Wuhan University
Journal of Natural Sciences

2016, Vol.21 No.6, 482-490

Article ID 1007-1202(2016)06-0482-09
DOI 10.1007/s11859-016-1200-7

A Multiple Feature Approach for Disorder Normalization in Clinical Notes

☐ LÜ Chen[1], CHEN Bo[1,2], LÜ Chaozhen[1], QIU Likun[3], JI Donghong[1†]

1. School of Computer, Wuhan University, Wuhan 430072, Hubei, China;

2. Department of Chinese Language and Literature, Hubei University of Art and Science, Xiangyang 441053, Hubei, China;

3. Shandong Key Lab of Language Resource Development and Application, Ludong University, Yantai 264025, Shandong, China

Abstract: In this paper we propose a multiple feature approach for the normalization task which can map each disorder mention in the text to a unique unified medical language system (UMLS) concept unique identifier (CUI). We develop a two-step method to acquire a list of candidate CUIs and their associated preferred names using UMLS API and to choose the closest CUI by calculating the similarity between the input disorder mention and each candidate. The similarity calculation step is formulated as a classification problem and multiple features (string features, ranking features, similarity features, and contextual features) are used to normalize the disorder mentions. The results show that the multiple feature approach improves the accuracy of the normalization task from 32.99% to 67.08% compared with the MetaMap baseline.

Key words: natural language processing; disorder normalization; Levenshtein distance; semantic composition; multiple features
CLC number: TP391

Received date: 2016-03-23
Foundation item: Supported by the National Natural Science Foundation of China (61133012, 61202193, 61373108), the Major Projects of the National Social Science Foundation of China (11&ZD189), the Chinese Postdoctoral Science Foundation (2013M540593, 2014T70722) and the Open Foundation of Shandong Key Laboratory of Language Resource Development and Application
Biography: LÜ Chen, male, Ph.D. candidate, research direction: natural language processing. E-mail: lvchen1989@ whu.edu.cn
† To whom correspondence should be addressed. E-mail: dhji@ whu.edu.cn

0 Introduction

Clinical natural language processing (NLP) such as resource construction and machine learning methods plays an important role in recent years [1-3]. One of the fundamental tasks in clinical NLP is clinical concept extraction from clinical documents, especially electronic health records (EHRs) data. In general, there are two steps in this task: 1) recognizing clinical relevant entities (e.g., diseases, drugs etc.) in text; 2) mapping these entities to identifiers in standard vocabularies, e.g., concept unique identifier (CUI) in unified medical language system (UMLS) [4]. For free-text electronic health records, particularly, it is one of the most important tasks that the automatic identification of clinical conditions, anatomical sites, medications, procedures, and their normalization [5]. A shared task of clinical notes analysis in Semeval-2014 [6] includes two sub-tasks: 1) identifying the disorder mentions (DMs) from clinical notes, which is referred to as an identification task; 2) mapping each disorder mention to a unique UMLS CUI, which is referred to as a normalization task. Figure 1 shows two example sentences from clinical notes, together with disorder mentions and their respective normalization concepts.

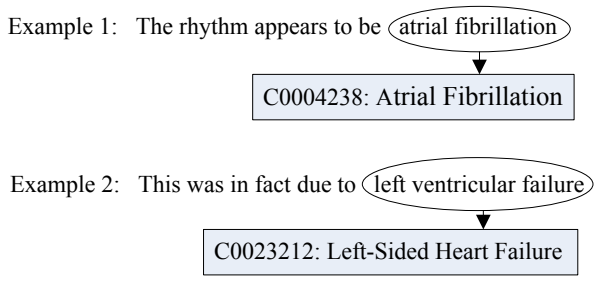

Fig. 1 Examples of normalization in clinical notes

There are four strategies for disorder mention normalization: 1) symbolic NLP systems; 2) approach based on vector space model (VSM); 3) dictionary lookup algorithm; 4) the rule-based methods. Currently, the approach based on VSM has been widely used in biomedical text processing [7-9], by which feature vector can be directly used to calculate the similarity between a disorder mention and their candidates. In this approach, different feature vector and similarity calculation method can be utilized in this method. However, continuous distributed word representation is not satisfactory for the normalization of disorder mention which has misspellings. While, for different phrase forms with same concept, the performance of Levenshtein distance method (one of the similarity calculation method) is not satisfactory.

In this paper, we propose an approach with multiple features including the similarity calculation, which can predicate the correct candidate for a given disorder mention and its candidates in clinical notes. Our approach can integrate the advantages and improve the performance of different similarity calculation methods.

