Knowledge-Based Systems 106 (2016) 74–84

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Near-synonym substitution using a discriminative vector space model

Liang-Chih Yu[a,b,*], Lung-Hao Lee[c], Jui-Feng Yeh[d], Hsiu-Min Shih[e], Yu-Ling Lai[e]

[a] Department of Information Management, Yuan Ze University, 135 Yuan-Tung Road, Taoyuan 32003, Taiwan
[b] Innovative Center for Big Data and Digital Convergence, Yuan Ze University, Taiwan
[c] Information Technology Center, National Taiwan Normal University, Taiwan
[d] Department of Computer Science and Information Engineering, National Chiayi University, Taiwan
[e] Department of Mathematics, National Chung Cheng University, Taiwan

ARTICLE INFO

Article history:
Received 12 September 2015
Revised 23 March 2016
Accepted 13 May 2016
Available online 20 May 2016

Keywords:
Natural language processing
Lexical substitution
Near-synonym learning
Discriminative training
Vector space model

ABSTRACT

Near-synonyms are fundamental and useful knowledge resources for computer-assisted language learning (CALL) applications. For example, in online language learning systems, learners may have a need to express a similar meaning using different words. However, it is usually difficult to choose suitable near-synonyms to fit a given context because the differences of near-synonyms are not easily grasped in practical use, especially for second language (L2) learners. Accordingly, it is worth developing algorithms to verify whether near-synonyms match given contexts. Such algorithms could be used in applications to assist L2 learners in discovering the collocational differences between near-synonyms. We propose a *discriminative vector space model* for the near-synonym substitution task, and consider this task as a classification task. There are two components: a *vector space model* and *discriminative training*. The vector space model is used as a baseline classifier to classify test examples into one of the near-synonyms in a given near-synonym set. A discriminative training technique is then employed to improve the vector space model by distinguishing positive and negative features for each near-synonym. Experimental results show that the DT-VSM achieves higher accuracy than both pointwise mutual information and n-gram-based methods that have been used in previous studies.

1. Introduction

Near-synonyms are words that are almost synonyms, representing a group of words with similar meanings [16]. They can be derived from manually constructed dictionaries such as WordNet [9], EuroWordNet [31], Chinese WordNet [14], and clusters derived using statistical approaches [4,22,36]. These knowledge resources have been widely investigated in many natural language applications such as information retrieval (IR) [2,25,32,34,43] and (near-) duplicate detection for text summarization [26,37]. In addition to the above applications, near-synonyms are also fundamental and useful knowledge resources for computer-assisted language learning (CALL) [15,28,39]. For example, in online language learning systems, learners can use near-synonyms to express similar meanings so as to enrich the content of their spoken and written language.

However, it is usually difficult to choose suitable near-synonyms to match a given context because the differences of near-synonyms are not easily grasped in practical use, especially for second language (L2) learners. Previous studies have provided examples to explain the context mismatch problem [15,29,42]. For instance, one should use "*strong* coffee" but not "*powerful* coffee" and use "ghastly *mistake*" but not "ghastly *error*". Accordingly, it is worth developing algorithms to verify whether near-synonyms do match given contexts. Such algorithms can be used in applications to provide more effective services [18,35]. For instance, a computer-assisted language learning system can assist L2 learners in discovering the collocational differences between near-synonyms, as well as suggest an alternative word that best fits a given context from a list of near-synonyms.

In measuring the substitutability of words, the co-occurrence information between a target word (the gap) and its context words is commonly used in statistical approaches. Edmonds [8] built a lexical co-occurrence network from the 1989 Wall Street Journal to determine near-synonyms that are most typical or expected in given contexts. Inkpen [15] used the pointwise mutual information (PMI) formula to select the best near-synonym that can fill the gap in a given context. The PMI scores for each candidate near-synonym are computed using a larger web corpus, the Waterloo terabyte corpus, which can alleviate the data sparseness problem encountered in Edmonds' approach. Gardiner and Dras [10] also used the PMI formula but with a different corpus (Web 1T 5-gram

* Corresponding author.
E-mail addresses: lcyu@saturn.yzu.edu.tw (L.-C. Yu), lhlee@ntnu.edu.tw (L.-H. Lee), ralph@mail.ncyu.edu.tw (J.-F. Yeh), culy33@yahoo.com.tw (H.-M. Shih), yllai@math.ccu.edu.tw (Y.-L. Lai).

http://dx.doi.org/10.1016/j.knosys.2016.05.025
0950-7051/

Table 1
Example of a near-synonym set and a sentence to be evaluated.

Sentence:	The _____ under the bay is closed because of an accident.
Original word:	tunnel
Near-synonym set:	{tunnel, bridge, overpass, viaduct}

corpus) to explore whether near-synonyms differ in terms of attitude.

