Improving Relationship Extraction from Clinical Notes by Sentence Classification

Ehsan Emadzadeh
Arizona State University
Department of Biomedical Informatics, Arizona State University, Tempe, AZ, USA
eemadzad@asu.edu

Azadeh Nikfarjam
Arizona State University
Department of Biomedical Informatics, Arizona State University, Tempe, AZ, USA
anikfarj@asu.edu

Graciela Gonzalez
Arizona State University
Department of Biomedical Informatics, Arizona State University, Tempe, AZ, USA
graciela.gonzalez@asu.edu

ABSTRACT

In recent years, there has been an increasing interest in automating knowledge extraction from biomedical text, including both clinical notes and published literature. Extracting relationships between concepts in a sentence is an essential step in many knowledge extraction tasks. Identifying whether a sentence contains any relationship between its concepts, can potentially improve all relationship extraction methods. Here we seek to evaluate the effectiveness of binary sentence classification on relationship extraction from clinical notes, as well as the effect of different classes of features on the same task. We use 2010 i2b2/VA shared task clinical notes corpus as the gold standard for evaluation. The sentence binary classification achieves 90.14% f-measure (91.42% precision and 88.9% recall), improving the relationship extraction by 2.19% f-measure.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing - Language parsing and understanding, Text analysis

I.2.6 [Artificial Intelligence]: Learning - Knowledge acquisition

I.5.4 [Pattern Recognition]: Applications – text processing.

I.5.2 [Pattern Recognition]: Design Methodology - Classifier design and evaluation, Feature evaluation and selection

General Terms

Algorithms, Performance

Keywords

Relationship extraction, clinical notes, NLP, text mining

KDD-DMH'13, August 11, 2013, Chicago, Illinois, USA.

1. INTRODUCTION

In recent years, there has been an increasing interest in automating knowledge extraction from biomedical text, including both clinical notes and published literature. Knowledge extraction methods can be used to create ontologies[1], advance question answering systems[2], support biomedical hypothesis generation and many other intelligent applications that can impact healthcare quality. The text fields and clinical notes inside electronic health records include valuable knowledge that is not reflected in the structured data. In order to automatically understand free text, the first step is to identify the concepts of interest in the text. The second, and more difficult, step is finding the relationships between the discovered concepts. There are different approaches to find relationships between two concepts. The best performing methods are based on machine-learning techniques[3], [4]. Nguyen et al.[5] proposed an approach using sub-tree mining. Bunescu et al.[6] used a matching technique on dependency paths for extracting relationships between biological entities.

The aim of this paper is to investigate whether binary sentence classification before the main relationship classification, such as the mentioned methods, improves the results of relationship extraction. Yamamoto et al. used sentence classification for summarization[7]. He et al.[8] used sentence classification to find if a sentence contains an event description or not. To the best of our knowledge, there is no study on the effect of binary sentence classification on relationship extraction from clinical notes. Furthermore we evaluate the effectiveness of different features for relationship extraction from clinical notes.

2. METHODS

In this section, we explain the model for sentence classification as part of the relationship extraction task, and then we explore the effectiveness of different features. We use the 2010 i2b2/VA shared task clinical notes corpus for the experiments. The corpus includes a total number of 871 health records, 394 for training and 477 in the test set. It is annotated for concepts, assertions and relationships. More details about the corpus can be found in Uzuner et al.[9]. In this research, we focus on evaluation of a machine-learning-based system without any hard coded rules. Our goal is to find the effect of sentence classification on a pure machine learning system without manually crafted rules. Table 1 shows different relation types defined in the corpus and their number of testing and training examples. A total number of 8 relation types exist in the dataset. The number of annotation examples for some relation types are extremely small to be used in

a machine learning method, and they can be handled more effectively by using predefined rules. Therefore, we exclude relation types with a small number of examples from our analysis.

