Unsupervised Relation Extraction using Dependency Trees for Automatic Generation of Multiple-Choice Questions

Naveed Afzal[1], Ruslan Mitkov[1], Atefeh Farzindar[2]
[1] Research Institute for Information and Language Processing (RIILP)
University of Wolverhampton, Wolverhampton, UK
{N.Afzal@wlv.ac.uk, R.Mitkov@wlv.ac.uk}
[2] NLP Technologies Inc. 1255 University Street, Suite 1212
Montreal (QC), Canada, H3B 3W9
{farzindar@nlptechnologies.ca}

Abstract. In this paper, we investigate an unsupervised approach to Relation Extraction to be applied in the context of automatic generation of multiple-choice questions (MCQs). MCQs are a popular large-scale assessment tool making it much easier for test-takers to take tests and for examiners to interpret their results. Our approach to the problem aims to identify the most important semantic relations in a document without assigning explicit labels to them in order to ensure broad coverage, unrestricted to predefined types of relations. In this paper, we present an approach to learn semantic relations between named entities by employing a dependency tree model. Our findings indicate that the presented approach is capable of achieving high precision rates, which are much more important than recall in automatic generation of MCQs, and its enhancement with linguistic knowledge helps to produce significantly better patterns. The intended application for the method is an e-Learning system for automatic assessment of students' comprehension of training texts; however it can also be applied to other NLP scenarios, where it is necessary to recognise the most important semantic relations without any prior knowledge as to their types.

Keywords: E-Learning, Information Extraction, Relation Extraction, Biomedical domain, Dependency Tree, MCQ generation.

1 Introduction

Multiple choice questions (MCQs) also known as multiple-choice tests are a form of objective assessment in which a user selects one answer from a set of alternative choices for a given question. MCQs are straightforward to conduct and instantaneously provide an effective measure of test-takers performance and feedback test results to the learner. In many disciplines instructors use MCQs as a preferred assessment tool and it is estimated that 45% - 67% student assessments utilise MCQs [2]. The fast developments of e-Learning technologies have in turn stimulated method for automatic generation of MCQs and today they have become an actively

developing topic in application-oriented NLP research. The work done in the area of automatic generation of MCQs does not have a long history [e.g., 18, 19, 28, 3 and 10]. Most of the aforementioned approaches rely on the syntactic structure of a sentence.

We present a new approach to MCQs generation, where in order to automatically generate MCQs we first identify important concepts and the relationships between them in the input texts. In order to achieve this, we study unsupervised Information Extraction methods with the purpose of discovering the most significant concepts and relations in the domain texts, without any prior knowledge of their types or their exemplar instances (seeds). Information Extraction (IE) is an important problem in many information access applications. The goal is to identify instances of specific semantic relations between named entities of interest in the text. Named Entities (NE's) are generally noun phrases in the unstructured text e.g. names of persons, posts, locations and organisations while relationships between two or more NE's are described in a pre-defined way e.g. "interact with" is a relationship between two biological objects (proteins).

Dependency trees are regarded as a suitable basis for semantic patterns acquisition as they abstract away from the surface structure to represent relations between elements (entities) of a sentence. Semantic patterns represent semantic relations between elements of sentences. In a dependency tree a pattern is defined as a path in the dependency tree passing through zero or more intermediate nodes within a dependency tree [27]. An insight of usefulness of the dependency patterns was provided by [26] in their work as they revealed that dependency parsers have the advantage of generating analyses which abstract away from the surface realisation of text to a greater extent than phrase structure grammars tend to, resulting in semantic information being more accessible in the representation of the text which can be useful for IE.