1 Related Work

1.1 Four Strategies for Disorder Mention Normalization

There are four strategies for disorder mention normalization. Strategy 1 uses symbolic NLP systems to encode disorder mention to UMLS CUI. Representative systems include MedLEE [10], MetaMap [11], KnowledgeMap [12], cTAKES [13], HITEx [14] and YTEX [15]. Strategy 2 uses VSM-based approach to map each disorder to a unique CUI defined in UMLS. Strategy 3 uses different dictionary lookup algorithm to assign CUI to the identified disorder mention. Strategy 4 applies post-filtering rule to the intermediate results from other clinical concept extraction systems to filter wrong results.

Strategy 2 is commonly used for disorder mention normalization. In this strategy, because feature vector can be used to not only similarity calculation but also ranking algorithm, it avoids relying on domain knowledge and filter rules. This approach based on VSM is developed to map each disorder to a unique CUI defined in UMLS.

In this strategy, disorder mention normalization is treated as a ranking problem, where each recognized disorder entity is considered as a query and candidates terms in UMLS as documents. The process consists of two steps: 1) generate candidate CUIs from UMLS; 2) rank candidate CUIs and then choose the top ranked CUI as the systems output.

1.2 Feature Representation for Text

Traditionally, the feature vector for text is represented by TF-IDF method. The vector contains words in text, weighted by term frequency-inverse document frequency derived from the corpus.

Recently, researchers pay more attention to continuous distributed word representation, in which each word is represented by a low dimensional dense representation. It captures distributional syntactic and semantic information learned from a large corpus. Regarding the representation of larger constructions, such as phrases and sentences, semantic compositional model attracts much attention and has achieved considerable success in practical applications [16, 17].

2 Methods

2.1 Overall Structure of Our Method for Normalization

Figure 2 shows the overall structure of the method, which includes two sub-steps, retrieval and comparison. The retrieval step is to acquire a list of candidate CUIs and their associated preferred names using UMLS API, and the comparison step is to calculate the similarity between the input disorder mentions and each candidate, choose the closest one and output its CUI.

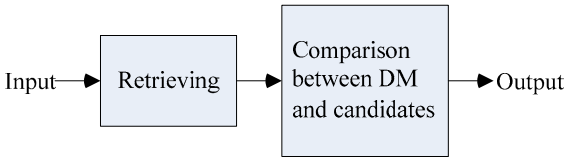

Fig. 2 Overall structure of our method for normalization

For the retrieval step, we use the findConcepts_NORMstr API provided by UMLS to acquire the candidate results. "Normalized String" is the searchType parameters in the API findConcepts and we set the source, which will be included in our result set, as the controlled vocabulary SNOMED-CT.

The API findConcepts_NORMstr can only be used for English language terms. It removes lexical variations such as plural and upper case text and compares search terms to the Metathesaurus normalized string index to retrieve relevant concepts. In Metathesaurus normalized string index, the normalization process involves breaking

a string into its constituent words, lowercasing each word, converting each word to its uninflected form, and sorting the words in alphabetic order. Normalized strings are generated by uninflecting each word, leaving out a small number of stop words.

For example, when you enter "Crohn's disease", it is normalized to "crohn disease". "C0010346|Crohn Disease" has the atom "Crohn's disease", which is also normalized to "crohn disease". So "C0010346|Crohn Disease" will be retrieved by the findConcepts_ NORMstr API. Table 1 gives some examples of the findConcepts_NORMstr API.

Table 1 Examples of the findConcepts_NORMstr API

Query	Retrieval results
Crohn's disease	C0010346|Crohn Disease
Red	C0332575|Redness
	C1260956|Red color

For the comparison step, the following methods are used to calculate the similarity between disorder mentions and the candidates.

2.2 Levenshtein Distance

In information theory and computer science, the Levenshtein distance [18] is a metric for measuring the difference between two sequences and is defined as the minimum number of edits needed to convert one term into another. The edits are in the form of insertions, deletions and substitution of characters. The phrase edit distance is often used to refer specifically to Levenshtein distance. We calculate the Levenshtein distance between phrases based on characters. For example, the Levenshtein distance between "hypokinesis" and "hypokinesia" is 1, since the term "hypokinesis" can be converted into "hypokinesia" in minimum 1 step by substituting "s" for "a". Then we can define the similarity between string *a* and string *b* as follows:

$$\text{similarity} = 1 - \frac{\text{Lev}(a,b)}{\max(\text{length}(a),\text{length}(b))} \quad (1)$$

In our method, we divide the disorder mentions into two types:

1) One Character It is difficult to deal with the disorder mention which contains only one character, and the same character is mapped to different CUI code in the training set. So we directly map all the one character disorder mention to "CUI-less".

