Journal of Biomedical Informatics 56 (2015) 94–102

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

Identifying synonymy between relational phrases using word embeddings

Nhung T.H. Nguyen [a,b,*,1], Makoto Miwa [c], Yoshimasa Tsuruoka [d], Satoshi Tojo [b]

[a] University of Science, Vietnam National University, Ho Chi Minh City, 227 Nguyen Van Cu St., Ward 4, Dist. 5, Ho Chi Minh City, Viet Nam
[b] Japan Advanced Institute of Science and Technology, 1-8 Asahidai, Nomi-shi, Ishikawa 923-1292, Japan
[c] Toyota Technological Institute, 2-12-1 Hisakata, Tempaku-ku, Nagoya 468-8511, Japan
[d] The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

ARTICLE INFO

Article history:
Received 7 October 2014
Revised 12 May 2015
Accepted 15 May 2015
Available online 22 May 2015

Keywords:
Word embeddings
Synonym resolution
Relational phrase clustering
Topic modeling

ABSTRACT

Many text mining applications in the biomedical domain benefit from automatic clustering of relational phrases into synonymous groups, since it alleviates the problem of spurious mismatches caused by the diversity of natural language expressions. Most of the previous work that has addressed this task of synonymy resolution uses similarity metrics between relational phrases based on textual strings or dependency paths, which, for the most part, ignore the context around the relations. To overcome this shortcoming, we employ a word embedding technique to encode relational phrases. We then apply the *k*-means algorithm on top of the distributional representations to cluster the phrases. Our experimental results show that this approach outperforms state-of-the-art statistical models including latent Dirichlet allocation and Markov logic networks.

1. Introduction

Many of the robust text mining systems in the biomedical domain allow end-users to browse and retrieve information from their databases [1–3]. Implementing such retrieval functionality is usually not so difficult if the system is only concerned with a specific type of information, such as protein–protein interaction and gene-disease association, since they can apply some matching techniques to the input entities to extract the answers. However, the problem becomes much more difficult when the system is designed to cover unrestricted types of relations, which requires the relation in a query to be specified using a natural language expression, such as ‘be induced by’ or ‘result in’. Such relational phrases expressed in natural language often cause spurious mismatches between the user’s query and the textual data in the underlining database. For example, given the input query “What genes are essential for cell survival?”, the system can fail to return the result <stat1, *be critical for*, cell survival> due to the string-level mismatch between *be essential for* and *be critical for*. In most situations, *be essential for* is equivalent to *be critical for*, i.e., they form a pair of synonyms, which can be used for alleviating the mismatch problem. Therefore, the major objective of this work is to identify synonymy between relational phrases in biomedical relations, which should be beneficial for many text mining applications in the domain, such as question answering, event extraction, and entailment detection [4,5].

Identifying synonymy between relational phrases can be seen as clustering synonymous phrases that represent identical or similar relationships between entities. Since this task is performed on top of a relation extraction system, the performance of clustering can be affected by the performance of the extraction system. Another difficulty of the task is the polysemy of natural language, i.e., a relational phrase can have multiple senses. This problem could be addressed by using a soft clustering approach, but we leave it for future work and assume that a relation phrase belongs to a single cluster.

Previous work that tackled this task employed similarity metrics based on textual strings [6] or dependency paths [7–9] of the two relational phrases. Kok and Domingos [10] proposed a probabilistic model based on two Markov logic networks (MLNs) [11] to simultaneously cluster objects and relations. Nebot and Berlanga [12] used a probabilistic model inspired by statistical machine translation to cluster relations in biomedical documents. These models are unsupervised in the sense that no manual

[*] Corresponding author at: University of Science, Vietnam National University, Ho Chi Minh City, 227 Nguyen Van Cu St., Ward 4, Dist. 5, Ho Chi Minh City, Viet Nam.

E-mail addresses: nthnhung@jaist.ac.jp (N.T.H. Nguyen), makoto-miwa@toyota-ti.ac.jp (M. Miwa), tsuruoka@logos.t.u-tokyo.ac.jp (Y. Tsuruoka), tojo@jaist.ac.jp (S. Tojo).

[1] This work was carried out while the first author was a doctoral student at JAIST.

http://dx.doi.org/10.1016/j.jbi.2015.05.010
1532-0464/© 2015 Elsevier Inc. All rights reserved.

labeling of clusters by human is needed. One of the major shortcomings of their approaches, however, is that they only focus on the textual surface of arguments of a relation to estimate the synonymy probability and cannot effectively capture other features, such as the context around the relations.

To address the above shortcoming, we apply the continuous bag-of-words (CBOW) model, a deep-learning technique proposed by Mikolov et al. [13], to represent our relational phrases. A relation in the format of <entity 1, relational phrase, entity 2> is identified in a sentence, and each of the two entities and relational phrase is regarded as a newly defined *word*. We thus treat the entities and the phrase differently from the other words depending on their corresponding roles in the relation. The CBOW model then learns the distributional representations of the relational phrases through a feed-forward neural network language model [14], which allows us to capture the context around a relational phrase when learning its representation.