The main advantage of our approach is that it can cover a potentially unrestricted range of semantic relations while most supervised and semi-supervised approaches can learn to extract only those relations that have been exemplified in annotated text, seed patterns. Our assumption for Relation Extraction (RE) is that it is between NE's stated in the same sentence and that presence or absence of relation is independent of the text prior to or succeeding the sentence. Moreover, our approach is suitable in situations where a lot of unannotated text is available as it does not require manually annotated text or seeds. These properties of the method can be useful, specifically, in such applications as MCQs generation [18, 19] or a pre-emptive approach in which viable IE patterns are created in advance without human intervention [23, 24].

2 Related Work

There is a large body of research dedicated to the problem of extracting relations from texts of various domains. Most previous work focused on supervised methods and tried to both extract relations and assign labels describing their semantic types. As a rule, these approaches required a manually annotated corpus, which is very laborious and time-consuming to produce.

Semi-supervised and unsupervised approaches relied on seeds patterns and/or examples of specific types of relations [1, 25]. An unsupervised approach based on clustering of candidate patterns for the discovery of the most important relation types among NE's from a newspaper domain was presented by [9]. In the biomedical domain, most approaches were supervised and relied on regular expressions to learn patterns [5], while semi-supervised approaches exploited pre-defined seed patterns and cue words [11, 17].

Several approaches in IE have relied on dependency trees in order to extract patterns for the automatic acquisition of IE systems [27, 25 and 7]. Apart from IE, [15] used dependency trees in order to infer rules for question answering while [29] had made use of dependency trees for paraphrase identification. Moreover, dependency parsers are used most recently in the systems which identify protein interactions in biomedical texts [13, 6].

In dependency parsing main objective is to describe syntactic analysis of a sentence using dependency links which shows the head-modifier relations between words. All the IE approaches that relied on dependency trees have used different pattern models based on the particular part of the dependency analysis. The motive behind all of these models is to extract the necessary information from text without being overly complex. All of the pattern models have made use of the semantic patterns based on the dependency trees for the identification of items of interest in text. These models vary in terms of their complexity, expressivity and performance in an extraction scenario.

3 Our Approach

Our approach is based on the Linked Chain Pattern Model presented by [7]. Linked Chain Pattern Model combines the pair of chains in a dependency tree which share common verb root but no direct descendants.

In our approach, we have treated every NE as a chain in a dependency tree if it is less than 5 dependencies away from the verb root and the word linking the NE's to the verb root are from the category of content words (Verb, Noun, Adverb and Adjective) along with prepositions. We consider only those chains in the dependency tree of a sentence which contain NE's, which is much more efficient than the subtree model of [27], where all subtrees containing verbs are taken into account. This allows us to extract more meaningful patterns from the dependency tree of a sentence. We extract all NE chains which follow aforementioned rule from a sentence and combine them together. Figure 1 shows the whole system architecture.

According to the system architecture, in Section 3, we elaborate the NER process, Section 4 explains the process of candidates patterns extraction, we use GENIA corpus for candidate patterns extraction. Section 5 describes various information theoretic measures and statistical tests for patterns ranking depending upon their associations with domain corpus. Section 6 discusses the evaluation procedures (rank-thresholding and score-thresholding); GENIA EVENT Annotation corpus is used for evaluation while Section 7 explains the experimental results obtained via various patterns ranking methods.