2) Others The procedure for other types of disorder mentions is described in Fig. 3. FindC_NORMstr returns the top-20 candidates retrieved by the findConcepts

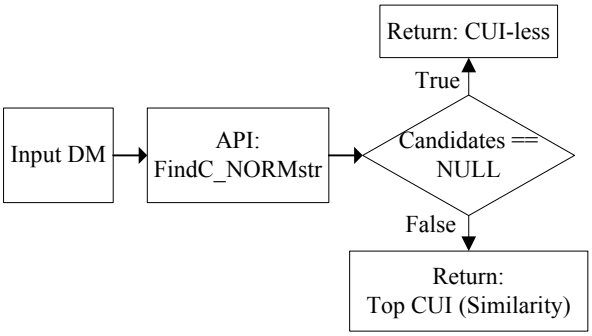

Fig. 3 Procedure for mapping different kind of disorder mentions to the UMLS CUI

NORMstr API. Top CUI returns the CUI with the highest similarity which is computed by Levenshtein distance.

When computing the similarity, we stem each word in the phrase using Porter Stemmer (*http://tartarus.org/~martin/PorterStemmer/index.html*).

2.3 Semantic Composition

2.3.1 Vector space model for words

The idea of neural language models introduced by Bengio *et al* [19] is to jointly learn an embedding of words into an *n*-dimensional vector space and to use these vectors to predict how likely it is for each word to occur in its context.

Collobert and Weston [20] introduced a new neural network model to compute such an embedding. The idea is to construct a neural network. While it outputs high scores for windows that occur in a large unlabeled corpus, low scores are output for windows that one word is replaced by a random word. The word vectors inside the embedding matrix capture distributional syntactic and semantic information via the word co-occurrence statistics.

Formally, we denote each word as a vector. These word vectors are then stacked into a word embedding matrix $\boldsymbol{W} \in \mathbf{R}^{n\times|V|}$, where $|V|$ is the vocabulary size. The resulting W matrix is used as follows. Given a sentence as an ordered list of m words, each word w has an index k into the embedding matrix $\boldsymbol{W}$ that we use to retrieve the words vector representation. This lookup operation can be seen as a simple projection layer:

$$x_i = \boldsymbol{W}\boldsymbol{b}_k \in \mathbf{R}^n \quad (2)$$

where $\boldsymbol{b}_k$ is a binary vector which is *zero* everywhere except at the k-th index.

This neural network will take a long training time (some weeks) on Wikipedia corpus to get word embeddings [20]. Word embeddings (CW word embeddings) provided by Huang *et al* [21], which are learned by incor-

porating both local and global document context following the framework of Collobert and Weston [20], are used for word representations in semantic composition. Wikipedia is chosen as the corpus to train word embeddings because of its wide range of topics and word usages, and its clean organization of document by topic. Another start-of-the-art method word2vec [22, 23], which is a relatively simple log-linear model, can be trained to produce high-quality word embeddings on the entirety of English Wikipedia text in less than half a day on one machine.

We train word embeddings (word2vec word embeddings) using this method on a set of unannotated clinical reports, which are provided in the task to support unsupervised learning methods.

2.3.2 Vector space model for phrases

Distributional semantic models (DSMs) approximate the meaning of words with vectors summarizing their patterns of co-occurrence in a large corpus. Recently, several compositional extensions of distributional semantic models (Compositional DSMs, or CDSMs) have been used to represent the meaning of phrases and sentences by composing the distributional representations of the words they contain [24-26].

Mitchell and Lapata [25] introduced a general framework for studying vector composition, which they formulate as a function f of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. Different composition models arise, depending on how f is chosen. In this model, each word of phrase or sentence is represented as a vector, and then through combinations, the vector for the phrase or sentence can be derived. There are many ways for combination, such as: Addition: $\boldsymbol{p}=\alpha\boldsymbol{u}+\beta\boldsymbol{v}$, Dilation: $\boldsymbol{p}=(\lambda-1)(\boldsymbol{u}\cdot\boldsymbol{v})\boldsymbol{u}+(\boldsymbol{u}\cdot\boldsymbol{u})\boldsymbol{v}$, or Multiplication: $\boldsymbol{p}=\boldsymbol{u}\odot\boldsymbol{v}, \boldsymbol{p}_i=\boldsymbol{u}_i\boldsymbol{v}_i$.

