PE_TU Participation at TAC 2018 Drug-Drug Interaction Extraction from Drug Labels

Yue Zhang[1], Parisa Kordjamshidi[2],
[1]Peking University, China [2]Tulane University, USA
[1]zhangyuejoslin@pku.edu.com [2]pkordjam@tulane.edu

Abstract

The Drug-Drug Reaction Extraction at NIST TAC 2018 conference aims to extract information on drug reactions from drug labels. In this paper, we describe the PE-TU' model participated in mention extraction task. We provided a hybrid method that combines a dictionary-based, rule-based and machine learning techniques to extract mention from mention label text. The pipeline process of our method contains data graph creating, feature designing and model training, which are built on the machine learning platform, Saul. The experimental results prove the efficiency of our methods.

1 Introduction

The Drug-Drug Interaction Extraction from Drug Labels at NIST TAC 2018 aims to extract information of interactions among drugs. The purpose of DDI is to test various Natural Language Processing (NLP) methods for their information extraction performance on drug-drug interactions from the heterogeneous textual sources which contain different drugs, supporting researchers and clinicians with the challenging task of transforming unstructured into structured data and improve the patient's drug safety. The labeled drugs can be accessed, searched and sorted electronically, which is an essential step towards the creation of a fully automated health information exchange system. The track consists of four subtasks: 1) Extracting mentions of interacting drugs/substances, interaction triggers and specific interactions at sentence level 2) Identifying interactions at sentence level, including the interacting drugs, the specific interaction types 3) Normalizing 4) Generating a global list of distinct interactions for the label in normalized form. Our team (PE_TU) participated in Task 1, which is similar to Named Entity Recognition (NER) task in nature.

NIST provides TAC participants with annotated data for training and plain test for test for training and testing. Dealing with health-related data requires dealing with complex vocabularies and specific syntax analysis, and our team proposed system adopted a hybrid approach combining dictionary-based matching, rule-based extraction, and a support vector machine (SVM) classifier (Hsu et al. 2003) to identify the label of entities in sentences.

The dictionary matching is the simplest method, and there are several resources developed by different institutions that can be used for this purpose, such as PubMed[1], SNOMED[2] and ICD[3]. We used adopted MedDRA[4] as a source of information about the drug names and used it in our feature extraction step. By analyzing the training data, we found a set of rules that helped in feature extraction and candidate generation. The dictionary matching and our preprocessing rule set enhance the features of entities for training machine learning models. Though we use classical classification techniques we use the machine learning framework called Saul (Kordjamshidi et al., 2015) that helps relational feature engineering and specifically extraction of linguistically-motivated features based on the underlying Cogcomp NLP tools. It is a helpful tool for relational domains and working on graph representations of the data (Kordjamshidi et al., 2018).

In this workshop notebook paper, we describe our participation at TAC DDI track with the proposed system for drug entity name recognition extraction and present the experiment results.

[1] https://www.ncbi.nlm.nih.gov/pubmed/
[2] http://www.snomed.org/
[3] http://www.who.int/classifications/icd/en/
[4] https://www.meddra.org/

2 Data Description

The dataset used for the tasks of DDI extraction other non-drug substances.

Specific Interaction: Results of interaction. consists of 22 annotated drug labels in XML format for training, and at least 50 unlabeled data for each testing set. The gold standard contains the following entity-style annotations.

Trigger: Trigger words and phrases for an interaction event.

Precipitant: A substance interacting with the labeled drug.

All labels are in XML format, and contain a `<Text>` element with one or more `<Mention>` elements with different type attributes consisting of Precipitant, Trigger and Specific Interaction.

3 Entity Extraction Method

The Drug-Drug interaction extraction task (DDI) of the TAC shared task requires systems to identify all mentions in the drug label text. The proposed method is designed using Saul platform, a machine learning framework. Saul emphasizes separating the aspect of data modeling and knowledge representation from the configuration of the learning models. So, people can design their data model, feature and classifier based on their own needs. Saul is programmed by Scala, which can help to have a more declarative problem specification and write declarative learning based programs. The project code of Saul is on *https://gtihub.com/CogComp/Saul*.