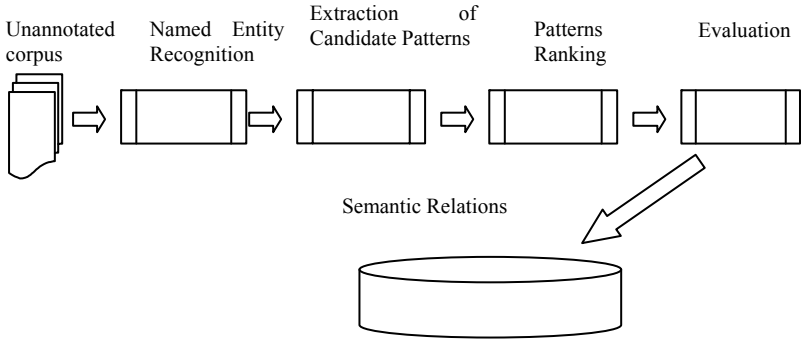

Figure 1. System Architecture

4 Named Entity Recognition (NER)

NER is an integral part of any IE system as it identifies NE's present in a text. Presently many NER tools are developed for various domains as there is a lot of research being done in the area of NER spreading across various languages, domains and textual genres. In our work, we used biomedical data as biomedical NER is generally considered to be more difficult as compared to other domains like newswire text. There are huge numbers of NE's in the biomedical domain and the new ones are consistently added [32] which means that neither dictionaries nor training data approach will be sufficiently comprehensive for NER task. The volume of published biomedical research is expanding at a rapid rate in the recent past. Due to the syntactic and semantic complexity of biomedical domain many IE systems have utilised tools (e.g., part-of-speech tagger, NER, parsers) specifically designed and developed for the biomedical domain [21]. Moreover, [8] presented a report, investigating the suitability of current NLP resources for syntactic and semantic analysis for biomedical domain.

The GENIA NER[1] [31, 32] is a specific tool designed for biomedical texts; the NE tagger is designed to recognise mainly the following NE's: protein, DNA, RNA, cell_type and cell_line. Table 1 shows the performance of GENIA NER[3].

Table 1. GENIA NER Performance

Entity Type	Precision	Recall	F-score
Protein	65.82	81.41	72.79
DNA	65.64	66.76	66.20
RNA	60.45	68.64	64.29
Cell Type	56.12	59.60	57.81
Cell Line	78.51	70.54	74.31
Overall	67.45	75.78	71.37

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

5 Extraction of Candidate Patterns

Our general approach to learn dependency tree-based patterns consists of two main stages: (i) the construction of potential patterns from an unannotated domain corpus and (ii) their relevance ranking.

After NER the next step is the construction of candidate patterns. We will explain the whole process of candidate patterns extraction from the dependency trees with the help of an example shown below:

Fibrinogen activates NF-kappaB transcription factors in mononuclear phagocytes.

After the NER the aforementioned sentence is transformed into following:

<protein> Fibrinogen </protein> activates <protein> NF-kappaB </protein> <protein> transcription factors </protein> in <cell_type> mononuclear phagocytes </cell_type>.

Once the NE's are recognised in the domain corpus by the GENIA tagger, we replace all the NE's with their semantic class respectively, so the aforementioned sentence is transformed into following sentence.

PROTEIN activates PROTEIN PROTEIN in CELL.

The transformed sentences are then parsed by using the Machinese Syntax[2] parser [30]. Machinese Syntax parser uses a functional dependency grammar for parsing. The analyses produced by the Machinese Syntax parser are encoded to make the most of information they contain and ensure consistent structures from which patterns could be extracted. Figure 2 shows the dependency tree for the aforementioned adapted sentence:

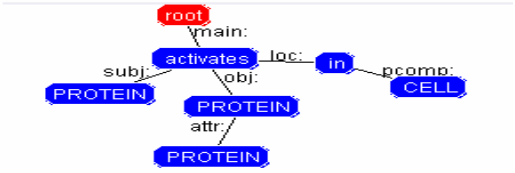

Figure 2. Example of a dependency tree

After the encoding process, the patterns are extracted from dependency trees using the methodology describe in Section 3. From Figure 2, the following patterns are extracted:

<NE ID="0" func="SUBJ" Dep="1"> "PROTEIN" </NE>
<W ID="1" func="+FMAINV" Dep="none">"activate"</W>
<NE ID="2" func="A" Dep="3"> "PROTEIN" </NE>
<NE ID="3" func="OBJ" Dep="1"> "PROTEIN" </NE>
<W ID="0" func="+FMAINV" Dep="none">"activate"</W>
<NE ID="1" func="A" Dep="2"> "PROTEIN" </NE>
<NE ID="2" func="OBJ" Dep="0"> "PROTEIN" </NE>