According to Blacoe and Lapata [16], simple composition can achieve very good results, and a complex composition may perform almost the same as a simple combination. Moreover, it may take a longer time for the complex combinations. So we use two kinds of addition composition. One is simple average addition composition: $\boldsymbol{p}=(\boldsymbol{u}+\boldsymbol{v})/2$, the other is a weighted addition composition: $\boldsymbol{p}=(\alpha\boldsymbol{u}+\beta\boldsymbol{v})/2$, where $\alpha+\beta=1$. For example, Table 2 shows the vectors for the individual words: "heart" and "failure", as well as the new vector for the phrase "heart failure" via simple average addition composition. In our method, the weights of word vectors are determined by the POS-tags of the words. The weight α is set to words which are determiner, preposition, conjunction or subordinating, and the weights β is set to other words. After obtaining the vectors, we use cosine similarity, as shown in Eq. (3), to calculate the similarity between the disorder mention and each candidate.

Table 2 Example of average addition composition

Word/Phrase	Atrial	Diastolic	Clinic	Mild
Heart	6	2	10	4
Failure	8	4	4	0
(Heart+failure)/2	7	3	7	2

$$\cos(\boldsymbol{a},\boldsymbol{b})=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\times\|\boldsymbol{b}\|}=\frac{\sum_{i=1}^{n}a_i\times b_i}{\sqrt{\sum_{i=1}^{n}a_i^2}\times\sqrt{\sum_{i=1}^{n}b_i^2}} \quad (3)$$

We divide the disorder mentions into two types as described in Section 2.1 and the similarity is computed by semantic composition model.

2.4 Multiple Feature Approach

Considering the advantages and disadvantages of both the Levenshtein distance and semantic composition methods for computing the similarity between phrases, we utilize the results of both methods and other effective features to get better result.

The comparison step can be formulated as a binary classification problem. The correct answer in the candidates is a positive instance, and others can be seen as the negative instances. Given a disorder mention and its candidates retrieved by API, the classification model decides which candidate is the correct CUI for the disorder mention.

We divide the disorder mentions into two types: 1) one character; 2) others as described in Section 2.1.

1) One Character We directly map all the disorder mention which contains only one character to "CUI- less".

2) Others Similar to the process described in Fig. 2, we directly label the disorder mention, which has no candidates retrieved by the findConcepts_NORMstr API, as "CUI-less". And the classification model is used to predict the result of the disorder mentions, which have candidates retrieved by API.

Given a disorder mention and its candidates, the classification model predicts the label for each candidate and the corresponding CUI can be got through the following steps.

1) If only one candidate of the disorder mention is labeled as positive, we get it as the correct answer for the disorder mention.

2) If all the candidates are labeled as negative, we label the disorder mention as "CUI-less".

3) If more than one candidate is labeled as positive, we choose the candidate, which ranks first in the re-

trieved candidate set, as the correct answer.

Multiple features can be utilized in the classification model, and support vector machines (SVM) machine learning method is used for the classification. The features are described as follows:

Ranking Features The findConcepts_NORMstr API retrieve the candidates for the disorder mention and the rank of the candidate in the retrieval results is a useful feature.

Similarity Features To integrate the advantages of the Levenshtein distance and semantic composition method, we use the similarity features computed by both methods.

1) Similarity between the disorder mention and the candidates computed by Levenshtein distance.

2) Similarity between the disorder mention and the candidates computed by semantic composition method using CW word embeddings.

3) Similarity between the disorder mention and the candidates computed by semantic composition method using word2vec word embeddings.

Contextual Features The disorder mention itself does not provide enough information for us to determine which candidate is the correct CUI, especially for the disorder mentions which are the acronym of the normalized disorder mentions. Contextual information should be considered in this task and we use the similarity between the disorder mention's context and the candidates as simple contextual features. We consider a fixed size t window of words around the disorder mention. We set the window size $t = 3$ in the experiment.

1) Similarity between the context and the candidates computed by semantic composition method using CW word embeddings.

2) Similarity between the context and the candidates computed by semantic composition method using-word2vec word embeddings.

String Features The idea of the string features is simple: if the surface string of the disorder mention is very similar to that of the candidate, it is an indication that the disorder and the candidate may have the same meaning and the candidate may be the corresponding answer for the disorder mention.