In addition to PMI, *n*-gram modeling is another commonly used method. Yu et al. [42] computed a substitution score for each near-synonym based on *n*-gram frequencies obtained by querying Google. A statistical test is then applied to determine whether or not a target word can be substituted by its near-synonyms. The dataset used in their experiments is derived from the OntoNotes corpus [13,30], where each near-synonym set corresponds to a *sense pool* in OntoNotes. Islam and Inkpen [17] used a 5-gram language model with the Web 1T 5-gram corpus for near-synonym choice. The *n*-gram modeling approach which considers the frequency of contiguous words may suffer from the data sparseness problem due to insufficient training data, especially for high-order *n*-gram. Skip-gram modeling can reduce the impact of the data sparseness problem by retrieving the frequency of non-contiguous words using *wildcards* in the *n*-gram [11].

In addition to the aforementioned methods, another direction to the task of near-synonym substitution is to use dimension reduction techniques [38,44] or identify the senses of a target word and its near-synonyms using word sense disambiguation (WSD) to determine whether they were of the same sense [24,7].

In this paper, we consider the near-synonym substitution task as a classification task, and develop a *discriminative vector space model* (*DT-VSM*) to perform the classification task. The DT-VSM consists of two components: a *vector space model* (*VSM*) [1,6] and *discriminative training* (*DT*) [20,27,40,41]. The vector space model represents both near-synonyms and test examples as vectors, where each dimension represents a distinct word in the context of the near-synonyms. The classification is then performed to classify each test example into one of the near-synonyms in a given near-synonym set using the cosine measure. However, near-synonyms share more common context words (features) than semantically dissimilar words. Such similar contexts may decrease classifiers' ability to discriminate among near-synonyms. Therefore, we propose the use of a supervised discriminative training technique to improve the vector space model by distinguishing positive and negative features for each near-synonym. The basic idea of discriminative training herein is to adjust feature weights according to the minimum classification error (MCE) criterion. The features that contribute to discriminating among near-synonyms will receive a greater positive weight, whereas the noisy features will be penalized and might receive a negative weight. This re-weighting scheme helps increase the separation of the correct class against its competing classes, thus improves the classification performance. To sum up, the proposed DT-VSM model distinguishes between positive and negative features for near-synonyms and incorporates them into a vector space model.

The overall framework of the DT-VSM model is illustrated in Fig. 1. Given a near-synonym set and a sentence containing one of the near-synonyms, the near-synonym is deleted to form a gap in the sentence, as shown in Table 1. The given near-synonym set {bridge, overpass, tunnel, viaduct} represents the meaning of a physical structure that connects separate places by traversing an obstacle, and the near-synonym *tunnel* in the given sentence is deleted to form a gap. The DT-VSM is then applied to recommend near-synonyms that can fill the gap. Various applications can thus benefit from the recommended near-synonyms to provide more intelligent services such as difference discovery between near-synonyms and writing supports. In experiments, the proposed supervised DT-VSM is compared with three unsupervised methods based respectively on PMI [15,10], *n*-gram [42] and skip-grams [11]. Each method is evaluated by predicting an answer (best near-synonym) that can fill the gap. The possible candidates are all the near-synonyms (including the original word) in the given set. Ideally, the correct answers should be provided by human experts. However, such data is usually unavailable, especially for a large set of test examples. Therefore, we follow Inkpen's experiments to consider the original word as the correct answer. The proposed methods can then be evaluated by examining whether they can restore the original word by filling the gap with the best near-synonym.

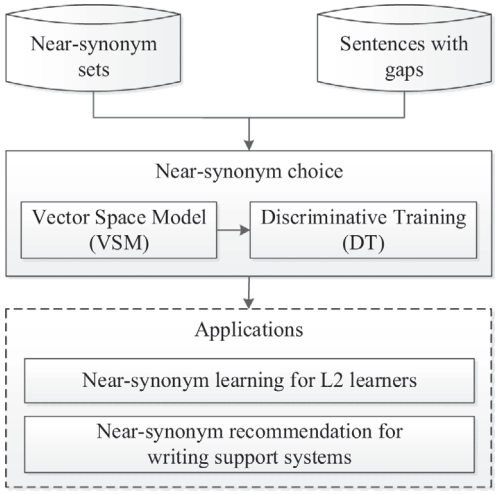

Fig. 1. Overall framework of the discriminative vector space model.

The rest of this work is organized as follows. Section 2 describes three previously proposed methods based respectively on PMI, *n*-gram and skip-grams, for near-synonym substitution. Section 3 presents a new method, called the discriminative vector space model. Section 4 summarizes comparative results. Conclusions are finally drawn in Section 5.

2. Unsupervised methods

This section introduces three unsupervised methods respectively based on PMI [15,10], *n*-gram [42] and skip-grams [11] for near-synonym substitution. These three methods are considered unsupervised because they measure the substitutability of words from unannotated corpora. The following subsections describe each method in turn.

2.1. PMI-based method

Mutual information can be used to measure the co-occurrence strength between a near-synonym and words used in a given context. A higher mutual information score indicates that the near-synonym fits well in the given context, and thus is more likely to be the correct answer. The pointwise mutual information [5,23] between two words x and y is defined as

$$PMI(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}, \tag{1}$$

where $P(x,y) = C(x,y)/N$ denotes the probability that x and y co-occur; $C(x,y)$ is the number of times x and y co-occur in the corpus, and N is the total number of words in the corpus. Similarly, $P(x) = C(x)/N$, where $C(x)$ is the number of times x occurs in the