[2] http://www.connexor.com/software/syntax/

<W ID="0" func="+FMAINV" Dep="none">"activate"</W>
<NE ID="1" func="OBJ" Dep="0"> "PROTEIN" </NE>
<W ID="2" func="PREP" Dep="0">"in"</W>
<NE ID="3" func="P" Dep="2"> "CELL_TYPE" </NE>

Here *<NE>* tag represents the Named Entity (semantic class) while *<W>* tag represent the lexical words while *ID* represent the word id, *func* represent function of the word and *Dep* represents the id of the word on which this word depends in a dependency tree. The extracted patterns along with their frequencies are then stored in a database. We filtered out the patterns containing only stop-words in dependency-based patterns using stop-words corpus. Table 2 shows the examples of dependency-based patterns along with their frequencies.

Table 2. Example of dependency-based patterns along with frequencies

Patterns	Frequency
<NE ID="0" func="SUBJ" Dep="1"> "DNA" </NE> <W ID="1" func="+FMAINV" Dep="none">"contain"</W> <NE ID="2" func="OBJ" Dep="1"> "DNA" </NE>	34
<NE ID="0" func="SUBJ" Dep="1"> "PROTEIN" </NE> <W ID="1" func="+FMAINV" Dep="none">"activate"</W> <NE ID="2" func="OBJ" Dep="1"> "PROTEIN" </NE>	32
<NE ID="0" func="SUBJ" Dep="1"> "PROTEIN" </NE> <W ID="1" func="+FMAINV" Dep="none">"contain"</W> <NE ID="2" func="OBJ" Dep="1"> "PROTEIN" </NE>	19
<NE ID="0" func="SUBJ" Dep="2"> "PROTEIN" </NE> <NE ID="1" func="APP" Dep="0">"PROTEIN" </NE> <W ID="2" func="+FMAINV" Dep="none">"induce"</W>	19

6 Pattern Ranking

After candidate patterns have been constructed, the next step is to rank the patterns based on their significance in the domain corpus. The ranking methods we use require a general corpus that serves as a source of examples of pattern use in domain-independent texts. To extract candidates from the general corpus, we treated every noun as a potential NE holder and the candidate construction procedure described above was applied to find potential patterns in the general corpus. In order to score candidate patterns for domain-relevance, we measure the strength of association of a pattern with the domain corpus as opposed to the general corpus. The patterns are scored using the following methods for measuring the association between a pattern and the domain corpus: Information Gain (IG), Information Gain Ratio (IGR), Mutual Information (MI), Normalised Mutual Information (NMI)[3], Log-likelihood (LL) and Chi-Square (CHI). These association measures were included in the study as they

[3] Mutual Information has a well-known problem of being biased towards infrequent events. To tackle this problem, we normalised the MI score by a discounting factor, following the formula proposed in Lin and Pantel (2001).

have different theoretical principles behind them: IG, IGR, MI and NMI are information-theoretic concepts while LL and CHI are statistical tests of association.

Information Gain measures the amount of information obtained about domain specialisation of corpus *c*, given that pattern *p* is found in it.

$$IG(p,c) = \sum_{d \in \{c,c'\}} \sum_{g \in \{p,p'\}} P(g,d) \log \frac{P(g,d)}{P(g)P(d)}$$

where *p* is a candidate pattern, *c* – the domain corpus, *p'* – a pattern other than *p*, *c'* – the general corpus, *P*(*c*) – the probability of *c* in "overall" corpus {*c*,*c'*}, and *P*(*p*) – the probability of *p* in the overall corpus.