2.1 Relationship extraction

The relationship extraction problem can be modeled as a multi-class, single-label classification of a link between two concepts (words or phrases) in a sentence. In order to train/test a classifier, training/testing examples should be prepared and fed into the classifier. The selected features incorporated in classification examples can significantly affect the final performance of the system. Instead of generating examples for all possible links between concepts in a sentence, we only included valid links based on prior knowledge and defined relation types in the corpus. For example, the link between a "Problem" and a "Treatment" is a valid link. However, we do not consider any relation between two concepts tagged as "Test". If we had included all possible links, the training set size would have increased inefficiently. In addition, since most of the generated links are not valid relations, the accuracy of the classifier would have been reduced dramatically. Figure 1 shows the steps for training and testing. We use SVM^{multiclass}[10] that is an implementation of SVM[11]. A detailed explanation of the sentence classification and relation extraction process is presented in the following sections.

Figure 1. This figure shows the stages of the proposed pipeline. It is divided into three sections: pre-processing, sentence classification and relationship classification.

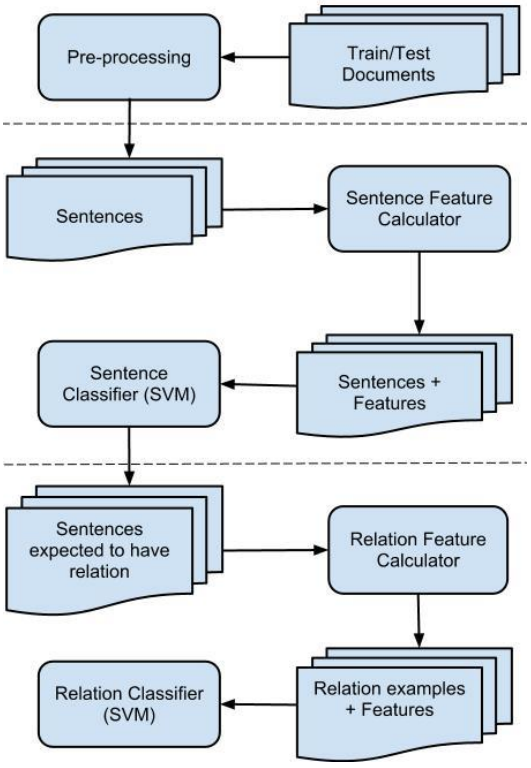

Table 1. Clinical notes corpus. This table shows number of the different relation types in the train-set and test-set. "TrWP", "TrIP" and "TrNAP" don't have enough training examples and will be excluded for further evaluations.

Relation	Train-set	Test-set
TeRP (A test has revealed some medical problem)	1733	3032
TrAP (A treatment administered for a medical problem)	1422	2487
PIP (Two problems are related to each other)	1239	1986
TeCP (A test was performed to investigate a medical problem)	303	588
TrCP (A treatment caused a medical problem)	296	444
TrIP (A certain treatment has improved or cured a medical problem)	107	198
TrNAP (The administration of a treatment was avoided because of a medical problem)	106	191
TrWP (A patient's medical problem has deteriorated or worsened because of or in spite of a treatment being administered)	56	143
Overall	5262	9069

2.2 Pre-processing and feature generation

To build and evaluate SVM models, the first step is to convert the train and test documents into SVM expected input format. All documents were tokenized and converted to database records for faster conversions to the SVM^{multiclass}[10] expected format. For the sentence classifier, a set of features are calculated for each sentence (listed in Table 2). For the relationship classifier, a different set of features is used that are listed in Table 4. After importing all documents and adding calculated features into the database, the next step is to create SVM examples which are explained in the following sections.

2.3 Sentence classification

The sentence classifier is used for filtering out sentences that do not have any relation. Each sentence in the training set is assigned to positive (has relation) or negative (does not have any relation) , using an SVM classifier with linear kernel. The list of features that are used for sentence classification is listed in Table 2. For training the model, we selected negative examples from the sentences that had enough concepts (e.g at least 2 "Problems"). Limiting the scope of negative examples improved the sentence classification significantly.