Sun and Korhonen [15] also used the context around verbs for the task of verb classification by introducing a rich set of semantic features. The features include collocations of verbs, prepositional preference, and lexical preference in subject, object and indirect object relations. The key difference between their work and ours is that we cluster verbs and verb phrases that compose biomedical relations while they only focus on single verbs.

We have compared our approach with three unsupervised methods: bag-of-words (BOW), latent Dirichlet allocation (LDA) [16], and Semantic Network Extractor (SNE) [10]. Regarding BOW and LDA, we treat a relational phrase as a *document* (in LDA terms) and entities that share the same phrase as *words* in the document. The BOW model represents each relational phrase as a sparse vector of occurrence counts of entities. LDA-SP [17], which is developed from LinkLDA [18] to model selectional preferences, simultaneously models two sets of distributions for two entities of a relation. Each entity is drawn from a hidden topic. LDA-SP assigns a higher probability to the state in which the two hidden topics are equal. For each relational phrase, the model outputs a vector of the prior topic distribution. We then apply the *k*-means algorithm on top of vector representations to cluster phrases into synonymous groups.

SNE tackles the task of clustering relational phrases by a probabilistic model trained on two MLNs. Unlike the other methods, SNE performs clustering on a database of relations, i.e., it does not consider the context or the frequency of relations. However, SNE can automatically identify the best number of clusters and simultaneously cluster objects and relational phrases.

We have conducted experiments using a large set of biomedical relations extracted from MEDLINE by PASMED, a pattern-based open information extraction (Open IE) system [19,20]. The results show that word embeddings significantly outperform BOW, LDA-SP and SNE. They can boost the performance of clustering by 9% of F-score compared with the other methods. In addition, we demonstrate how the obtained clusters of relational phrases could be used to improve the performance of high-level text-mining applications such as question answering and entailment detection.

The main contribution of this article is that we have applied LDA-SP and CBOW models to the task of identifying synonymy between relational phrases. For the CBOW model, we have introduced a simple but effective representation of relations. Because the representation can exploit various information relevant to relations, e.g., the textual surface of the two entities, the context around a relation in its sentence, and the corresponding role of each component in a relation, the performance of CBOW is boosted significantly.

2. Clustering relational phrases

We first encode our relational phrases into vector format by using three different unsupervised techniques: bag-of-words, topic model and word embeddings. Next, we apply the *k*-means algorithm on top of these vector representations to cluster relational phrases into synonymous groups. In addition to vector representations, we have also employed SNE, a Markov logic network-based system, to identify synonymous relational phrases. An overview of our working flow is shown in Fig. 1.

2.1. Word embeddings

Mikolov et al. [13] introduced two effective techniques for learning vector representations of words from large amounts of unstructured text data: the Continuous Bag-Of-Word (CBOW) model and the continuous Skip-gram model.

The CBOW model is similar to the feed-forward neural network language model [14], where there is no hidden layer and the projection layer is shared for all words. Unlike the BOW model, this model predicts a word by using the continuous context around that word. Given a sequence of training words $w_1, w_2, w_3 \ldots w_T$, the objective of this model [21] is to maximize the average log probability as shown in Eq. (1), where C_t is words in the context of w_t within a window size of c, $C_t = w_{t-c}, w_{t-c+1} \cdots w_{t-1}, w_{t+1} \cdots w_{t+c}$.

$$\frac{1}{T}\sum_{t=1}^{T} \log p(w_t|C_t) \tag{1}$$

The probability of $p(w_t|C_t)$ is estimated by using the softmax function:

$$p(w_t|C_t) = \frac{\exp\left(v_t'^{\top} v_{C_t}\right)}{\sum_{i=1}^{V} \exp\left(v_i'^{\top} v_{C_t}\right)} \tag{2}$$

where v and v' are the *input* and *output* vector representation of a word w, and V is the number of words in the vocabulary. In contrast with the CBOW model, the Skip-gram model receives the current word and predicts words within a certain window.

Recently, distributed representations have been shown to effectively improve the performance of many NLP tasks such as paraphrase detection [22], sentiment prediction [23], semantic relation classification [24], word alignment [25], entity mention tagging [26], and machine translation [27–29].

In this paper, we use the CBOW model[2] to estimate vector representations of our relational phrases. More specifically, for each relation in the format of <entity 1, relational phrase, entity 2>, which is given by an Open IE system, we retrieve the sentence that contains the relation from the original text database. We then identify the words or phrases that correspond to the entities and relational phrases, and create newly-defined *words* for them depending on their roles in the relation.

We introduce three different representations of a relation:

(i) *Relation*: treating a relation as a sentence, this representation uses the same information as BOW, LDA-SP, and SNE.
(ii) *Sentence*: embedding the relation in the sentence in which it appears and assigning a role to the relational phrase.
(iii) *Role*: embedding the relation in the sentence in which it appears and assigning corresponding roles to the relational phrase and its two entities.

For example, a relation of <parkinson's disease, treat with, dopaminergic drug> will be represented in three ways shown in

[2] The model is implemented in the word2vec tool: http://code.google.com/p/word2vec/.