Information Gain Ratio aims to overcome one disadvantage of IG consisting of the fact that IG grows not only with the increase of dependence between *p* and *c*, but also with the increase of the entropy of *p*. IGR removes this factor by normalizing IG by the entropy of the patterns in the corpora:

$$IGR(p,c) = \frac{IG(g,c)}{-\sum_{g \in \{p,p'\}} P(g) \log P(g)}$$

Pointwise Mutual Information between corpus *c* and pattern *p* measures how much information the presence of *p* contains about *c*, and vice versa:

$$MI(p,c) = \log \frac{P(p,c)}{P(p)P(c)}$$

Chi-Square and Log-likelihood are statistical tests which work with frequencies and rank-order scales, both calculated from a contingency table with observed and expected frequency of occurrence of a pattern in the domain corpus. **Chi-Square** is calculated as follows:

$$x^2(p,c) = \sum_{d \in \{c,c'\}} \frac{(O_d - E_d)^2}{E_d}$$

where *O* is the observed frequency of *p* in domain and general corpus respectively and *E* is the expected frequency of *p* in two corpora.

Log-likelihood is calculated according to the following formula:

$$LL(p,c) = 2\left(O_1 \log\left(\frac{O_1}{E_1}\right) + O_2 \log\left(\frac{O_2}{E_2}\right)\right)$$

where O_1 and O_2 are observed frequencies of *p* in the domain and general corpus respectively, while E_1 and E_2 are its expected frequency values in the two corpora.

In addition to these six measures, we introduce a **meta-ranking** method that combines the scores produced by several individual association measures, in order to leverage agreement between different association measures and downplay idiosyncrasies of individual ones. Because the association functions range over

different values (for example, IGR ranges between 0 and 1, and MI between $+\infty$ and $-\infty$), we first normalise the scores assigned by each method[4]:

$$s_{norm}(p) = \frac{s(p)}{\max_{q \in P}(s(q))}$$

where $s(p)$ is the non-normalised score for pattern p, from the candidate pattern set P. The normalised scores are then averaged across different methods and used to produce a meta-ranking of the candidate patterns.

Given the ranking of candidate patterns produced by a scoring method, a certain number of highest-ranking patterns can be selected for evaluation. We studied two different ways of selecting these patterns: (i) one based on setting a threshold on the association score below which the candidate patterns are discarded (henceforth, *score-thresholding method*) and (ii) one that selects a fixed number of top-ranking patterns (henceforth, *rank-thresholding method*). During the evaluation, we experimented with different rank- and score-thresholding values.

7 Evaluation

Biomedical NE's are expressed in various linguistic forms such as abbreviations, plurals, compound, coordination, cascades, acronyms and apposition. Sentences in such texts are syntactically complex as the subsequent Relation Extraction phase depends upon the correct identification of the named entities and correct analysis of linguistic constructions expressing relations between them [34].

We used the GENIA Corpus as the domain corpus while British National Corpus (BNC) was used as a general corpus. GENIA corpus consists of 2,000 abstracts extracted from the MEDLINE containing 18,477 sentences. In the evaluation phase, GENIA EVENT Annotation corpus[5] is used [14]. It consists of 9,372 sentences. The numbers of dependency patterns extracted from each corpus are: GENIA 5066, BNC 419274 and GENIA EVENT 3031 respectively.

In order to evaluate the quality of the extracted patterns, we examined their ability to capture pairs of related NE's in the manually annotated evaluation corpus, without recognising the type of semantic relation. Selecting a certain number of best-ranking patterns, we measure precision, recall and F-score. To test the statistical significance of differences in the results of different methods and configurations, we used a paired t-test, having randomly divided the evaluation corpus into 20 subsets of equal size; each subset containing 461 sentences on average.

8 Results

Table 3 shows the results of precision scores for ranked-thresholding method.

[4] Patterns with negative MI scores are discarded.

[5] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi?page=Event+Annotation

Table 3. Precision scores of rank-thresholding method

Ranking Methods	Dependency Tree Patterns		
	Top 100 Ranked Patterns	**Top 200 Ranked Patterns**	**Top 300 Ranked Patterns**
IG	0.770	0.800	0.780
IGR	0.770	0.800	0.787
MI	0.560	0.560	0.540
NMI	0.940	0.815	0.707
LL	0.770	0.800	**0.790**
CHI	**0.960**	**0.815**	0.710
Meta	0.900	0.830	0.740

Table 4 shows the results of score-thresholding method, the left side of the Table 4 shows the precision (P), recall (R) and F-score values for score-threshold values where we are able to achieve high F-scores while right side of the Table 4 shows the high precision scores.

