Computational Biology and Chemistry 32 (2008) 287–291

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem

Brief Communication

Exploiting the performance of dictionary-based bio-entity name recognition in biomedical literature

Zhihao Yang*, Hongfei Lin, Yanpeng Li

Department of Computer Science and Engineering, Dalian University of Technology, 116023 Dalian, China

ARTICLE INFO

Article history:
Received 8 March 2007
Received in revised form 20 March 2008
Accepted 24 March 2008

Keywords:
Text mining
Entity recognition
Edit distance
Conditional random fields

ABSTRACT

Bio-entity name recognition is the key step for information extraction from biomedical literature. This paper presents a dictionary-based bio-entity name recognition approach. The approach expands the bio-entity name dictionary via the Abbreviation Definitions identifying algorithm, improves the recall rate through the improved edit distance algorithm and adopts some post-processing methods including Pre-keyword and Post-keyword expansion, Part of Speech expansion, merge of adjacent bio-entity names and the exploitation of the contextual cues to further improve the performance. Experiment results show that with this approach even an internal dictionary-based system could achieve a fairly good performance.

1. Introduction

Along with the rapid expansion of biomedical literature, the demand for efficiently extracting biomedical information from the huge amount of literature resources offers an excellent opportunity for biomedical text mining. Among others, extracting relationship between bio-entities such as protein, gene and virus from biomedical literature has become a research focus. To accomplish it, the fundamental task is named entity recognition (NER), which is the identification of text terms referring to items of interest. In biomedical domain, named entities (called bio-entities) include protein, DNA, RNA, virus, etc.

NER is not a new task in text mining. In previous research work, many NER systems have been applied successfully in the newswire domain. But in biomedical domain it remains a challenging task due to the irregularities and ambiguities in gene and protein nomenclature. The irregularities and ambiguities are mainly the result of a lack of naming conventions, as well as the widespread practice of using many synonyms for one gene or protein. For compound bio-entity names, there is additional requirement of determining their boundary. These factors make NER in the biomedical domain difficult. The best system in JNLPBA2004 (Kim et al., 2004) achieved an *F*-score of 72.6% on GENIA corpus (Kim et al., 2003); in BioCreative 2004 task 1A (Hirschman et al., 2005) the best system (Finkel et al., 2005) obtained an *F*-score of 83.2% using relax matching and this score reduced to 74.3% using exact matching (Tsai et al., 2006). These results show the performance of NER in the biomedical domain is far below the one of NER in the general domain.

The most popular techniques in biomedical NER are machine learning techniques which include HMM (Zhou and Su, 2004), MEMM (Finkel et al., 2004), CRFs (Settles, 2004), etc. Machine learning techniques can identify potential bio-entities which are not previously included in standard dictionaries. However, one drawback of these machine learning based approaches is that they do not provide identification information of recognized terms. For the purpose of information extraction of bio-entities interaction, the ID information of recognized bio-entities, such as GenBank ID or SwissProt ID, is indispensable to integrate the extracted information with the data in other information sources.

Dictionary-based approaches can intrinsically provide ID information since they recognize a term by searching the most similar (or identical) one in the dictionary to the target term. This advantage makes dictionary-based approaches particularly useful as the first step for practical information extraction from biomedical literature. Tsuruoka and Tsujii (2003) tagged proteins in GENIA 3.01 with a combination of dictionary and Naive Bayes Classifier, achieving an *F*-score of 66.6%. Cohen (2005) achieved an *F*-score of 75.6% in gene and protein NER on GENIA 3.02 corpus through building the dictionaries from online genomics resources.

However, the performance of dictionary-based approaches depend badly on the size and quality of the dictionary while it is difficult to create a complete dictionary since new bio-entity names continue to be created and there are often many variations in the way identical bio-entities are referred to. Fortunately, there is an enormous amount of manual curation activity related to gene

* Corresponding author. Tel.: +86 41184706009x3926; fax: +86 411 84708116.
E-mail address: yangzh@dlut.edu.cn (Z. Yang).

1476-9271/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2008.03.008

and protein function. Several genomics databases contain large amounts of curated gene and protein name symbols as well as full names. Groups such as the Human Genome Organization (HUGO), Mouse Genome Institute (MGI), UniProt, and the National Center for Biotechnology Information (NCBI) collect and organize information on gene and proteins including gene names, symbols, and synonyms. A dictionary can be composed by via of these curated genomics databases and can automatically incorporate additional new names and symbols as the databases are updated by the curating organization.

There are also some solutions to the spelling variation problem, which is a common phenomenon in biomedical literature. For example, the protein name "NF-Kappa B" has many spelling variants such as "NF Kappa B," "NF kappa B," "NF kappaB," and "NFkappaB." Exact matching techniques, however, regard these terms as completely different terms. This problem can be alleviated by using approximate string matching methods (such as the edit distance algorithm described later in this paper) in which surface-level similarities between terms are considered.

Our work aims to exploit the performance of the dictionary-based bio-entity name recognition. This paper presents a dictionary-based approach, which expands the bio-entity name dictionary via the Abbreviation Definitions identifying algorithm and improves the recall rate through the improved edit distance algorithm. Some post-processing methods are also applied including Pre-keyword and Post-keyword expansion, Part of Speech (POS) expansion, merge of adjacent bio-entity names and the exploitation of the contextual cue.

