# DUTIR at the BioCreative V CDR Task: Disease Named Entity Recognition and Normalization and the Chemical-Disease Relation Extraction from Biomedical Text

Zhiheng Li[1],YaYang[1], Zhihao Yang*[1], Ziwei Zhou[1],Hongfei Lin[1]

[1] College of Computer Science and Technology, Dalian University of Technology, Dalian, China 116024
zhihengli@mail.dlut.edu.cn; yyang@mail.dlut.edu.cn; yangzh@dlut.edu.cn; ziweizhou@mail.dlut.edu.cn;hflin@dlut.edu.cn

**Abstract.** Adverse drug reactions between chemicals and diseases make the topic of chemical-disease relations (CDR) become a focus that receives much concern. In this paper, we introduce our methods used to create our submissions to the BioCreative V CDR subtask, i.e. Disease Named Entity Recognition and Normalization (DNER) and Chemical-Induced Diseases (CID). In our DNER method, firstly, a CRF model with a dictionary is used to recognize disease mentions. Secondly, the dictionary look-up that combines the exact and approximate matching is employed to map disease mentions to disease identifiers. Finally, disambiguation is implemented by choosing a unique disease identifier for an ambiguous disease mention using extended semantic information. Experimental results show that our approach achieves an F-score of 64.46% on the test set of CDR DNER task. Our CID method combines the feature-based kernel and graph kernel. A semi-supervised learning method, Co-Training, is introduced which makes use of the unlabeled data to boost the performance of a classifier. Finally, we use the obtained model to extract the CID relations at the sentence level, and then use some rules to obtain the final results at the abstract level. Our system achieved an F-score of 52% on the development set, and an F-score of 35.52% on the test set of the CID subtask, respectively.

**Keywords:** Disease named entity recognition; Chemical-induced diseases relation extraction; CRF; Co-training; Full Name-Abbreviation Pairs

## 1 Introduction

In recent years, many systems have been developed for the automatic extraction of biomedical events from text, such as protein-protein interactions and gene-disease relations [1-3]. However, relatively few studies addressed the extraction of information about potential adverse drug reactions hidden in the text of the medical case reports [4], which is important for improving chemical safety and toxicity studies and facilitating new screening assays for pharmaceutical compound survival. Therefore, automatic extraction of chemical-induced diseases relation information from biomedical literature has become an important research area.

BioCreative V proposes a challenge task of automatic extraction of mechanistic and biomarker chemical-disease relations (CDR) from the biomedical literature in support of biocuration, new drug discovery and drug safety surveillance [5,6]. The task is aimed to advance text-mining research on relationship extraction and provide practical benefits to biocuration. The first subtask is Disease Named Entity Recognition and Normalization (DNER), an intermediate step for automatic CDR extraction, which was found to be highly difficult on its own [7] in previous BioCreative CTD tasks [8,9]. For the subtask, participating systems will be given PubMed abstract and asked to return normalized disease concept identifiers. The second subtask is chemical-induced diseases relation extraction (CID). Participating systems will be provided with raw text of PubMed articles as input and asked to return a ranked list of pairs with normalized confidence scores for which drug-induced diseases are asserted in the abstract. We participated in both subtasks and our methods and results are presented in the following sections.

## 2 Discussion

### 2.1 DNER subtask

In this task, we present an approach integrating various resources for disease name normalization. The pipeline architecture of the approach is summarized in Figure 1. A CRF model [10] with a dictionary is used to recognize disease mentions. Then they are mapped to disease identifiers in a synonym dictionary. The disambiguation is implemented using extended semantic information extracted from MEDIC vocabulary and MEDLINE abstracts, which is used to calculate the similarity with the context information of an ambiguous disease mention. Finally, the disease identifier with the highest score is regarded as the identifier of the ambiguous disease name.