1) If the disorder mention is part of the candidate, the feature is 1. Otherwise, the feature is 0.

2) If the candidate starts with the disorder mention, the feature is 1. Otherwise, the feature is 0.

3) If the candidate ends with the disorder mention, the feature is 1. Otherwise, the feature is 0.

4) If feature(1) = 1, feature(2) = 0, and feature(3) = 0, the feature is 1. Otherwise, the feature is 0.

3 Experiment

3.1 Experiment Settings

We conduct our experiments on Semeval-2014 task7 data set. The organizer of this task has not released golden standard of test set, so the training set and the development set, which have golden standard, are used in the experiment. The data we used contains 302 clinical documents and 11 380 disorder mentions.

Two word embeddings are used in the experiments. One is CW word embeddings provided by Huang *et al* [21]. It is trained on Wikipedia corpus. The other is word2vec word embeddings. It is trained on the unannotated clinical reports provided in the task.

In this work, we use accuracy, as defined in Eq. (4), to evaluate the performance, where N_All is the number of all the disorder mentions in the data, and N_Correct is the number of those disorder mentions whose CUIs are correctly acquired.

$$\text{accuracy} = \frac{\text{N\_Correct}}{\text{N\_All}} \tag{4}$$

In addition, we also calculate Top-3 accuracy, with the assumption that if Top-3 results contain the true answer, it is regarded as the correct retrieval.

3.2 Baseline

We use MetaMap [11] as the baseline system. MetaMap is one of the foundations of NLM's medical text indexer (MTI) which is being used for both semiautomatic and fully automatic indexing of biomedical literature at NLM. MetaMap maps biomedical text to concepts in the UMLS Metathesaurus. Several types of lexical/syntactic analysis are performed on the input text to perform this mapping: tokenization, part-of-speech tagging, lexical lookup in the SPECIALIST lexicon and shallow parsing.

Each noun phrase obtained is applied in the next processes: variant generation, candidate identification, mapping construction and word sense disambiguation. Final scores are computed by combining different measures for each candidate mapping. It provides a JAVA API for users to find the CUI-code in the UMLS, and for an input disorder, the CUI-code will be returned.

3.3 Result of the Normalization by Levenshtein Distance

Table 3 shows the results of Levenshtein distance

method and the baseline. Compared with the baseline, Levenshtein distance method improves the accuracy from 32.99% to 63.97%.

Table 3 Comparision of Levenshtein distance method and the baseline for normalization

Method	Accuracy/%
Baseline	32.99
Levenshtein distance	**63.97**

Levenshtein distance method can provide correct CUI-code for the disorder mentions which include a misspelling. For example, "C0000833|Abscess" is the candidate of the disorder mention "abcess" retrieved by the findConcepts_NORMstrAPI, and "C0000833" is also the correct CUI. This disorder mention includes a misspelling and we can get the correct CUI by the similarity computed Levenshtein distance.

3.4 Results of Semantic Composition

We explore the performance of semantic composition model. The models are listed below:

- SC_{add} + CW: Simple average addition semantic composition and CW word embeddings are used.
- SC_{weight} + CW: Weighted addition combination semantic composition and CW word embeddings are used. We set the weights as $\alpha = 0.48$ and $\beta = 0.52$. The POS-tags are assigned by Stanford POS-tagger (*http://nlp.stanford.edu/software/tagger.shtml*).
- SC_{add} + word2vec: Semantic composition using simple average addition combination and word2vec word embeddings are used.
- SC_{weight} + word2vec: Weighted addition combination semantic composition and word2vec word embeddings are used. The weights are the same as SC_{weight} + CW.

Table 4 shows the results of the six models. From the table, we can see that the accuracy of semantic composition model is higher than that of the baseline. Compared with the baseline, SC_{add} + word2vec model improves the accuracy from 32.99% to 64.23%. The performance of each semantic composition method using word2vec word embeddings is superior to the method using CW word embeddings.

Table 4 Results of semantic composition

Method	Accuracy/%
Baseline	32.99
Levenshtein Distance	63.97
SC_{add} + CW	63.95
SC_{weight} + CW	63.94
SC_{add} + word2vec	**64.23**
SC_{weight} + word2vec	**64.22**

To investigate the influence for different weight settings in SC_{weight} + word2vec model, we change the weight α from 0.12 to 0.84.