2.3.1 Sentence classification features

Table 2 lists all of the features used for sentence classification. The first feature is the position (offset) of the sentence in the document. The numbers of different concepts such treatments, problems or test are distinguishing features to predict whether a sentence has a relation or not. Features 2-10 are the features

related to concept counts in a sentence. Feature 11-13 describe the context of the concepts in the sentence. In clinical notes previous words before each concept seemed to be a good indicator of relationship in the sentence. Feature 14 consists of tf-idf (Term frequency- inverse document frequency)[12] weights of each term that appears in the sentence. TF-IDF feature represents all words in the sentence.

2.4 Relationship classification

For classifying the relationship type, one multi-class SVM model was trained using $SVM^{multiclass}$. Only candidate relations inside the sentences that were predicted to include a relation were used for training and testing. In the next section, we will explain how we define features of a relation. These features represent a relation at various levels (word, sentence, and document) and are used for classifier example generation. Imbalance of positive and negative examples causes performance problems in SVM. To overcome this, for each relation type we kept the proportion of positive to negative examples equal to a constant value (α), and adjusted α to get the best performance according to each relation type. Negative examples were selected randomly; however, this could be improved by selecting negative examples from different relation types.

For post-processing, examples are divided into 3 different categories: 1.Problem-Problem (PIP), 2.Problem-Test (TeRP, TeCP) and 3.Problem-Treatment (TrAP, TrNAP, TrCP, TrIP, TrWP). For each category if SVM predicted a test example to be of a class other than the possible classes in that category, then it was considered as a false positive prediction. In the case of a categorical false positive we assigned the class with highest prevalence inside the category, calculated from the training set. For example, a link between a "Problem" and a "Test" can be either TeRP or TeCP. If the classifier selects a class other than TeRP or TeCP, then the link would be assigned as TeRP (TeRP has higher prevalence than TeCP). The testing process is similar to the training which consists of two steps: (1) Using sentence classifier to find whether the sentence has at least a relation, and (2) Predicting the class of each valid link in the selected sentences. Testing examples were created for each valid link between the concepts in a sentence.

Table 2. Set of 14 features used for sentence classification.

1.	Position in document (offset)
2.	Number of problems in the sentence
3.	Number of problems, with "Present" assertion
4.	Number of problems, with "Absent" assertion
5.	Number of problems with "Associated With Someone Else" assertion
6.	Number of problems with "Conditional" assertion
7.	Number of problems with "Hypothetical" assertion
8.	Number of problems with "Possible" assertion
9.	Number of tests in the sentence
10.	Number of treatments in the sentence
11.	Tokens in the sentence appearing before treatments
12.	Tokens in the sentence appearing before tests
13.	Tokens in the sentence appearing before problems
14.	The sentence words TF-IDFs

2.4.1 Relationship classification features

Table 4 shows the list of all features used for relationship extraction. Most of them are traditional features and the calculations are straightforward. Therefore here we only discuss the new and complex features.

The semantic feature is the semantic similarity of the two concepts in an association. We use the Semantic Vectors[13] package for empirically calculating semantic similarity of the two concepts. Semantic Vectors provide a fast and approximate implementation of the Latent Semantic Analysis (LSA)[14] algorithm. The vector for a concept is the weighted sum of the vectors of the documents that contain the concept. We use the training documents as the corpus for creating semantic vectors. Semantic features are calculated by getting the semantic similarity of two participating entities in a potential association. For calculating semantic similarity, common words in two concepts were removed. Then we removed stop-words and non-alphanumeric characters. The value of semantic similarity is the average of semantic similarities of the phrases containing the two concepts.

We also added a feature based on NELL, which was not used in previous publications for relationship extraction. NELL (Never-Ending Language Learner)[15] is a system that extracts structured information from 500 million unstructured web pages. The system is based on a predefined ontology of concepts and relations, and it tries to learn new categories and relations continuously using semi-supervised learning. The "NELL feature" is a Boolean feature showing if there is a path between two concepts in an association in the NELL network.