Table 4. Results of score-thresholding method

Ranking Methods	Dependency Tree Patterns					
	P	**R**	**F-score**	**P**	**R**	**F-score**
	Threshold score > 0.01			*Threshold score > 0.09*		
IG	0.748	0.107	0.187	0.733	0.007	0.014
IGR	0.748	0.107	0.187	0.733	0.007	0.014
MI	0.567	0.816	0.669	0.563	0.593	0.578
NMI	0.566	0.767	0.651	0.572	0.507	0.538
LL	0.748	0.107	0.187	0.733	0.007	0.014
CHI	0.577	0.529	0.552	**0.900**	0.036	0.069
Meta	0.571	0.643	0.605	0.860	0.048	0.092
	Threshold score > 0.02			*Threshold score > 0.1*		
IG	0.796	0.051	0.097	0.704	0.006	0.012
IGR	0.796	0.051	0.097	0.704	0.006	0.012
MI	0.566	0.744	0.643	0.564	0.588	0.576
NMI	0.570	0.706	0.631	0.569	0.483	0.523
LL	0.796	0.051	0.097	0.704	0.006	0.012
CHI	0.591	0.243	0.344	**0.898**	0.035	0.067
Meta	0.569	0.547	0.558	0.856	0.047	0.089
	Threshold score > 0.03			*Threshold score > 0.2*		
IG	0.785	0.035	0.067	0.571	0.003	0.005
IGR	0.785	0.035	0.067	0.571	0.003	0.005
MI	0.566	0.711	0.631	0.566	0.473	0.515
NMI	0.568	0.663	0.612	0.600	0.133	0.218
LL	0.785	0.035	0.067	0.571	0.003	0.005
CHI	0.613	0.146	0.236	**1.000**	0.015	0.029
Meta	0.577	0.355	0.439	1.000	0.013	0.025

In both tables (3 and 4), the results of the best performing ranking method in terms of precision are shown in bold font. Although our main focus is on achieving higher precision scores it is quite obvious from Table 4 that our method achieved low recall, one reason of having a low recall is due to the small size of GENIA corpus which can be encountered by using a large corpus as large corpus will produce much greater number of patterns and increase the recall.

The CHI and NMI are the best performing ranking methods in terms of precision in both rank-thresholding and score-thresholding method while IG, IGR and LL achieve quite similar results. Moreover in Table 4 we are able to achieve 100% precision. Figure 3 shows the precision scores for the best performing ranking methods (CHI and NMI) in score-thresholding method.

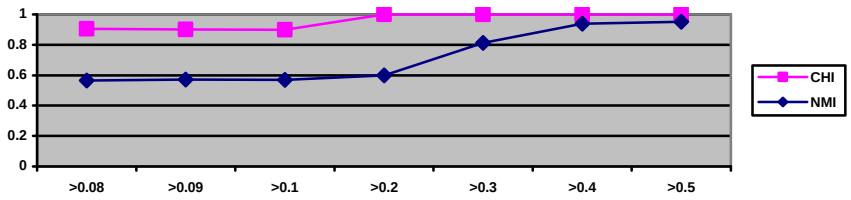

Figure 3. Example of a dependency tree

The literature on the topic suggests that IGR performs better than the IG [22, 16]; we found that in general there is no statistically significant difference between IG and IGR, IGR and LL. In both sets of experiments, obviously due to the aforementioned problem, MI performs quite poorly; the normalised version of MI helps to alleviate this problem. Moreover, there exists a statistically significant difference ($p < 0.01$) between NMI and the other ranking methods. The meta-ranking method did not improve on the best individual ranking method as expected.