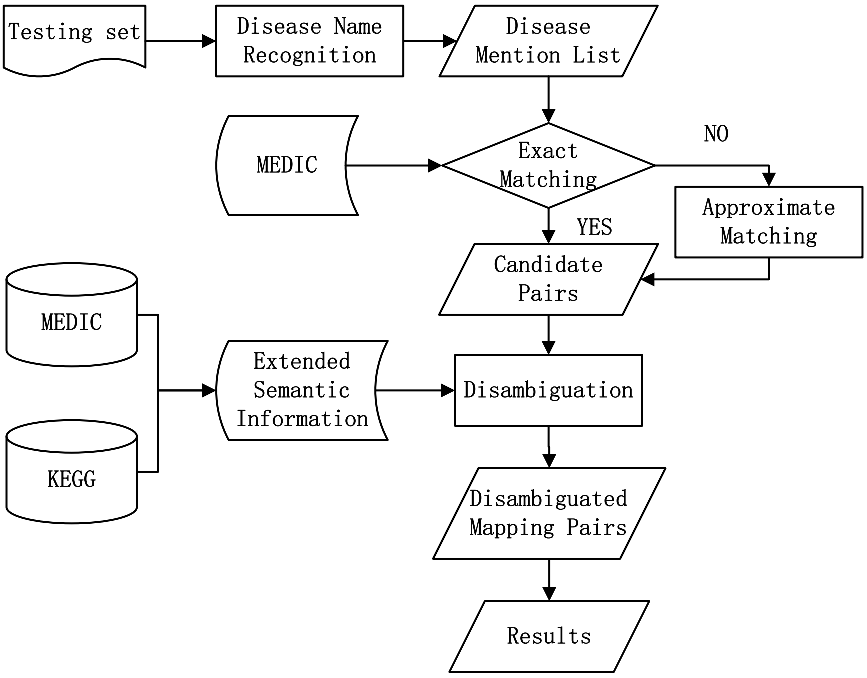

Figure 1 Architecture of our disease name normalization system

### 2.1.1 Disease Name Recognition

In this task, we combine a CRF model with a dictionary to recognize disease names. Firstly, a disease name dictionary is constructed using a publicly available biomedical resource, PharmGKB. Secondly, the dictionary search features are introduced into the CRF model, which will help to improve the recognition performance. Finally, the contextual cues of full disease names with their abbreviations are employed to further improve the recognition performance.

1) Disease Name Dictionary Construction

A disease name dictionary was constructed using PharmGKB (http://www.pharmgkb.org/downloads) which consists of 3,204 disease names with multiple alternate names. These alternate names were added to the disease dictionary for the sake of improving the coverage of the dictionary. Finally, 28,596 disease names are included in disease dictionary. In addition, the tagged disease names from the training set of the CDR task are also introduced into the dictionary.

2) Disease Name Recognition

BANNER [10] is used as our CRF-based tagger since recent studies have shown that it achieves significantly better performance than existing baseline systems. In addition, the following two types of lexical features are introduced into the BANNER system to improve the performance [11].

a) Prefix match features: conjunction of 'part in dictionary' and 'depth of prefix'.

b) Strict match features: conjunction of 'is in dictionary' and token number of the dictionary entry.

3) The Contextual Cue of Full Name-Abbreviation Pairs

There are many full name-abbreviation pairs of disease names (e.g. "X-linked Adrenoleukodystrophy (X-ALD)") in biomedical text. Our approach use a full name and abbreviation extraction algorithm, similar to Schwartz and Hearst [12], to extract these pairs and adjust the recognition results with them. Table 1 presents an example of the adjusted recognition result using full name-abbreviation pair contextual cues.

Table 1 An Example of the Adjusted Recognition Result Using the Contextual Cue of the Full Name-Abbreviation Pairs

| **Before Adjustment** | | **After Adjustment** | |
|---|---|---|---|
| adenomatous | B-disease | adenomatous | B-disease |
| polyposis | I-disease | polyposis | I-disease |
| coli | I-disease | coli | I-disease |
| ( | O | ( | O |
| APC | O | APC | B-disease |
| ) | O | ) | O |

### 2.1.2 Entity Mapping

In the disease normalization task, we need to link disease mentions to the terms in the database. In this task, dictionary look-up combining the exact and approximate matches is used.

1) Dictionary Construction

The dictionary we used is MEDIC disease vocabulary, which is composed of 9,700 unique diseases described by more than 67,000 terms (including synonyms). In addition, the disease entities annotated in the training set are also added in the disease dictionary.

