Adverse–Effect Relations Extraction from Massive Clinical Records

Yasuhide Miura [a]**, Eiji Aramaki** [b]**, Tomoko Ohkuma** [a]**, Masatsugu Tonoike** [a]**,**
Daigo Sugihara [a]**, Hiroshi Masuichi** [a] **and Kazuhiko Ohe** [c]

[a] Fuji Xerox Co., Ltd.

[b] Center for Knowledge Structuring, University of Tokyo

[c] University of Tokyo Hospital

```
yasuhide.miura@fujixerox.co.jp, eiji.aramaki@gmail.com,
{ohkuma.tomoko,masatsugu.tonoike,daigo.sugihara,
       hiroshi.masuichi}@fujixerox.co.jp,
          kohe@hcc.h.u-tokyo.ac.jp
```

Abstract

The rapid spread of electronic health records raised an interest to large-scale information extraction from clinical texts. Considering such a background, we are developing a method that can extract adverse drug event and effect (adverse–effect) relations from massive clinical records. Adverse–effect relations share some features with relations proposed in previous relation extraction studies, but they also have unique characteristics. Adverse–effect relations are usually uncertain. Not even medical experts can usually determine whether a symptom that arises after a medication represents an adverse–effect relation or not. We propose a method to extract adverse–effect relations using a machine-learning technique with dependency features. We performed experiments to extract adverse–effect relations from 2,577 clinical texts, and obtained F_1-score of 37.54 with an optimal parameters and F_1-score of 34.90 with automatically tuned parameters. The results also show that dependency features increase the extraction F_1-score by 3.59.

1 Introduction

The widespread use of electronic health records (EHR) made clinical texts to be stored as computer processable data. EHRs contain important information about patients' health. However, extracting clinical information from EHRs is not easy because they are likely to be written in a natural language.

We are working on a task to extract adverse drug event and effect relations from clinical records. Usually, the association between a drug and its adverse–effect relation is investigated using numerous human resources, costing much time and money. The motivation of our task comes from this situation. An example of the task is presented in Figure 1. We defined an adverse–effect relation as a relation that holds between a drug entity and a symptom entity. The sentence illustrates the occurrence of the adverse–effect *hepatic disorder* by the *Singulair* medication.
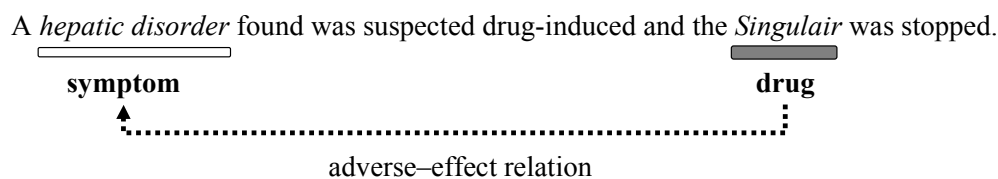

A *hepatic disorder* found was suspected drug-induced and the *Singulair* was stopped.

Figure 1. Example of an adverse–effect relation.

75

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 75–83,
Beijing, August 2010

ACTOS 30 recovered HbA1c to 6.5% but an *edema* appeared after the medication.

drug   **symptom**

adverse-effect relation
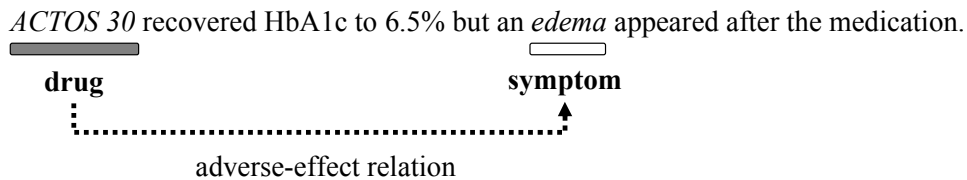
Figure 2. The example of an adverse-effect relation where the suspicion is not stated.

nominal subject   nominal subject

A suspected drug-induced *hepatic disorder* was found and the *Singulair* was stopped.