Using Saul framework, we designed the following three components for our model:

- First, **building data model** includes creating data graph, generating candidates and splitting training/testing dataset.
- Second, **designing feature** identifies various features that is suitable for different mentions.
- Third, **training classifier** learns from annotated training data to train different classifiers and predict the labels from the plain test data.

In this section, we describe the development of these components.

3.1 Building Data Model

Saul can directly read linguistically annotated co-

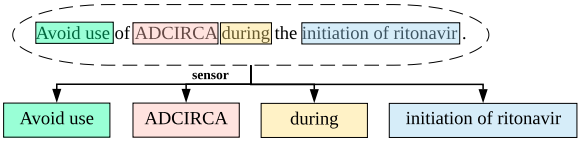

Fig 1: the data graph of a sentence, which populates four phrases. The sensor is the link of these two units.

rpora with XML format and put it in a graph structure that contains various linguistic units including sentences, phrases, tokens, etc. The XML format is almost the same as the XML format that TAC conference provided except some different attributes. The schema of the graph and the type of linguistic units is defined by the programmers.

3.1.1 Data Graph

After reading the data, we need to populate data into a declared data graph. Since for the drug label task, we need to label the trigger, precipitant and specific interaction from sentences, the linguistic units in data graph should be sentences and phrases. Regarding links, we need to add a sensor, which can generate phrases from sentence automatically with the integrated shallow parser. The data graph is illustrated as Fig 1.

3.1.2 Generating Candidates

To generate the training and testing candidates, we use an integrated shallow parser (i.e., Chunker) in Saul to get the phrases of each sentence. Then we match each chunk with the labeled data in each sentence. If the chunk has the same head word as the labeled data, this chunk will get the same label. For instance, the sentence in Fig 1 is "Avoid use of ADCIRCA during the initiation of ritonavir," and "avoid use" is labeled as "Trigger" in training data. Our chuncker generates "avoid" and "use" as to separate chuncks. In such a case, we label both of them as Trigger.

We noticed that many sentences in training data that contain drug mentions are not labeled since there is no drug interaction. Since our focus was on the drug mentioned we augmented the training with additional automatic annotations by matching the phrases and the MedDRA dictionaries (Brown et al. 2019) to identify drug and disease names. Finally, we got a total of 7827 phrases.

Ratio	F1-Score
7:1	47.347%
8:2	45.344%
9:3	43.321%
10:3	39.522%
11:4	38.621%

Table 1: F1 score of mention extraction based on different ratio of training and validation datasets.

3.1.3 Splitting training and test sets

Since there is no testing data provided at the beginning, we adopted the cross-validation method to split all candidate phrases (7827 phrases) into training, validation and testing datasets randomly to train model.

According to the number of generating phrases of each testing set (around 300). We tried to take out 300 phrases randomly as testing data, and remaining are training and validation data, which can be split into eight parts. We varied the ratio, and the result showed that the perfect ratio is 7:1 (See Table 1). After getting the optimal features and parameters for the model, we trained the models with the whole training set to predict the labels of the generated candidate phrases in testing datasets.

3.2 Feature Designing

We used various linguistically motivated features including syntactic and semantic features (Kambhatla N. 2004) as well as the pre-trained word2vec dense vector (Mikolov et al. 2013) as common features for four classes (which are the Trigger, Precipitant, Specific Interaction, and none). The following list is the common features for classes of mention types.

- *Pos*: the pos-tag of each word in the phrase.
- *Word form*: the exact words in the phrase.
- *Head wordform*: the head word of the phrase.
- *Phrase Pos*: the pos-tag of the whole phrase which is the concatenation of pos of the including tokens.
- *Lemma*: the lemma of phrase which is the concatenation of the lemma of the including tokens.
- *Head word Lemma*: the lemma of head word.