2) Exact String Matching

Some heuristic rules are employed to improve the coverage of the disease dictionary and precision during the exact matching.

a) If the disease mention contains a space or hyphen, both the original form and the variants without the delimiter are considered.
b) For the disease mention including slash, the strings on both sides of the oblique are considered as two disease names to match the disease dictionary.
c) All disease names in the dictionary are converted to lowercase.

3) Approximate Matching

Sometimes only a part of disease names can be covered by exacting match. To solve the problem, an approximate matching method based on information retrieval is used, in which the disease mention without mapping are treated as a query and disease names of the disease dictionary as documents. Then the query term is used to search the disease dictionary, and the similarity between query term and disease names in the dictionary is calculated. Those with the similarities greater than and equal to 0.6 are chosen as the final candidates. Here BM25[13] retrieval algorithm is used.

### 2.1.3 Disambiguation

In the stage of approximate matching, there are multiple candidate identifiers for one disease mention. In order to further determine the specific identifier of the disease mention, context relevant to the disease mention as well as extended semantic information associating with the candidates are extracted. Then the similarity between them is calculated and used to choose the most related disease identifier for this disease mention.

A retrieval algorithm based on vector space model (VSM) is used to calculate the similarity. We use the bag-of-words as the features and TF-IDF is used to calculate the weight of the feature. The context of disease mention in the test set is treated as the query vector and the extended semantic information as the document vector, which comprises of two parts, i.e. MEDIC disease vocabulary and MEDLINE.

The query vector is represented as $Q$ and document vector as $D$. similarity between $Q$ and $D$ is calculated and the disease identifier with the high score is the final result. We use the TF-IDF to calculate the weight of features. The formula is defined as follows:

$$w_{t,d} = \frac{tf_{t,d} * \log(N/n_t + 0.01)}{\sqrt{\sum_{k \in d} [tf_{k,d} * \log(N/n_k + 0.01)]^2}} \qquad (1)$$

where $tf_{t,d}$ is the frequency of term *t* appearing in the document *D*, $n_t$ is the number of documents including term t, and $N$ is the number of all the documents. A simple cosine is used to calculate similarity, which is defined as follows:

$$Sim(Q, D) = \cos\theta = \frac{\sum_{k=1}^{n} W_{1k} \times W_{2k}}{\sqrt{(\sum_{k=1}^{n} W_{1k}^2)(\sum_{k=1}^{n} W_{2k}^2)}} \qquad (2)$$

In this formula, $W_{1k}$ and $W_{1k}$ are the weight of the *k*th elements in vectors *Q* and *D*, and *n* is the dimension of the vector model. In our method, the disease identifier with the high score is used as the identifier of the disease mention.

### 2.2 CID subtask

CID subtask evaluates the performance of the chemical-disease relations at the ab-stract level. Our solution first extracts the chemical-disease relations at the sentence level and then forms the relations at the abstract level. As the flow chart shown in Figure 2, we, firstly, constructed the training and development sets at the sentence level using the training and development sets provided by the task orgnizaitor, respectively. Only if a pair of chemical-disease (an instance) in a sentence has true chemical-induced diseases relation (the relation is explicitly mentioned in the sentence), it is labeled as a postive example. Otherwise, it is labeled as a negtive example. With these annotation rules, we manually labeled 1,200 positive examples and over 3,100 negative examples. In addition, we labeled 1,400 positive and 2,900 negative examples in the development set for tuning the system. Subsequently, we use the model to extract the CID relations at the sentence level, and then use some rules (as will be discussed in later section) to obtain the final results at the abstract level.