We also find out that score-thresholding method produces better results than rank-thresholding as we are able to achieve up to 100% precision with the former technique. High precision is quite important in applications such as MCQ generation. In score-thresholding, it is possible to optimise for high precision (up to 100%), though recall and F-score is generally quite low. MCQ applications rely on the production of good questions rather than the production of all possible questions, so high precision plays a vital role in such applications.

9 Future work

In the future, we plan to employ the RE method for automatic MCQ generation, where it will be used to find relations and NE's in educational texts that are important for testing students' familiarity with key facts contained in the texts. In order to achieve this, we needed an IE method that has a high precision and at the same time works with unrestricted semantic types of relations (i.e. without reliance on seeds), while recall is of secondary importance to precision. The distractors will be produced using distributional similarity measures.

10 Conclusion

In this paper, we have presented an unsupervised approach for RE from dependency trees intended to be deployed in an e-Learning system for automatic generation of MCQs by employing semantic patterns. We explored different ranking methods and found that the CHI and NMI ranking methods obtained higher precision than the other ranking methods. We employed two techniques: the rank-thresholding and score-thresholding and found that score-thresholding perform better.

References

1. Agichtein, E. and Gravano, L.: Snowball: Extracting Relations from Large Plaintext Collections. In Proc. of the 5th ACM International Conference on Digital Libraries (2000).
2. Becker, W.E. and Watts, M.: Teaching methods in U.S. and undergraduate economics courses. Journal of Economics Education, 32(3), 269 – 279 (2001).
3. Brown, J., Frishkoff, G., and Eskenazi, M.: Automatic question generation for vocabulary assessment. In Proc. of HLT/EMNLP. Vancouver, B.C. (2005).
4. Cohen, A. M., and Hersh, W. R.: A Survey of Current Work in Biomedical Text Mining. Briefings in Bioinformatics, pp. 57-71 (2005).
5. Corney, D. P., Jones, D., Buxton, B., and Langdon, W.: BioRAT: Extracting Biological Information from Full-length Papers. Bioinformatics, pp. 3206-3213 (2004).
6. Erkan, G., Ozgur, A., and Radev, D.R.: Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In Proc. of CoNLL-EMNLP (2007).
7. Greenwood, M., Stevenson, M., Guo, Y., Harkema, H. and Roberts, A.: Automatically Acquiring a Linguistically Motivated Genic Interaction Extraction System. In Proc. of the 4th Learning Language in Logic Workshop, Bonn, Germany (2005).
8. Grover, C., Lascarides, A. and Lapata, M.: A Comparison of Parsing Technologies for the Biomedical Domain. Natural Language Engineering 11 (1), pp. 27 -65 (2005).
9. Hasegawa, T., Sekine, S., and Grishman, R.: Discovering relations among named entities from large corpora. In Proc. of ACL'04 (2004).
10. Hoshino, A. and Nakagawa, H.: A Real-time Multiple-choice Question Generation for Language Testing – A Preliminary Study. In Proc. of the 43rd ACL'05 2nd Workshop on Building Educational Applications Using Natural Language Processing, pp.17-20., Ann Arbor, U.S. (2005).
11. Huang, M., Zhu, X., Payan, G. D., Qu, K., and Li, M.: Discovering patterns to extract protein-protein interactions from full biomedical texts. Bioinformatics, pp. 3604-3612 (2004).
12. Jurafsky, D. and Martin, J. H.: Speech and Language Processing. Second Edition. Prentice Hall (2008).
13. Katrenko, S. and Adriaans, P.: Learning relations from biomedical corpora using dependency trees. In Proc. of the 1st International Workshop on Knowledge Discovery and Emergent Complexity in Bioinformatics, Ghent, pp. 61–80 (2006).
14. Kim, J-D., Ohta, T., and Tsujii, J.: Corpus Annotation for Mining Biomedical Events from Literature, BMC Bioinformatics (2008).
15. Lin, D. and Pantel, P.: Concept Discovery from Text. In Proc. of Conference on CL'02. pp. 577-583. Taipei, Taiwan (2002).
16. Manning, C. and Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, U.S. (1999).