Features	F1-score
syntactic features	29.113%
syntactic + semantic features	36.522%
syntactic + semantic +word2vec	41.321%

Table 2: the F1-score with various features.

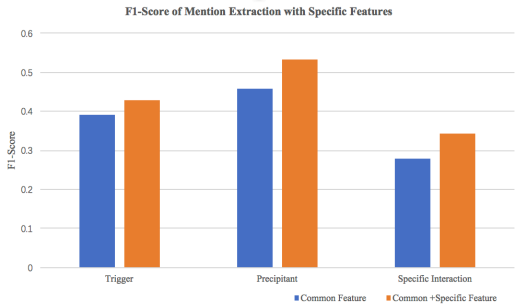

Fig 2: F1-socre of Mention Extraction with Specific Features.

- *Subcategorization*: the categorization of each phrase.
- *dependency Relation*: the dependency relation of each phrase in sentences (De Marneffe et al. 2014).
- *Head Dependency Relation*: the dependency relation of head word of each phrase.
- *Head subcategorization*: the categorization of head word in each phrase.
- *Word2vec*: the pre-trained google news word dense vector.

According to the experiment, the semantic feature such as dependency relation and categorization can improve the F1 score of extraction about 5%-8%. The word2vec feature can improve the F1-score about 3%-5%. The experiment result is shown in Table 2. The syntactic features contain pos, word form, headword form, phrase pos, lemma and headword form, phrase pos, lemma and headword lemma. The semantic features comprise dependency, headword dependency relation, and subcategorization.

It is evident headword is an important feature for our model. However, the shallow parser has errors, so does the headword extraction tool. For instance, the headword extracted with our tools for the phrase "should be monitored" is "should," which is wrong. To solve this error, we make some rules to modify

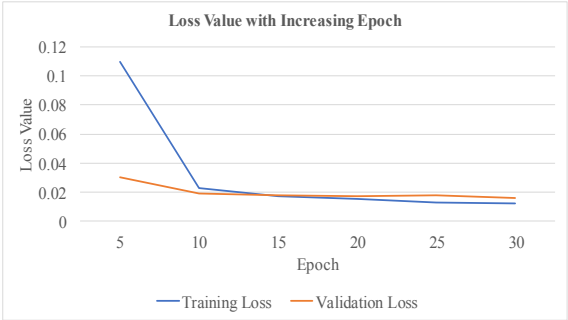

Fig 3: Loss Value with Increasing Epoch for Trigger classification

headword extraction for our model. Since the trigger, is always a verb phrase, the pos-tag of headword should be the verb. By using this rule, the headword of "should be monitored" becomes "monitored" which is the correct candidate for triggers. By adding these rules, the F1-score of entity extraction improve about 3%. The experimental results show that some features are more useful for a specific class.

Regarding trigger, since most of the triggers are verb phrases, we added the feature of pre-trained phrase2vec (Yimam et al. 2018) to capture the verb phrase information.

As for precipitants, since most of the phrases are drug and entity name, so we used drug dictionary feature to distinguish precipitant. The dictionary resource came from MedDRA, which is the international medical terminology developed under the auspices of the International Conference.

For specific interactions, we use the common features as well as some rules. Most phrases of specific interactions contain "effect." So, we expand specific interactions on the phrase with "effect" in it after classification.

Figure 2 shows the performance of mention recognition with their specific feature. The experiment result shows that the specific feature improves the performance of mention extraction

3.3 Model Training

There are four models used to train, which are phrase2vec and three SVM classifiers.

The phrase2vec takes the average of pre-trained word vector of the words to capture verb phrase information of trigger. The pre-trained word vector came from google news, and the dimension is 300.

