**82**	**84.4**	**63.4**	**72.4**
	Citizen-Of	36	25	8	75.8	69.4	72.6
	General-Staff	201	108	46	71.1	53.7	62.3
	Management	165	106	72	59.6	64.2	61.8
	Member	224	104	36	74.3	46.4	57.1
SOCIAL		**95**	**60**	**21**	**74.1**	**63.2**	**68.5**
	Other-Professional	29	16	32	33.3	55.2	41.6
	Parent	25	17	0	100	68.0	81.0

Table 4: Performance of different relation types and major subtypes in the test data

System	Relation Detection			RDC on Types			RDC on Subtypes		
	P	R	F	P	R	F	P	R	F
Ours: feature-based	**84.8**	**66.7**	**74.7**	**77.2**	**60.7**	**68.0**	63.1	**49.5**	**55.5**
Kambhatla (2004):feature-based	-	-	-	-	-	-	**63.5**	45.2	52.8
Culotta et al (2004):tree kernel	81.2	51.8	63.2	67.1	35.0	45.8	-	-	-

Table 5: Comparison of our system with other best-reported systems on the ACE corpus

Error Type		#Errors
Detection Error	False Negative	462
	False Positive	165
Characterization Error	Cross Type Error	154
	Inside Type Error	83

Table 6: Distribution of errors

6 Discussion and Conclusion

In this paper, we have presented a feature-based approach for relation extraction where diverse lexical, syntactic and semantic knowledge are employed. Instead of exploring the full parse tree information directly as previous related work, we incorporate the base phrase chunking information first. Evaluation on the ACE corpus shows that base phrase chunking contributes to most of the performance improvement from syntactic aspect while further incorporation of the parse tree and dependence tree information only slightly improves the performance. This may be due to three reasons: First, most of relations defined in ACE have two mentions being close to each other. While short-distance relations dominate and can be resolved by simple features such as word and chunking features, the further dependency tree and parse tree features can only take effect in the remaining much less and more difficult long-distance relations. Second, it is well known that full parsing is always prone to long-distance parsing errors although the Collins' parser used in our system achieves the state-of-the-art performance. Therefore, the state-of-art full parsing still needs to be further enhanced to provide accurate enough information, especially PP (Preposition Phrase) attachment. Last, effective ways need to be explored to incorporate information embedded in the full parse trees. Besides, we also demonstrate how semantic information such as WordNet and Name List, can be used in feature-based relation extraction to further improve the performance.

The effective incorporation of diverse features enables our system outperform previously best-reported systems on the ACE corpus. Although tree kernel-based approaches facilitate the exploration of the implicit feature space with the parse tree structure, yet the current technologies are expected to be further advanced to be effective for relatively complicated relation extraction tasks such as the one defined in ACE where 5 types and 24 subtypes need to be extracted. Evaluation on the ACE RDC task shows that our approach of combining various kinds of evidence can scale better to problems, where we have a lot of relation types with a relatively small amount of annotated data. The experiment result also shows that our feature-based approach outperforms the tree kernel-based approaches by more than 20 F-measure on the extraction of 5 ACE relation types.

In the future work, we will focus on exploring more semantic knowledge in relation extraction, which has not been covered by current research. Moreover, our current work is done when the Entity Detection and Tracking (EDT) has been perfectly done. Therefore, it would be interesting to see how imperfect EDT affects the performance in relation extraction.

References

Agichtein E. and Gravano L. (2000). Snowball: Extracting relations from large plain text collections. In *Proceedings of 5th ACM International Conference on Digital Libraries*. 4-7 June 2000. San Antonio, TX.

Brin S. (1998). Extracting patterns and relations from the World Wide Web. In *Proceedings of WebDB workshop at 6th International Conference on Extending DataBase Technology (EDBT'1998).*23-27 March 1998, Valencia, Spain

Collins M. (1999). Head-driven statistical models for natural language parsing. *Ph.D. Dissertation*, University of Pennsylvania.

Collins M. and Duffy N. (2002). Covolution kernels for natural language. In Dietterich T.G., Becker S. and Ghahramani Z. editors. *Advances in Neural Information Processing Systems 14*. Cambridge, MA.

Culotta A. and Sorensen J. (2004). Dependency tree kernels for relation extraction. In *Proceedings of 42th Annual Meeting of the Association for Computational Linguistics*. 21-26 July 2004. Barcelona, Spain

Cumby C.M. and Roth D. (2003). On kernel methods for relation learning. In Fawcett T. and Mishra N. editors. In Proceedings of 20th International Conference on Machine Learning (ICML'2003). 21-24 Aug 2003. Washington D.C. USA. AAAI Press.

Haussler D. (1999). Covention kernels on discrete structures. *Technical Report UCS-CRL-99-10*. University of California, Santa Cruz.

Joachims T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In *Proceedings of European Conference on Machine Learning(ECML'1998)*. 21-23 April 1998. Chemnitz, Germany

Miller G.A. (1990). WordNet: An online lexical database. *International Journal of Lexicography*. 3(4):235-312.

Miller S., Fox H., Ramshaw L. and Weischedel R. (2000). A novel use of statistical parsing to extract information from text. In *Proceedings of 6th Applied Natural Language Processing Conference*. 29 April - 4 May 2000, Seattle, USA

MUC-7. (1998). *Proceedings of the 7th Message Understanding Conference (MUC-7).* Morgan Kaufmann, San Mateo, CA.

Kambhatla N. (2004). Combining lexical, syntactic and semantic features with Maximum Entropy models for extracting relations. In *Proceedings of 42th Annual Meeting of the Association for Computational Linguistics*. 21-26 July 2004. Barcelona, Spain.

Roth D. and Yih W.T. (2002). Probabilistic reasoning for entities and relation recognition. In *Proceedings of 19th International Conference on Computational Linguistics(CoLING'2002)*. Taiwan.

Vapnik V. (1998). Statistical Learning Theory. Whiley, Chichester, GB.

Zelenko D., Aone C. and Richardella. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*. pp1083-1106.

Zhang Z. (2004). Weekly-supervised relation classification for Information Extraction. In Proceedings of ACM 13th Conference on Information and Knowledge Management (CIKM'2004). 8-13 Nov 2004. Washington D.C., USA.