Figure 4 shows the results of SC_{weight} + word2vec model with different weights. It indicates that both types of accuracy (accuracy and Top-3 accuracy) do not change significantly with different weights. So we randomly set the weights as α= 0.48 and β= 0.52 in the model.

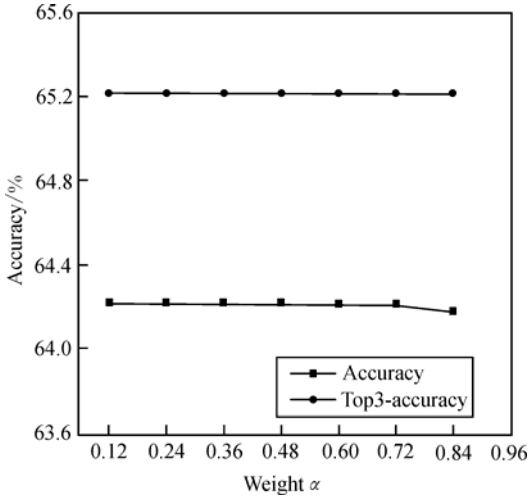

Fig. 4 Results of SC_{weight} model with different weights

Two CUIs, "C0011127|Pressure Ulcer" and "C0442303|Decubitus direction", are the candidates of the disorder mention "decubitus" retrieved by the findConcepts_NORMstr API, in which "C0011127|Pressure Ulcer" is correct. Levenshtein distance method cannot give the correct answer, while "C0011127|Pressure Ulcer" has the highest similarity with this disorder mention in semantic composition model. Different phrase forms, which are representations of the same concept, are close each other in semantic vector space. Semantic composition model helps to normalize the disorder mentions which have quite different phrase forms with correct CUI.

3.5 Results of Multiple Feature Approach

As described in Section 2.4, we process the dataset firstly, and then 4 095 disorder mentions is labeled as "CUI-less". The rest 7 285 disorder mentions need to be processed by the multiple feature approach.

We divide the rest disorder mentions, which need to be processed by the multiple feature approach, into 10 sections, and conduct 10-fold cross-validation to get the result of them. In feature extraction, we use SC_{add} for the semantic composition method. LIBSVM (*http://www.csie.ntu. edu.tw/~cjlin/libsvm/*) classifier, which is a li-

brary for SVM, is used in the experiment and we choose the RBF kernel function for the classifier. Table 5 shows the result of the multiple feature approach on the dataset. Compared with Levenshtein distance method and SC_{add} + word2vec model, the multiple feature approach improves the performance by utilizing multiple features (ranking features, similarity features, contextual features and string features). In addition, compared with the baseline, our multiple feature approach improves the accuracy from 32.99% to 67.08%.

Table 5 Results of the multiple feature approach

Method	Accuracy/%
Baseline	32.99
Levenshtein distance	63.97
SC_{add} + word2vec	64.23
Multiple feature approach	**67.08**

To estimate the contribution of each feature to the performance of the multiple feature approach, we make ablation tests by removing one kind of features for each time.

Table 6 shows the results of the ablation test. From Table 6, we can see that each kind of features contribute to the performance. While similarity features contribute to the performance more than others, ranking features, contextual features and string features still contribute to increasing performance. For Levenshtein distance similarity and semantic composition similarity, each of them has its own advantages, the performance will decrease without any of them.

Table 6 Results of the ablation test

Method	Accuracy/%
Multiple feature approach	**67.08**
Without ranking features	66.78
Without similarity features	64.18
Without contextual features	66.92
Without string features	66.80
Without Levenshtein distance similarity	66.21
Without semantic composition similarity	66.35

3.6 Analysis

Currently, several existing clinical NLP systems such as MetaMap [11] and cTAKES [13] can extract clinical concepts and map them to UMLS CUIs, and the performance are not very satisfactory. In this study, experimental results show that our multiple feature approach can improve the performance, but there are also some difficult situations for further improvement.

1) **"CUI-less" disorder mentions** Levenshtein distance method and semantic composition method will return the CUI with the highest similarity if candidate set retrieved by the findConcepts_NORMstrAPI is not null. It is obviously that the correct answer for the disorder mention may be "CUI-less", this process will give wrong CUI for this case. We find out that the ratio of the "CUI-less" mentions in the errors of SC_{add} + word2vec model is 33.97% and this is a major source of the errors.