conjunct

nominal subject   nominal subject

ACTOS 30 recovered HbA1c to 6.5% but an *edema* appeared after the medication.
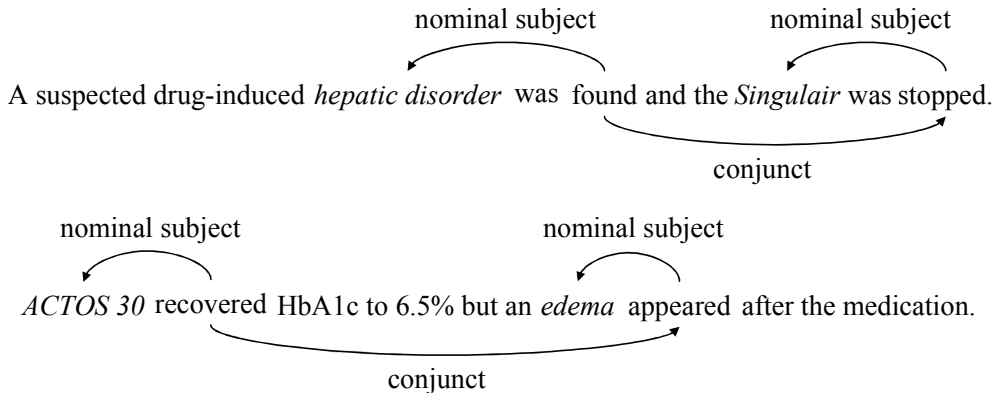
conjunct

Figure 3. The example of a similarity within dependency structures.

A salient characteristic of adverse–effect relations is that they are usually uncertain. The sentence in the example states that the hepatic disorder is *suspected drug-induced,* which means the hepatic disorder is likely to present an adverse–effect relation. Figure 2 presents an example in which an adverse–effect relation is suspected, but words to indicate the suspicion are not stated. The two effects of the drug—the recovery of HbA1c and the appearance of the edema—are expressed merely as observation results in this sentence. The recovery of HbA1c is an expected effect of the drug and the appearance of the edema probably represents an adverse–effect case. The uncertain nature of adverse–effect relations often engenders **the statement of an adverse–effect relation as an observed fact**. A sentence including an adverse–effect relation occasionally becomes long to list all observations that appeared after administration of a medication. Whether an interpretation that expresses an adverse–effect relation, such as *drug-induced* or *suspected to be an adverse–effect*, exists in a clinical record or not depends on a person who writes it. However, an adverse–effect relation is associated with an undesired effect of a medication. Its appearance would engender an extra action (e.g. *stopped in* the first example)

or lead to an extra indication (e.g. *but ... appeared in* the second example). Proper handling of this extra information is likely to boost the extraction accuracy.

The challenge of this study is to capture relations with various certainties. To establish this goal, we used a dependency structure for the adverse–effect relation extraction method. **Adverse–effect statements are assumed to share a dependency structure to a certain degree**. For example, if we obtain the dependency structures as shown in Figure 3, then we can easily determine that the structures are similar. Of course, obtaining such perfect parsing results is not always possible. A statistical syntactic parser is known to perform badly if a text to be parsed belongs to a domain which differs from a domain on which the parser is trained (Gildea, 2001). A statistical parser will likely output incomplete results in these texts and will likely have a negative effect on relation extraction methods which depend on it. The specified research topic of this study is to investigate whether incomplete dependency structures are effective and how they behave in the extraction of uncertain relations.

2 Related Works

Various studies have been done to extract semantic information from texts. SemEval-2007 Task:04 (Girju et al., 2007) is a task to extract semantic relations between nominals. The task includes "Cause–Effect" relation extraction, which shares some similarity with a task that will be presented herein. Saeger et al. (2008) presented a method to extract potential troubles or obstacles related to the use of a given object. This relation can be interpreted as a more general relation of the adverse–effect relation. The protein–protein interaction (PPI) annotation extraction task of BioCreative II (Krallinger et al., 2008) is a task to extract PPI from PubMed abstracts. BioNLP'09 Shared Task on Event Extraction (Kim et al., 2009) is a task to extract bio-molecular events (bio-events) from the GENIA event corpus.

Similar characteristics to those of the adverse–effect relation are described in previous reports in the bio-medical domain. Friedman et al. (1994) describes the certainty in findings of clinical radiology. Certainty is also known in scientific papers of biomedical domains as *speculation* (Light et al., 2004). Vincze et al. (2008) are producing a freely available corpus including annotations of uncertainty along with its scope.