The remaining part of this paper is organized as follows: Section 2 describes our methods. Section 3 presents the experiment results using the JNLPBA2004 dataset. Section 4 summarizes the annotation error causes. Finally, Section 5 offers some concluding remarks.

2. Methods

Our approach includes three processing steps: the construction and expansion of the bio-entity name dictionary, the approximate string matching and the post-processing using methods including Pre-keyword and Post-keyword expansion, POS expansion, merge of adjacent bio-entity names and exploitation of contextual cues. The details are described in the following sections.

2.1. Construction and Expansion of the Bio-entity Name Dictionary

The bio-entity name dictionary used in our approach is an internal dictionary which is constructed through extracting the annotated bio-entity names from the training set in JNLPBA2004 (2000 MEDLINE abstracts). After filtering out some noise entries like "protein", "DNA", 17 726 entries are left in the dictionary.

The size of the dictionary is crucial to the performance of the dictionary-based method. In order to expand the dictionary we adopted a full name-abbreviation pair expansion method. There are many bio-entity full name-abbreviation pairs in biomedical literature such as "Toll-like receptor 2 (TLR2)" and "NF-Y-associated factors (YAFs)". We found 3252 such pairs appeared in the training set, among which there are 2260 pairs are annotated, accounting for about 69.5%. Therefore it is meaningful to extract the full name-abbreviation pairs from the test set to expand the dictionary.

Generally, there are two patterns of full name-abbreviation pair: "expanded form (abbreviation)" and "abbreviation (expanded form)". We used an algorithm similar to Schwartz and Hearst (2003) to extract these full name-abbreviation pairs from the test set and got 654 pairs. To filter out the false positives among them, we introduced a CRFs model.

CRFs are undirected statistical graphical models, a special case of which is a linear chain that corresponds to a conditionally trained finite-state machine. Such models are well suited to sequence analysis. Let $o = \langle o_1, o_2, \ldots, o_n \rangle$ be a sequence of observed words of length n. Let S be a set of states in a finite state machine, each corresponding to a label $\in L$. Let $s = \langle s_1, s_2, \ldots, s_n \rangle$ be the sequence of states in S that correspond to the labels assigned to words in the input sequence o. Linear chain CRFs define the conditional probability of a state sequence given an input sequence to be:

$$P(s|o) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_k f_k(s_{i-1}, s_i, o, i) \right) \quad (1)$$

where Z is a normalization factor of all state sequences, $f_k(s_{i-1}, s_i, o, i)$ is one of m functions that describes a feature, and λ_k is a learned weight for each such feature function. CRFs are presented in more complete detail by Lafferty et al. (2001).

Eight features are chosen in our CRFs model: Surface Word Features, Orthographic Features, Prefix/Suffix Features, Word Shape Features, Compound Features, Part-of-Speech Features, Keyword Features, and Boundary Word Features. A quasi-Newton method called L-BFGS is used to find these feature weights. Trained on the training set of JNLPBA2004 our CRFs model achieved an F-score of 71.87% on a test set of 404 records.

Through our CRFs model filtering, 450 full name-abbreviation pairs are left, achieving a precision of 81.6% and a recall of 86.4%. The abbreviation is classified as the same class of the expanded form if the expanded form is found in the dictionary and vice versa. Otherwise, they are assigned the class which the CRFs model assigns them.

2.2. Approximate String Matching

After the dictionary is constructed and expanded, it can be used to identify the bio-entity names in the test set. The most straightforward way to exploit a dictionary for candidate recognition is the exact (longest) matching algorithm. However, the existence of many spelling variations for the same bio-entity name makes the exact matching less attractive. For example, even a short protein name "EGR-1" has at least six variations: "EGR-1", "EGR 1", "Egr-1", "Egr 1", "egr-1", "egr 1". Since longer protein names have a huge number of possible variations, it is impossible to enrich the dictionary by expanding each protein name as described above.

To tackle the problem of spelling variation, we employed the edit distance algorithm, the most popular measure of similarity between two strings (Navarro, 2001). The edit distance algorithm calculates the minimum number of operations on individual characters (e.g. substitutions, insertions, and deletions) required to transform one string of symbols into another. The calculation of edit distance is accomplished by via of a matrix $C_{0\ldots|x|;0\ldots|y|}$, where $C_{i,j}$ represents the minimum number of operations needed to match $x_{1\ldots i}$ to $y_{1\ldots j}$. This is computed as follows.

$$C_{i,0} = i \quad (2)$$

$$C_{0,j} = j \quad (3)$$

$$C_{i,j} = \text{if}(x_i = y_j)\,\text{then}\,C_{i-1,j-1} \text{ else } 1 + \min(C_{i-1,j}, C_{i,j-1}, C_{i-1,j-1}) \quad (4)$$

An example of edit distance computation is illustrated in Table 1. The edit distance between "IL-2" and "IL 2" is one.

To take into account the length of a bio-entity name, we adopt a normalized cost, which is calculated by dividing the cost by the