The three SVM classifiers are for the classification of Trigger, Precipitant, and Specific Interaction respectively. The parameters

Metric	Soft Match	Exact Match
Precision	45.23%	32.33%
Recall	46.02%	35.13%
F1-score	45.62%	33.68%

Table 3: Experiment Results for Trigger

Metric	Soft Match	Exact Match
Precision	71.62%	53.55%
Recall	73.51%	54.72%
F1-score	72.57%	55.14%

Table4: Experiment Results for Precipitant

Metric	Soft Match	Exact Match
Precision	31.24%	25.55%
Recall	32.57%	26.78%
F1-score	31.91%	26.17%

Table5: Experiment Results for Specific Interaction

of the three models are same. The learning rate is 0.2; the thickness is 1.

We trained each of three classifiers for 30 epochs, and ultimately used the model that was saved when the training and validation lose vegan diverge after epoch 15. So, for prediction, we used the model saved after epoch 15.

4 Evaluation and Results

To measure the performance of the model, we evaluated the precision, recall, and F-score of the split testing set (around 300 phrases). We examined two metrics, which are approximate matching, and exact matching. The soft matching considers the predicted mention span to be correct if it overlaps with any ground-truth span, and exact matching considers a predicted mention span to be correct only if it exactly matches a ground-truth span. The experiment result can be shown as follows:

Regarding error analysis of Trigger, we found that the shallow parser we used cannot capture the correct verb phrases. And also, the rule we used for headword (The pos-tag of the headword for the trigger is a verb) leads to the loss of some verb noun.

As for Precipitant classification, the errors occur mainly from training datasets. As mentioned

before, there is no label for precipitant if there is no drug interaction. Most of the precipitant is drug name and illness name. If there is no enough training data, the accuracy would be very low, so we labeled some training data based on the dictionary. It may be more efficient to match the training data with the dictionary directly, but we want to treat the dictionary as a kind of feature, that implies we need to have enough training examples still.

Regarding specific interaction, we noticed that the shallow parser can't locate the exact trigger phrase.

5 Conclusion

In this paper, we describe our participation at TAC DDI track. The proposed method adopts a hybrid approach combining dictionary-based, rule-based preprocessing and using classical machine learning techniques.

We use Saul framework, a convenient machine learning platform that facilitates createing the data graph, and extraction of relational linguistic features and training models. Saul provides the tools for joint inference which we did not use at this stage of our models and will be used to improve our models in our next steps of working on this task/data. The integration of additional resources such as MedDRA dictionary is used to improve the performance of mention recognition. And also, the rule-based method for some cases is proved useful.

Our results are very at the preliminary stage and our experiments show the necessity of using more sophisticated and label-specific set of features for the extraction of each mention. In the future work, we will explore deep learning models and perform join training and inference for mention and relation classification tasks to deal with the lack of the training examples.

6 References

Kordjamshidi P, Dan R, Wu H. *Saul: towards declarative learning based programming*[C]// International Conference on Artificial Intelligence. AAAI Press, 2015:1844-1851.

Kordjamshidi P, Roth D, Kersting K. *Systems AI: A Declarative Learning Based Programming Perspective*[C]//IJCAI. 2018: 5464-5471.

Brown E G, Wood L, Wood S. *The medical dictionary for regulatory activities* (MedDRA)[J]. Drug safety, 1999, 20(2): 109-117.

Mikolov T, Sutskever I, Chen K, et al. *Distributed representations of words and phrases and their compositionality*[C]//Advances in neural information processing systems. 2013: 3111-3119.

Hsu C W, Chang C C, Lin C J.*A practical guide to support vector classification*[J]. 2003.

Yimam S M, Biemann C. *Par4Sim--Adaptive Paraphrasing for Text Simplification*[J]. arXiv preprint arXiv:1806.08309, 2018

Kordjamshidi P, Singh S, Khashabi D, et al. Relational Learning and Feature Extraction by Querying over Heterogeneous Information Networks[J]. arXiv preprint arXiv:1707.07794, 2017.

Kambhatla N. *Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations*[C]//Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 2004: 22.

De Marneffe M C, Dozat T, Silveira N, et al. *Universal Stanford dependencies: A cross-linguistic typology*[C]//LREC. 2014, 14: 4585-4592.