Dependency structure feature which we utilized to extract adverse–effect relations are widely used in relation extraction tasks. We present previous works which used syntactic/dependency information as a feature of a statistical method. Beamer et al. (2007), Giuliano et al. (2007), and Hendrickx et al. (2007) all used syntactic information with machine learning techniques in SemEval-2007 Task:04 and achieved good performance. Riedel et al. (2009) used dependency path features with a statistical relational learning method in BioNLP'09 Shared Task on Event Extraction and achieved the best performance in the event enrichment subtask. Miyao et al. (2008) compared syntactic information of various statistical parsers on PPI.

3 Corpus

We produced an annotated corpus of adverse–effect relations to develop and test an adverse–effect relation extraction method. This section presents a description of details of the corpus.

3.1 Texts Comprising the Corpus

We used a discharge summary among various documents in a hospital as the source data of the task. The discharge summary is a document created by a doctor or another medical expert at the conclusion of a hospital stay. Medications performed during a stay are written in discharge summaries. If adverse–effect relations were observed during the stay, they are likely to be expressed in free text. Texts written in discharge summaries tend to be written more roughly than texts in newspaper articles or scientific papers. For example, the amounts of medications are often written in a name-value list as shown below:

"When admitted to the hospital, Artist 6 mg1x, Diovan 70 mg1x, Norvasac 5 mg1x and BP was 145/83, but after dialysis, BP showed a decreasing tendency and in 5/14 Norvasac was reduced to 2.5 mg1x."

3.2 Why Adverse–Effect Relation Extraction from Discharge Summaries is Important

In many countries, adverse–effects are investigated through multiple phases of clinical trials, but unexpected adverse–effects occur in actual medications. One reason why this occurs is that drugs are often used in combination with others in actual medications. Clinical trials usually target single drug use. For that reason, the combinatory uses of drugs occasionally engender unknown effects. This situation naturally motivates automatic adverse–effect relation extraction from actual patient records.
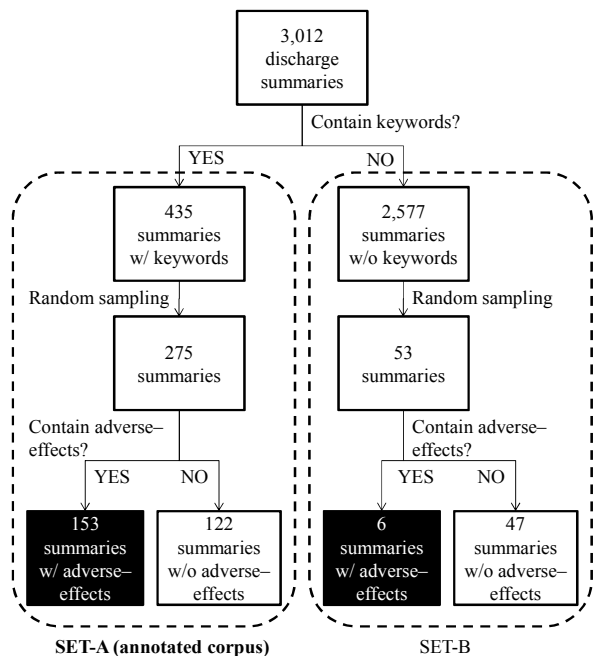
Figure 4. The overview of the summary selection.

3.3 Corpus Size

We collected 3,012 discharge summaries[1] written in Japanese from all departments of a hospital. To reduce a cost to survey the occurrence of adverse–effects in the summaries, we first split the summaries into two sets: SET-A, which contains keywords related to adverse–effects and SET-B, which do not contain the keywords. The keywords we used were "stop, change, adverse effect", and they were chosen based on a heuristic. The keyword filtering resulted to SET-A with 435 summaries and SET-B with 2,577 summaries. Regarding SET-A, we randomly sampled 275 summaries and four annotators annotated adverse–effect information to these summaries to create the adverse–effect relation corpus. For SET-B, the four annotators checked the small portion of the summaries. Cases of ambiguity were resolved through discussion, and even suspicious adverse–effect relations were annotated in the corpus as positive data. The overview of the summary selection is presented in Figure 4.

---

[1] All private information was removed from them. The definition of private information was referred from the HIPAA guidelines.