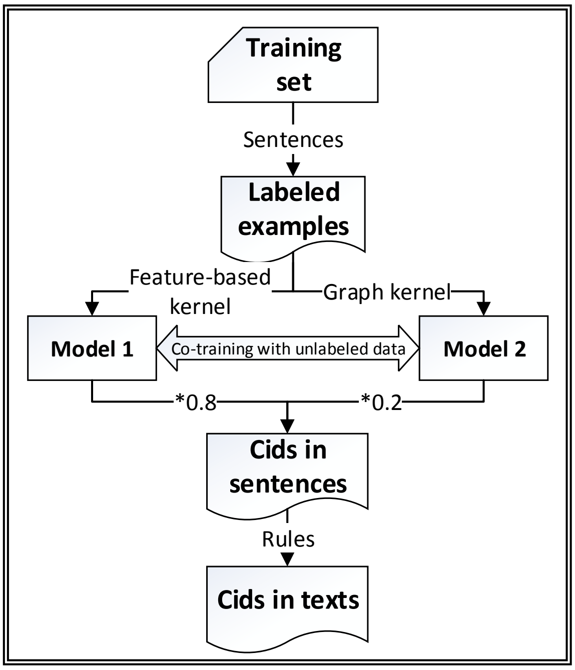

Figure 2 Architecture of our chemical-disease relation extraction system

In our method, we use kernel-based methods to extract CID relations. A kernel can be thought of as a similarity function for pairs of objects. Different kernels calculate the similarity with different aspects between the two sentences. In our method, we combine two types of kernels to extract CID relations, i.e. the feature-based kernel and graph kernel [14].

### 2.2.1 Feature-based kernel

In our experiments the following features are used in the feature-based kernel:

1) Word feature

The word kernel takes two unordered sets of words as feature vectors to calculate their similarity.

- Words between two entities: All words located between two entities are included in these features. And the features are labeled as “E_B_feature”.

- Words surrounding two entities: These features include left N words of the first en-tity name (labeled as “E1_L_feature”) and right N words of the second entity name (labeled as “E2_R_feature”). N is the number of surrounding words considered which is set to be four in our experiment.

2) N-gram words

N-gram features extend word feature by using 2-gram and 3-gram words as features.

3) Entity name distance feature

The longer the distance (the number of words) between two entity names is, the less likely the two entity names have relation. Therefore, the distance is used as a feature. For example, if the number of words between two entity names is less than three, the feature will have the value “DISLessThanThree”.

4) Keyword feature

Some keywords, such as “induce”, nearby the two entity names usually imply the existence of the CID relation. To identify the keywords in the text, we built a keyword list of about 200 words manually, including verbs and phrases. The existence of keywords is chosen as a binary feature. In addition, the keyword itself is also considered as a feature. For instance, if the keyword “induce” exists in the sentence, it will be labeled as “Key_induce”.

### 2.2.2 Graph kernel

In the graph kernel method, a syntax tree is used to represent a graphic structure of a sentence. The similarity of two graphs is calculated by comparing the public nodes in the graphs. We used an all-path graph kernel which consists of two unconnected subgraphs. One represents the dependency structure of the sentence, and the other the linear order of the words [15] (see Figure 3). We chose a simple weighting scheme where all edges on the shortest paths receive a weight of 0.9 and other edges receive a weight of 0.3. And each edge in the second subgraph is given the weight 0.9.

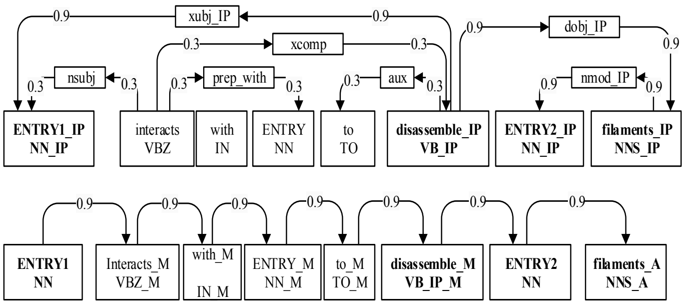

Figure 3 Graph kernel representation

The similarity of two input graphs is calculated by matrix *G*, which defined as:

$$G = L\sum_{n=1}^{\infty} A^n L^T \tag{3}$$

Where, *A* is an edge matrix, in which the element $A_{ij}$ is a weight of the edge connecting vertex $V_i$ and $V_j$. L is the label matrix; $L_{ij}$ =1 indicates that vertex $V_j$ contains the *i* label. With two input graph matrices *G* and *G* ', the graph kernel K (*G*, *G* ') is defined as:

$$k(G,G^{'}) = \sum_{i=1}^{L}\sum_{j=1}^{L} G_{ij} G_{ij}^{'} \tag{4}$$

### 2.2.3 Co-training algorithm

We trained two different classifiers based on feature-based kernel and graph kernel, respectively. Then using co-training algorithm, we trained two different classifiers based on feature-based kernel and graph kernel, respectively. The initial Co-Training algorithm [16] (or standard Co-Training algorithm) is proposed in 1998 by Blum et al. Our algorithm uses small set of labeled examples in the training set and a large number of unlabeled examples downloaded from PubMed to train a pair of classifiers. In the beginning, two classifiers were trained on the labeled examples. Then the unlabeled examples were classified by those two classifiers. Subsequently, the unlabeled examples labeled by one classifier confidently were added, with labels, to the training set of the other classifier. Thus, with the new training set to train the classifier, we could achieve two new and more effective models. We finally obtain the most efficient model by repeating the process four times when the F-score reaches the peak.

In order to obtain the better classification results, the final confidence score of one example equals the score calculated by two classifiers. The graph kernel and feature-based kernel classifiers were given the weight 0.2 and 0.8, respectively.

### 2.2.4 Final CID relation extraction

To extract the final CID relations at the abstract level with obtained relations at the sentence level, we applied the following rules: 1. if a CID relation is extracted in the title of the PubMed abstract, the confidence score of the relation will be added an extra value (0.2). 2. If a CID relation is extracted at the sentence level more than once, the score of the relation will be improved according to the frequency. The final CID relation score *score_f* is defined as follows:

$$score\_f{=}score\_h + 0.15{*}f + 0.2 \qquad \text{If the CID relation is extracted in the title of a PubMed abstract;}$$

$$score\_f{=}score\_h + 0.15{*}f \qquad \text{Otherwise;} \qquad (5)$$

where *score_f* is the highest score obtained at the sentence level and *f* represents the extraction frequency of the CID relation.

## 2.3 Results

### 2.3.1. DNER subtask results

The training set, development set and, test set of the two CDR subtasks are all 500 PubMed abstracts. The results on the development and test sets are shown in Table 2. The dictionary look-up is the baseline provided by the task organizers. It can be seen that our method achieves almost equal performance on the development set and test set.

Table 2. Results on the development and test sets

| Method | Precision(%) | Recall(%) | F-score(%) |
|---|---|---|---|
| dictionary look-up | 42.71 | 67.46 | 52.30 |
| Our method<br>(on development set) | 65.36 | 64.70 | 65.03 |
| Our method<br>(on test set) | 64.33 | 64.59 | 64.46 |

The F-score of our disease mention matching on the development set is 81.98% (Precision 84.81% and Recall 79.34%) while that of concept id matching drops to 65.03%. The error causes were analyzed. Of all the errors, the majority of the errors can be traced to the disease named entity recognition component, including non-annotated errors (which often occur when disease names are very short and include no obvious disease name features, like “diabetes” or “adenoma”), partial match errors (which mainly occur when some descriptive adjectives are annotated as parts of the following entity while others are not), and incorrectly annotated errors.

The second most error was mapping errors. Such errors mainly occur in the steps of approximate matching and disambiguation. In the step of approximate matching, incorrect candidates lead to the incorrect result. And in the step of disambiguation, some disease mentions are assigned to the uncorrected disease identifier.

### 2.3.2. CID subtask results

Our training set at the sentence level includes 1,200 positive and 3,100 negative instances. Our development set at the sentence level includes 1,400 positive and 2,900 negative examples used to adjust the system parameters. Two annotation tools provided by the task organizers, i.e. DNorm and tmChem, were used to recognize and normalize the disease and chemical concept.

Table 3 shows the F-score of our method on the development set. Feature-based kernel outperforms graph kernel and their combination achieve better performance. Table 4 shows the results on test set at the abstract level. In the table, we compared our results with that of the co-occurrence method provided by the task organizers.