Most of the classification algorithms cannot handle attributes with string values; therefore, the common approach is to convert an attribute with string values to multiple attributes by adding one attribute for each possible string value. In this setting, we consider a class of features as the general attribute (e.g. "POS"), before splitting it to multiple attributes (e.g POS=Verb, Noun). The feature evaluation is performed on a class of features, in which all value-features in a feature class are excluded or included together. For instance, we tried to evaluate if "Part of Speech" helps the classifier by including/excluding all possible values of part of speech, and we do not evaluate each possible value for part of speech (e.g. "POS_Verb", "POS_Noun").

3. RESULTS AND DISCUSSION

In this study, we examined the impact of sentence classifier and explored the effectiveness of various features in improving the relationship extraction from clinical notes, using the proposed machine learning based approach. Support vector machine is used for classifying each possible link between concepts in a sentence. The result has two main parts: 1. Sentence classifier evaluation, 2. Features evaluation. The results, as shown in Table 3, indicate the effectiveness of the sentence classification for relationship extraction. Table 4 presents the effect of the different features on the average f-measure. The next two sections discuss each part of the result in detail.

3.1 Sentence Classifier Evaluation

The sentence classifier is a gate classifier to filter out sentences without any relation. Table 3 shows the result after and before using the sentence classifier. For all the relation types, the recall increased and the precision decreased after using the sentence classifier, resulting on an increased f-measure. TrCP shows the

highest improvement for f-measure by 7.83%. The average increase in f-measure is 2.19%.

The proposed sentence classifier can be easily combined with any relationship extraction method. This study shows that filtering the sentences before the relationships classifier improves the overall performance. Using sentence classifier in any similar relationship extraction tasks is expected to improve the results. Evaluating the effect of the sentence classifier for relationship extraction in other corpora is a part of our future work.

3.2 Feature evaluation

Table 4 shows the detail evaluation of each feature. In this table, each feature was removed at a time, and the model was evaluated. A decrease in f-measure after the removal of a feature shows that the feature is beneficial. The "named entity" features are the most important ones. Interestingly, the named entity for the second concept in a relationship is more valuable than the first one. This can be due to the fact that most of the first concepts have the same named entity. The next key feature is "tokens in between" causing 1% increase on the result. The "Semantic similarity" feature has 0.11% effect on the f-measure. We expect to have a better estimation of semantic similarity by using a larger corpus for creating the semantic vectors. In the future, we are interested in comparing LSA with other semantic similarity kernels.

4. CONCLUSION

We focused on the relationship extraction task and showed that applying a sentence classifier can improve the extraction results. With little adaptation, the proposed sentence classifier can be used in other relationship extraction systems. Evaluating the effect of sentence classification on different relationship extraction tasks is part of our future work. In addition, this study has shown the effect of different features on the relationship extraction. For example we found using semantic similarity of two concepts can also improve the relationship extraction result, although its impact is more modest.

One of the features we didn't included in our experiment is the order of named entities that can be useful for sentence classification. For example appearance of a "test" before a "problem" can increase the chance of having TeRP/TeCP relationship in a sentence, while a "problem" before a "test" less likely implies TeRP/TeCP. We are thrilled to include more ordinal features in the sentence classifier and evaluate the change. In addition evaluating more computationally expensive features, like bi-grams, is part of our future works. Even though improving sentence classification with more complex features might subsequently improve the relationship extraction, but the significance of the gain considering the computational complexity is questionable.

5. REFERENCES

[1] C. Pesquita, D. Faria, A. O. Falcão, P. Lord, and F. M. Couto, "Semantic similarity in biomedical ontologies.," *PLoS computational biology*, vol. 5, no. 7, p. e1000443, Jul. 2009.

[2] Y. Cao, F. Liu, P. Simpson, L. Antieau, A. Bennett, J. J. Cimino, J. Ely, and H. Yu, "AskHERMES: An online question answering system for complex clinical questions.," *Journal of biomedical informatics*, vol. 44, no. 2, pp. 277–88, Apr. 2011.