Table 1. Markup scheme.

tag	Definition and Examples
drug	The expression of an administrated drug: e.g. *Levofloxacin, Flexeril*.
symptom	The expression of a disease or symptom: e.g. *endometrial cancer, headache*. This tag covers not only a noun phrase but also a verb phrase such as "<symptom>*feels a pain in front of the head*</symptom>".

Table 2. Annotation examples.

<drug relation="1">*Ridora*</drug> resumed because it is associated with an <symptom relation="1">*eczematous rash*</symptom>.
<drug relation="1">ACTOS(30)</drug> brought both <symptom relation="1">*headache*<symptom> and <symptom relation="1">*insomnia*</symptom>.

* If a drug has two or more adverse-effects, symptoms take a same relation ID.

3.4 Quantities of Adverse–Effects in Clinical Texts

55.6% (=158/275) of the summaries in SET-A contained adverse–effects. 11.3% (=6/53) of the summaries in SET-B contained adverse–effects. Since the ratio of SET-A:SET-B is 14.4:85.6, we estimated that about 17.7% (=0.556×0.144+0.113×0.856) of the summaries contain adverse–effects. Even considering that a summary may only include suspected adverse–effects, we think that discharge summaries are a valuable resource to explore adverse–effects.

3.5 Annotated Information

We annotated information of two kinds to the corpus: term information and relation information.

(1) Term Annotation

Term annotation includes two tags: a tag to express a drug and a tag to express a drug effect. Table 1 presents the definition. In the corpus, 2,739 drugs and 12,391 effects were annotated.

(2) Relation Annotation

Adverse–effect relations are annotated as the "relation" attribute of the term tags. We represent the effect of a drug as a relation between a drug tag and a symptom tag. Table 2 presents

78

Table 3. Features used in adverse-effect extraction.

ID	Feature	Definition and Examples
1	Character Distance	The number of characters between members of a pair.
2	Morpheme Distance	The number of morpheme between members of a pair.
3	Pair Order	Order in which a drug and a symptom appear in a text; "drug–symptom" or "symptom–drug".
4	Symptom Type	The type of symptom: "disease name", "medical test name", or "medical test value".
5	Morpheme Chain	Base–forms of morphemes that appear between a pair.
6	Dependency Chain	Base–forms of morphemes included in the minimal dependency path of a pair.
7	Case Frame Chain	Verb, case frame, and object triples that appear between a pair: e.g. "examine" –"*de*"(case particle) – "inhalation", "begin" –"*wo*"(case particle) –"medication".
8	Case Frame Dependency Chain	Verb, case frame, and object triples included in the minimal dependency path of a pair.

Figure 5. Pair extraction example.

Figure 6. Dependency chain example.

several examples, wherein "relation=1" denotes the ID of a adverse–effect relation. In the corpus, 236 relations were annotated.

4 Extraction Method

We present a simple adverse–effect relation extraction method. We extract drug–symptom pairs from the corpus and discriminate them using a machine-learning technique. Features based on morphological analysis and dependency analysis are used in discrimination. This approach is similar to the PPI extraction approach of Miyao et al. (2008), in which we binary classify pairs whether they are in ad-

verse–effect relations or not. A pattern-based semi-supervised approach like Saeger et al. (2008), or more generally Espresso (Pantel and Pennacchiotti, 2006), can also be taken, but we chose a pair classification approach to avoid the effect of seed patterns. To capture a view of an adverseness of a drug, a statistic of adverse–effect relations is important. We do not want to favor certain patterns and chose a pair classification approach to equally treat every relation. Extraction steps of our method are as presented below.

STEP 1: Pair Extraction
All combinations of drug–symptom pairs that appear in a same sentence are extracted. Pairs

Table 4. Best F_1-scores and their parameters.