The multiple feature approach can avoid the above problem in some ways. Given a disorder mention and its candidates retrieved by API, this approach decides which candidate is the correct CUI for the disorder mention. If the label of all the candidates is negative, the result of the disorder mention is "CUI-less". Table 7 shows the results on the "CUI-less" disorder mentions. Compared with Levenshtein distance method and semantic composition method, the multiple feature approach improves the accuracy on "CUI-less" disorder mentions from 59.86% to 78.52%. The ratio of the "CUI-less" disorder mentions in the errors of the multiple feature approach is 19.75% and it still needs further improvements.

Table 7 Results on the "CUI-less" mentions

Method	CUI-less accuracy/%
Levenshtein distance	59.86
SC_{add} + word2vec	59.86
Multiple feature approach	**78.52**

2) **Acronyms** As an example, for the disorder mention "CHI" in the training data, the correct answer is "C0085094, Closed head injuries". However, the candidates returned by the findConcepts_NORMstr API do not contain the answer. The findConcepts_NORMstr API cannot return relevant candidates for this case. The ratio of the acronyms mentions in the errors of the multiple feature approach is 18.77%. Contrasting this ratio with the ratio of acronyms mentions in the corpus (8.5%) shows that acronyms mentions are more difficult to recognize, and this is a major source of error.

3) **One-character disorder mention** Although the ratio of the disorder mention which contains one character in the corpus is only 1.5%, it is difficult to map them to correct CUI. For example, the correct CUI for the disorder mention "c" in one document is "C0149651|Clubbing". But for another "c" in the same document, the correct CUI is "C0010520|Cyanosis". "C0439106|Upper case sea", "C0439128|Lower case sea" and "C1720692|Roman numeral C" are the top candidates of the disorder mention "c" retrieved by the findConcepts_NORMstr API, and it does not contain the correct CUI in the candidates.

Although many candidates can be retrieved by the

UMLS API, it may not return appropriate candidates for the one-character disorder mentions and there is not enough information for us to determine which candidate is correct for them. Even though the disorder mentions with different positions in one document have the same phrase froms, their corresponding CUI may be different.

40.8% of the one-character disorder mentions are annotated as "CUI-less" in the corpus, and they are labeled as "CUI-less" in our experiment. While this is a limitation of our method, we notice that the one-character disorder mentions are only present in 2.7% of the errors and conclude that this is not a primary source of error in this corpus. The disorder mention itself does not provide enough information for the normalization task, especially for the acronyms and one-character disorder mentions. More contextual information should be considered to help with the normalization of disorder mentions.

4) **Disjoint disorder mentions** The ratio of the disjoint mentions in the errors of the multiple feature approach is 11.99%. It seems reasonable that disjoint mentions are more difficult to normalize than the contiguous mentions, comparing with their ratio in the corpus (9.76%). Compared to the contiguous mentions, we do not conduct special process for the disjoint mentions. This is a major source of the errors and further improvement is needed for the disjoint mentions.

5) **Dependence on API** The retrieved step is implemented by the API provided by UMLS. If the candidates retrieved by the API do not contain the correct answer, our method cannot choose the correct answer for the disorder mention. For example, the API cannot provide appropriate candidates for the one-character disorder mentions, because of their ambiguity.

4 Conclusion

In this paper, we exploit a multiple feature approach based on both Levenshtein distance method and semantic composition method. In this approach, multiple features (string features, ranking features, similarity features, and contextual features) are used for the normalization task. The multiple feature approach can integrate the advantages of both Levenshtein distance method and semantic composition method. The ablation test shows each kind of features contributes to improving the accuracy of the normalization task. Compared with the baseline, our approach improves the accuracy of the normalization task by 34.09%.

References

[1] Doğan R I, Leaman R, Lu Z. NCBI disease corpus: A resource for disease name recognition and concept normalization [J]. *Journal of Biomedical Informatics*, 2014, **47**: 1-10.

[2] Wei X, Huang S, Chen B, *et al*. BioTSA: Annotating token semantic association to support biomedical text mining [J]. *Wuhan University Journal of Natural Sciences*, 2015, **20**(2): 134-140.

[3] Zhou J, Lü C, Ji D, *et al*. Framework construction and application for global health information platform [J]. *Wuhan University Journal of Natural Sciences*, 2015, **20**(2): 153-158.

[4] Bodenreider O. The unified medical language system (UMLS): Integrating biomedical terminology [J]. *Nucleic Acids Research*, 2004, **32**(suppl 1): 267-270.