[3] B. De Bruijn and C. Cherry, "Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010," *Journal of the American …*, 2011.

[4] B. Rink, S. Harabagiu, and K. Roberts, "Automatic extraction of relations between medical concepts in clinical texts," *… of the American Medical …*, 2011.

[5] D. Nguyen, Y. Matsuo, and M. Ishizuka, "Relation extraction from wikipedia using subtree mining," *… of the National Conference on Artificial …*, pp. 1414–1420, 2007.

[6] R. Bunescu and R. Mooney, *Extracting Relations from Text: From Word Sequences to Dependency Paths*. Springer London, 2007, pp. 29–44.

[7] Y. Yamamoto and T. Takagi, "A sentence classification system for multi biomedical literature summarization," *Data Engineering Workshops, 2005. …*, 2005.

[8] Q. He, K. Chang, and E.-P. Lim, "Anticipatory Event Detection via Sentence Classification," *2006 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1143–1148, Oct. 2006.

[9] B. R. South, S. Shen, S. L. Duvall, and O. Uzuner, "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text.," *Journal of the American Medical Informatics Association : JAMIA*, pp. 552–557, Jun. 2011.

[10] T. Joachims, "Making large scale SVM learning practical," *Advances in Kernel Methods - Support Vector Learnin*, no. B. Schölkopf and C. Burges and A. Smola (ed.), 1999.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, 1995.

[12] K. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 60, no. 5, pp. 493–502, 1972.

[13] D. Widdows and T. Cohen, "The Semantic Vectors Package: New Algorithms and Public Tools for Distributional Semantics," in *2010 IEEE Fourth International Conference on Semantic Computing*, 2010, pp. 9–15.

[14] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[15] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr, and T. M. Mitchell, "Toward an Architecture for Never-Ending Language Learning."

Table 3. Sentence Classifier Evaluation: This table shows using the sentence classifier improves the baseline micro-average f-measure by 2.19%. Adding the sentence classifier improved recall and decreased the precision for all the relation types. The numbers are Precision/Recall/F-Measure.

Relation	**Without sentence classifier**	**With sentence classifier**	**Change**
TeRP	82.14/85.42/83.75	78.88/90.04/84.09	-3.27/4.62/**0.34**
TrAP	67.87/70.46/69.14	64.18/78.03/70.43	-3.68/7.57/**1.29**
PIP	76.09/52.95/62.44	72.13/64.75/68.24	-3.95/11.80/**5.80**
TeCP	78.04/28.40/41.65	68.60/34.18/45.63	-9.44/5.78/**3.98**
TrCP	65.61/32.73/43.67	61.18/44.47/51.50	-4.43/11.74/**7.83**
Micro-average	75.27/63.54/68.91	71.27/70.93/71.10	-4.00/7.38/**2.19**

Table 4. This table shows the micro-average f-measures change when each features is removed. The numbers are F-Measure. For a link between two concepts, Concept1 is the first concept (lower offset) and Concept2 is the second concept appearing in the sentence.

Feature removed	**f-measure**	**Change after removing**
With all features	71.12	-
Concept1 name entity (Problem, Treatment or Test)	70.22	-0.9
Concept2 name entity (Problem, Treatment or Test)	69.79	**-1.33**
Tokens in between of concepts	70.12	-1
Concepts distance (Number of tokens between two concepts)	70.97	-0.15
Semantic similarity	71.01	-0.11
POSs in between of concepts	70.96	-0.16
Nell link (shows if Nell link exists between two concepts)	71.08	-0.04
Edge type (entity types in the link, e.g. Problem-Test)	71.09	-0.03
Concept1 tokens in window next	70.32	-0.8
Concept1 tokens in window before	70.49	-0.63
Concept1 directly linked tokens in dependency graph	70.88	-0.24
Concept2 tokens in window next	70.62	-0.5
Concept2 tokens in window before	70.6	-0.52
Concept2 directly linked tokens in dependency graph	70.99	-0.13
Tokens in the path between two concepts in the parse tree	70.86	-0.26