ID	Feature Combination	Parameters	Precision	Recall	F_1-score
A	1,2,3,4,5	log(c)=3.0, log(g)=-5.0, p=0.10	26.72	46.21	33.05
B	1,2,3,4,5,6	log(c)=1.0, log(g)=-5.0, p=0.10	33.30	42.43	36.64
C	**1,2,3,4,5,6,7**	**log(c)=1.0, log(g)=-5.0, p=0.10**	**34.39**	**43.06**	**37.54**
D	1,2,3,4,5,6,8	log(c)=1.0, log(g)=-5.0, p=0.10	35.01	40.67	36.78
E	1,2,3,4,5,6,7,8	log(c)=1.0, log(g)=-5.0, p=0.10	35.45	41.05	37.18
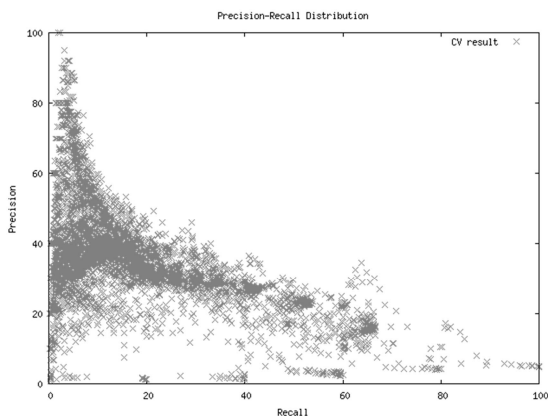
Figure 7. Precision–recall distribution.

with the same relation ID become positive samples; pairs with different relation IDs become negative samples. Figure 5 shows examples of positive and negative samples.

STEP 2: Feature Extraction

Features presented in Table 3 are extracted. The text in the corpus is in Japanese. Some features assume widely known characteristics of Japanese. For example, the dependency feature allows a phrase to depend on only one phrase that appears after a dependent phrase. Figure 6 portrays an example of a dependency chain feature. In the example, most terms were translated into English, excluding postpositions (PP) and auxiliaries (AUX), which are expressed in italic. To reduce the negative effect of feature sparsity, features which appeared in more than three summaries are used for features with respective IDs 5–8.

STEP 3: Machine Learning

The support vector machine (SVM) (Vapnik, 1995) is trained using positive/negative labels and features extracted in prior steps. In testing, an unlabeled pair is given a positive or negative label with the trained SVM.

5 Experiment

We performed two experiments to evaluate the extraction method.

5.1 Experiment 1

Experiment 1 aimed to observe the effects of the presented features. Five combinations of the features were evaluated with a five-fold cross validation assuming that an optimal parameter combination was obtained. The experiment conditions are described below:

A. Data

7,690 drug–symptom pairs were extracted from the corpus. Manually annotated information was used to identify drugs and symptoms. Within 7,690 pairs, 149 pairs failed to extract the dependency chain feature. We removed these 149 pairs and used the remaining 7,541 pairs in the experiment. The 7,541 pairs consisted of 367 positive samples and 7,174 negative samples.

B. Feature Combinations

We tested the five combinations of features in the experiment. Manually annotated information was used for the symptom type feature. Features related to morphemes are obtained by processing sentences with a Japanese morphology analyzer (JUMAN[2] ver. 6.0). Features related to dependency and case are obtained by processing sentences using a Japanese dependency parser (KNP ver. 3.0; Kurohashi and Nagao, 1994).

C. Evaluations

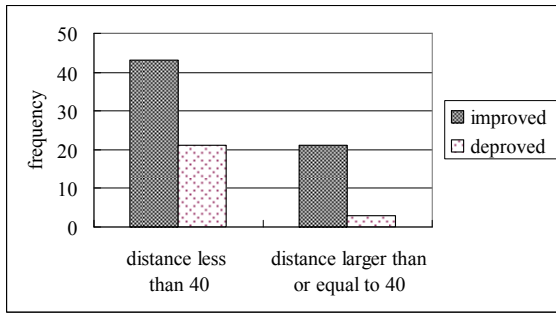We evaluated the extraction method with all combinations of SVM parameters in certain

---

[2] http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/juman-e.html

Figure 8. Relation between the number of pairs and the morpheme distance.

Figure 9. Number of dependency errors in the improved pairs sentences.

ranges. We used LIBSVM[3] ver. 2.89 as an implementation of SVM. The radial basis function (RBF) was used as the kernel function of SVM. The probability estimates option of LIBSVM was used to obtain the confidence value of discrimination**.**

The *gamma* parameter of the RBF kernel was chosen from the range of $[2^{-20}, 2^0]$. The *C* parameter of SVM was chosen from the range of $[2^{-10}, 2^{10}]$. The SVM was trained and tested on 441 combinations of *gamma* and *C*. In testing, the probability threshold parameter *p* between $[0.05, 0.95]$ was also chosen, and the F_1-scores of all combination of *gamma*, *C*, and *p* were calculated with five-fold cross validation. The best F_1-scores and their parameter values for each combination of features (optimal F_1-scores in this setting) are portrayed in Table 4. The precision–recall distribution of F_1-scores with feature combination C is presented in Figure 7.