Table 3. The F-scores on the development set

| | **Feature-based kernel** | **Graph kernel** | **Combination of kernels** |
|---|---|---|---|
| **Sentence level** | 79.94% | 69.97% | 83% |
| **Abstract level** | 50.91% | 47.54% | 52% |

Table 4. Results on the test set

| | **Precision(%)** | **Recall(%)** | **F-score(%)** |
|---|---|---|---|
| **Co-occurrence method** | 16.43% | 76.45% | 27.05% |
| **Our method** | 39.23% | 32.46% | 35.52% |

The error causes were analyzed on the development set. Some main error types are listed as follows:

1. Annotation error in the sentence level. Our training and development sets in the sentence level were labeled manually. There may exist some noises in it.
2. Disease and chemical concept recognition and normalization errors. In our method, the disease and chemical concept ids are returned by DNorm and tmChem. However, "highest performance from DNorm requires the UMLS Metathesaurus to provide lexical hints to BANNER and also Ab3P to resolve abbreviations" (according to the readme.txt of DNorm installation document) and we did not install the UMLS Metathesaurus. Therefore, quite a few disease names were not recognized or normalized correctly, and, therefore, the corresponding CID relations could not be extracted.
3. Span sentence CID extraction error. Since our method only extracts the CID relations in a sentence, the CID relations that span several sentences could not be extracted.

## 3 Acknowledgment

This work is supported by the grants from the Natural Science Foundation of China (No. 61070098, 61272373 and 61340020), Trans-Century Training Program Foundation for the Talents by the Ministry of Education of China (NCET-13-0084) and the Fundamental Research Funds for the Central Universities (No. DUT13JB09).

## REFERENCES

1. Zweigenbaum P, Demner-Fushman D, Yu H, Cohen KB. "Frontiers of biomedical text mining: current progress." Brief Bioinform. 2007, 8:358-375.
2. Cohen AM, Hersh WR. "A survey of current work in biomedical text mining." Brief Bioinform. 2005, 6:57-71.
3. Kang N, Singh, B., Bui, C., et al. "Knowledge-based extraction of adverse drug events from biomedical text." BMC Bioinformatics 2014,15, 64.
4. Krallinger M, Erhardt RA, Valencia A. "Text-mining approaches in molecular biology and biomedicine." Drug Discov. Today 2005, 10:439-445.
5. Wei CH, Peng Y, Leaman R, et al. "Overview of the BioCreative V Chemical Disease Relation (CDR) Task." in Proceedings of the fifth BioCreative challenge evaluation workshop, Sevilla, Spain, 2015.

6. Li J, Sun Y, Johnson R. et al. "Annotating chemicals, diseases, and their interactions in biomedical literature." in Proceedings of the fifth BioCreative challenge evaluation workshop, Sevilla, Spain, 2015.
7. Leaman R, Islamaj Dogan R, Lu Z. "DNorm: disease name normalization with pairwise learning to rank." Bioinformatics, 2013, 29: 2909-2917.
8. Wiegers TC, Davis AP, Mattingly CJ. "Web services-based text-mining demonstrates broad impacts for interoperability and process simplification." Database (Oxford), 2014, bau050.
9. Wiegers TC, Davis AP, Mattingly CJ. "Collaborative biocuration--text-mining development task for document prioritization for curation." Database (Oxford), 2012, bas037.
10. Leaman R and Gonzalez G. "BANNER: an executable survey of advances in biomedical named entity recognition." in Pacific Symposium on Biocomputing, Hawaii, USA, 2008, 652-663.
11. Li Y, Lin H, and Yang Z. "Incorporating rich background knowledge for gene named entity classification and recognition." BMC bioinformatics, 2009,10, 223.
12. Schwartz A and Hearst M. "A simple algorithm for identifying abbreviation definitions in biomedical text." in Pacific Symposium on Biocomputing, Hawaii, USA, 2003(8):451-462.
13. Jones KS, Walker S, and Robertson SE. "A probabilistic model of information retrieval: development and comparative experiments: Part 1." Information Processing & Management, 2000, 36: 779-808.
14. Yang Z, Lin H, Li Y. "BioPPISVMExtractor: A protein–protein interaction extractor for biomedical literature using SVM and rich feature sets." Journal of biomedical informatics, 2010, 43(1): 88-96.
15. Airola A, Pyysalo S, Björne J, et al. "All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning." BMC bioinformatics, 2008, 9(Suppl 11): S2.
16. Blum A, Mitchell T. "Combining labeled and unlabeled data with co-training." In Proceedings of the eleventh annual conference on Computational learning theory. ACM, 1998: 92-100.