[5] Pradhan S, Elhadad N, South B R, *et al*. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative [J]. *Journal of the American Medical Informatics Association*, 2015, **22**(1): 143-154.

[6] Pradhan S, Elhadad N, Chapman W, *et al*. Semeval-2014 task 7: Analysis of clinical text [C] // *Proceedings of the* 8*th International Workshop on Semantic Evaluation*. *Association for Computational Linguistics*, 2014: 54-62.

[7] Chen B, Lü C, Wei X, *et al*. Semantic relation annotation for biomedical text mining based on recursive directed graph [J]. *Wuhan University Journal of Natural Sciences*, 2015, **20**(2):141-145.

[8] Liu M, Jiang L, Hu H. Automatic extraction and visualization of semantic relations between medical entities from medicine instructions [EB/OL]. [2016-03-20]. *http://link.springer.com/article/*10.1007/*s*11042-015-3094-4.

[9] Liu M, Zhang H, Hu H, *et al*. Topic categorization and representation of health community generated data [EB/OL]. [2016-03-20]. *http://link.springer.com/article/*10.1007/*s*11042-015-3094-3.

[10] Friedman C. A broad-coverage natural language processing system [C] // *Proceedings of the AMIA Symposium*. Los Angeles: American Medical Informatics Association, 2000: 270-274.

[11] Aronson A R, Lang F M. An overview of MetaMap: Historical perspective and recent advances [J]. *Journal of the American Medical Informatics Association*, 2010, **17**(3): 229-236.

[12] Denny J C, Irani P R, Wehbe F H, *et al*. The KnowledgeMap project: Development of a concept-based medical school curriculum database [C] // *AMIA Annual Symposium Proceedings*. Washington D C: American Medical Informatics

Association, 2003:195-199.

[13] Savova G K, Masanz J J, Ogren P V, *et al*. Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications [J]. *Journal of the American Medical Informatics Association*, 2010, **17**(5): 507-513.

[14] Zeng Q T, Goryachev S, Weiss S, *et al*. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: Evaluation of a natural language processing system [J]. *BMC Medical Informatics and Decision Making*, 2006, **6**(1): 1-9.

[15] Garla V N, Brandt C. Knowledge-based biomedical word sense disambiguation: An evaluation and application to clinical document classification [J]. *Journal of the American Medical Informatics Association*, 2013, **20**(5): 882-886.

[16] Blacoe W, Lapata M. A comparison of vector-based representations for semantic composition [C] // *Proceedings of the* 2012 *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju: Association for Computational Linguistics, 2012: 546-556.

[17] Lü C, Lu Y, Ji D, *et al*. Deep learning for textual entailment recognition [C] // *Tools with Artificial Intelligence* (*ICTAI*), 2015 *IEEE* 27*th International Conference on*. Salerno: IEEE Press, 2015: 154-161.

[18] Levenshtein V I. Binary codes capable of correcting deletions, insertions, and reversals [J]. *Soviet Physics Doklady*, 1966, **10**(8):707-710.

[19] Bengio Y, Ducharme R, Vincent P, *et al*. A neural probabilistic language model [J]. *Journal of Machine Learning Research*, 2003, **3**: 1137-1155.

[20] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning [C] // *Proceedings of the* 25*th International Conference on Machine Learning*. Helsinki: ACM Press, 2008: 160-167.

[21] Huang E H, Socher R, Manning C D, *et al*. Improving word representations via global context and multiple word prototypes [C] // *Proceedings of the* 50*th Annual Meeting of the Association for Computational Linguistics*: *Long Papers-Volume* 1. Jeju: Association for Computational Linguistics, 2012: 873-882.

[22] Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space [EB/OL]. [2016-03-20]. *http://arxiv.org/pdf/*1301.3781.*pdf*.

[23] Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality [C] // *Advances in Neural Information Processing Systems*. Lake Tahoe: Curran Associates Inc, 2013: 3111-3119.

[24] Baroni M, Zamparelli R. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space [C] // *Proceedings of the* 2010 *Conference on Empirical Methods in Natural Language Processing*. Cambridge: Association for Computational Linguistics, 2010: 1183-1193.

[25] Mitchell J, Lapata M. Composition in distributional models of semantics [J]. *Cognitive Science*, 2010, **34**(8): 1388-1429.

[26] Socher R, Huval B, Manning C D, *et al*. Semantic compositionality through recursive matrix-vector spaces [C] // *Proceedings of the* 2012 *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju: Association for Computational Linguistics, 2012: 1201-1211.

□