5.2 Experiment 2

Experiment 2 aimed to observe the performance of our extraction method when SVM parameters were automatically tuned. In this experiment, we performed two cross validations: a cross validation to tune SVM parameters and another cross validation to evaluate the extraction method. The experiment conditions are described below:

A. Data
The same data as Experiment 1 were used.
B. Feature Combination
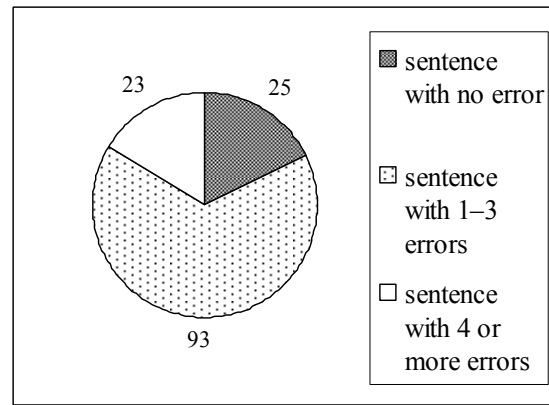Feature combination C, which performed best in Experiment 1, was used.
C. Evaluation

Two five-fold cross validations were performed. The first cross validation divided the data to 5 sets (A, B, C, D, and E) each consisting of development set and test set with the ratio of 4:1. The second cross validation train and test all combination of SVM parameters (*C*, *gamma*, and *p*) in certain ranges and decide the optimal parameter combination(s) for the development sets of A, B, C, D, and E. The second cross validation denotes the execution of Experiment 1 for each development set. For each optimal parameter combination of A, B, C, D, and E, the corresponding development set was trained and the trained model was tested on the corresponding test set. The average F_1-score on five test sets marked 34.90, which is 2.64 lower than the F_1-score of Experiment 1 with the same feature combination.

6 Discussion

The result of the experiment reveals the effectiveness of the dependency chain feature and the case-frame chain feature. This section presents a description of the effects of several features in detail. The section also mentions remaining problems in our extraction method.

6.1 Effects of the Dependency Chain Feature and Case-frame Features

A. Dependency Chain Feature
The dependency chain features improved the F_1-score by 3.59 (the F_1-score difference between feature combination A and B). This increase was obtained using 260 improved pairs and 127 deproved pairs. Improved pairs con-

tribute to the increase of a F_1-score. Deproved pairs have the opposite effect.

We observed that improved pairs tend to have longer morpheme distance compared to deproved pairs. Figure 8 shows the relation between the number of pairs and the morpheme distance of improved pairs and deproved pairs. The ratio between the improved pairs and the deproved pairs is 11:1 when the distance is greater than 40. In contrast, the ratio is 2:1 when the distance is smaller than 40. This observation suggests that adverse–effect relations share dependency structures to a certain degree.

We also observed that in improved pairs, dependency errors tended to be low. Figure 9 presents the manually counted number of dependency errors in the 141 sentences in which the 260 improved pairs exist: 65.96 % of the sentences included 1–3 errors. The result suggests that the dependency structure is effective even if it includes small errors.

B. Case-frame Features

The effect of the case-frame dependency chain feature differed with the effect of the dependency chain feature. The case-frame chain feature improved the F_1-score by 0.90 (the F_1-score difference between feature combination B and C), but the case-frame dependency chain feature decreased the F_1-score by 0.36 (the F_1-score difference between feature combination C and E). One reason for the negative effect of the case-frame dependency feature might be feature sparsity, but no clear evidence of it has been found.

6.2 Remaining Problems

A. Imbalanced Data

The adverse–effect relation pairs we used in the experiment were not balanced. Low values of optimal probability threshold parameter p suggest the degree of imbalance. We are considering introduction of some kind of methodology to reduce negative samples or to use a machine learning method that can accommodate imbalanced data well.