17. Martin, E. P., Bremer, E., Guerin, G., DeSesa, M-C., Jouve, O.: Analysis of Protein/Protein Interactions through Biomedical Literature: Text Mining of Abstracts vs. Text Mining of Full Text Articles. Berlin: Springer-Verlag, pp. 96-108 (2004).
18. Mitkov, R. and An, L.A.: Computer-aided generation of multiple-choice tests. In Proc. of the HLT/NAACL 2003 Workshop on Building educational applications using Natural Language Processing, 17-22. Edmonton, Canada (2003).
19. Mitkov, R., Ha, L. A. and Karamanis, N.: A computer-aided environment for generating multiple-choice test items. Natural Language Engineering 12(2). Cambridge University Press, pp. 177-194 (2006).
20. Ono, T., Hishigaki, H., Tanigami, A. and Takagi, T.: Automated Extraction of Information on Protein–Protein Interactions from the Biological Literature. Bioinformatics, pp. 155-161 (2001).
21. Pustejovsky, J., J Casta, Cochran, B. and Kotecki, M.: Robust relational parsing over biomedical literature: Extracting inhibit relations. In Proc. of the 7th Annual Pacific Symposium on Bio-computing (2002).
22. Quinlan, J.R.: Induction of decision trees. Machine Learning, 1(1), pp. 81-106 (1986).
23. Sekine, S.: On-Demand Information Extraction. In Proc. of the COLING/ACL (2006).
24. Shinyama, Y. and Sekine, S.: Preemptive Information Extraction using Unrestricted Relation Discovery. In Proc. of the HLT Conference of the North American Chapter of the ACL. New York, pp. 304-311 (2006).
25. Stevenson, M. and Greenwood, M.: A Semantic Approach to IE Pattern Induction. In Proc. of ACL'05, pages 379-386 (2005).
26. Stevenson, M. and Greenwood, M.: Dependency Pattern Models for Information Extraction. Research on Language and Computation (2009).
27. Sudo, K., Sekine, S. and Grishman, R.: An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition. In Proc. of the 41st Annual Meeting of ACL-03, pp. 224–231, Sapporo, Japan (2003).
28. Sumita, E., Sugaya, F. and Yamamoto, S.: Measuring non-native speakers' proficiency of English using a test with automatically-generated fill-in-the-blank questions. In Proc. of the 2nd Workshop on Building Educational Applications using NLP, pp. 61-68 (2005).
29. Szpektor, I., Tanev, H., Dagan, I., and Coppola, B.: Scaling Web-based acquisition of Entailment Relations. In Proc. of EMNLP-04, Barcelona, Spain (2004).
30. Tapanainen, P. and Järvinen, T.: A Non-Projective Dependency Parser. In Proc. of the 5th Conference on Applied Natural Language Processing, pages 64–74, Washington, (1997).
31. Tsuruoka, Y., Tateishi, Y., Kim, J-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J.: Developing a Robust PoS Tagger for Biomedical Text. Advances in Informatics – 10th Panhellenic Conference on Informatics, LNCS 3746, pp. 382-392 (2005).
32. Tsuruoka, Y. and Tsujii, J.: Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. Proc. of HLT/EMNLP, pp. 467-474 (2005).
33. Wilbur, J., Smith, L. and Tanabe, T.: BioCreative 2. Gene Mention Task. Proc. of the 2nd Bio-Creative Challenge Workshop pp. 7-16 (2007).
34. Zhou, G., Su, J., Shen, D. and Tan, C.: Recognizing Name in Biomedical Texts: A Machine Learning Approach. Bioinformatics, pp. 1178-1190 (2004).