B. Use of Medical Resources

The extraction method we propose uses no medical resources. Girju et al. (2007) indicate the effect of WordNet senses in the classification of a semantic relation between nominals. Krallinger et al. (2008) report that top scoring teams in the interaction pair subtask used sophisticated interactor protein normalization strategies. If medical terms in texts can be mapped to a medical terminology or ontology, it would likely improve the extraction accuracy.

C. Fully Automated Extraction

In the experiments, we used the manually annotated information to extract pairs and features. This setting is, of course, not real if we consider a situation to extract adverse–effect relations from massive clinical records, but we chose it to focus on the relation extraction problem. We performed an event recognition experiment (Aramaki et al., 2009) and achieved F_1-score of about 80. We assume that drug expressions and symptom expressions to be automatically recognized in a similar accuracy.

We are planning to perform a fully automated adverse–effect relations extraction from a larger set of clinical texts to see the performance of our method on a raw corpus. The extraction F_1-score will likely to decrease, but we intend to observe the other aspect of the extraction, like the overall tendency of extracted relations.

7 Conclusion

We presented a method to extract adverse–effect relations from texts. One important characteristic of adverse–effect relations is that they are uncertain in most cases. We performed experiments to extract adverse–effect relations from 2,577 clinical texts, and obtained F_1-score of 37.54 with optimal SVM parameters and F_1-score of 34.90 with automatically tuned SVM parameters. Results also show that dependency features increase the extraction F_1-score by 3.59. We observed that an increased F_1-score was obtained using the improvement of adverse–effects with long morpheme distance, which suggests that adverse–effect relations share dependency structures to a certain degree. We also observed that the increase of the F_1-score was obtained with dependency structures that include small errors, which suggests that the dependency structure is effective even if it includes small errors.

References

Aramaki, Eiji, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, and Kazuhiko Ohe. 2009. TEXT2TABLE: Medical Text Summarization System Based on Named Entity Recognition and Modality Identification. In *Proceedings of the BioNLP 2009 Workshop*, pages 185-192.

Beamer, Brandon, Suma Bhat, Brant Chee, Andrew Fister, Alla Rozovskaya, and Roxana Girju. 2007. UIUC: A Knowledge-rich Approach to Identifying Semantic Relations between Nominals. In *Proceedings of Fourth International Workshop on Semantic Evaluations*, pages 386-389.

Friedman, Carol, Philip O. Alderson, John H. M. Austin, James J. Cimino, and Stephen B. Johnson. 1994. A General Natural-language Text Processor for Clinical Radiology. Journal of the American Medical Informatics Association, 1(2), pages 161-174.

Gildea, Daniel. 2001. Corpus Variation and Parser Performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 1-9.

Girju, Roxana, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 task 04: Classification of Semantic Relations between Nominals. In *Proceedings of Fourth International Workshop on Semantic Evaluations*, pages 13-18.

Giuliano, Claudio, Alberto Lavelli, Daniele Pighin, and Lorenza Romano. 2007. FBK-IRST: Kernel Methods for Semantic Relation Extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 141-144.

Hendrickx , Iris, Roser Morante, Caroline Sporleder, and Antal van den Bosch. 2007. ILK: Machine learning of semantic relations with shallow features and almost no data. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 187-190.

Kim, Jin-Dong, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1-9.

Krallinger, Martin, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biology 2008, 9(Suppl 2):S4.

Kurohashi, Sadao and Makoto Nagao. 1994. KN Parser : Japanese Dependency/Case Structure Analyzer. In *Proceedings of The International Workshop on Sharable Natural Language Resources*, pages 22-28. Software available at http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/knp-e.html.

Light, Marc, Xin Ying Qiu, and Padmini Srinivasan. 2004. The Language of Bioscience: Facts, Speculations, and Statements in Between. In *Proceedings of HLT/NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 17-24.

Miyao, Yusuke, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented Evaluation of Syntactic Parsers and Their Representations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 46-54.

Pantel, Patrick and Marco Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113-120.

Riedel, Sebastian, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A Markov Logic Approach to Bio-Molecular Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 41-49.

Saeger, Stijn De, Kentaro Torisawa, and Jun'ichi Kazama. 2008. Looking for Trouble. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 185-192.

Vapnik, Vladimir N.. 1995. The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc..

Vincze, Veronika, György Szarvas, Richárd Farkas, György Móra, and János Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. BMC Bioinformatics 2008, 9(Suppl 11):S9.