Pattern Recognition Letters 88 (2017) 26–32

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems

Seonghan Ryu[a,*], Seokhwan Kim[b], Junhwi Choi[a], Hwanjo Yu[a], Gary Geunbae Lee[a]

[a] Pohang University of Science and Technology (POSTECH), 77 Cheongam-Ro, Nam-Gu, Pohang, 37673, Republic of Korea
[b] Institute for Infocomm Research (I2R), 1 Fusionopolis Way, # 21-01 Connexis (South Tower), 138632, Singapore

ARTICLE INFO

Article history:
Received 7 July 2016
Available online 13 January 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Natural language processing
Dialog systems
Out-of-domain sentence detection
Neural sentence embedding
Artificial neural networks
Distributional semantics

ABSTRACT

To ensure satisfactory user experience, dialog systems must be able to determine whether an input sentence is *in-domain* (ID) or *out-of-domain* (OOD). We assume that only ID sentences are available as training data because collecting enough OOD sentences in an unbiased way is a laborious and time-consuming job. This paper proposes a novel neural sentence embedding method that represents sentences in a low-dimensional continuous vector space that emphasizes aspects that distinguish ID cases from OOD cases. We first used a large set of unlabeled text to pre-train *word representations* that are used to initialize neural sentence embedding. Then we used domain-category analysis as an auxiliary task to train neural sentence embedding for OOD sentence detection. After the sentence representations were learned, we used them to train an autoencoder aimed at OOD sentence detection. We evaluated our method by experimentally comparing it to the state-of-the-art methods in an eight-domain dialog system; our proposed method achieved the highest accuracy in all tests.

1. Introduction

Dialog systems provide natural-language interfaces between humans and machines. Because human conversation can range among topics, many studies have been recently conducted on multi-domain dialog systems [5,8,14,24,27]. However, these systems are also restricted to a closed set of target domains and thus cannot provide appropriate responses to *out-of-domain* (OOD) requests. For example, a dialog system that was designed to cover `schedule` and `message` domains could receive OOD requests such as "*Would you recommend Italian restaurants for me?*" that is in the `restaurant` domain or "*Please record Game of Thrones.*" that is in the TV `program` domain. To maintain user experience, the system must detect OOD requests and provide appropriate back-off responses such as rejection, rather than providing unrelated responses.

The main goal of this paper is to develop an accurate *OOD sentence detection* method. We define OOD sentence detection as a *binary classification* problem of determining whether the system can respond appropriately to an input sentence, i.e.,

$$f(x) = \begin{cases} ID, & \text{if } x \text{ belongs to a domain } d \in D, \\ OOD, & \text{otherwise,} \end{cases} \quad (1)$$

where x is an input sentence, D is a closed set of target domain-categories such as `schedule` or `message`, *ID* denotes in-domain, and *OOD* denotes out-of-domain.

Most previous studies [19,31] use both ID sentences and OOD sentences to train OOD sentence detection. Collecting ID sentences is a necessary step in building many data-driven dialog systems. However, the task of collecting enough OOD sentences to cover all other domains is laborious and time-consuming. Therefore, the goal of this paper is to develop an accurate OOD sentence detection method that requires only ID sentences for training.

In this work, we present a novel *neural sentence embedding* method that represents sentences in a low-dimensional continuous vector space that emphasizes aspects that distinguish ID cases from OOD cases. First, we use large set of unlabeled text to pre-train *word representations* for the initialization of neural sentence embedding. Second, we use the similarity between OOD sentence detection and *domain-category analysis* [11,15,19,32] to train neural sentence embedding with only ID sentences.

Domain-category analysis is a task that assigns one of a closed set of target domains to a given sentence; this analysis system can

* Corresponding author.
E-mail addresses: ryush@postech.ac.kr (S. Ryu), kims@i2r.a-star.edu.sg (S. Kim), chasunee@postech.ac.kr (J. Choi), hwanjoyu@postech.ac.kr (H. Yu), gblee@postech.ac.kr (G.G. Lee).

http://dx.doi.org/10.1016/j.patrec.2017.01.008
0167-8655/

be trained using only ID sentences that are collected for each domain. We think that the task of OOD sentence detection is more similar to domain-category analysis than to other tasks such sentiment analysis or speech-act analysis, so we expect that the features (i.e., representation) of a sentence extracted by a domain-category analysis system can be used for OOD sentence detection too.

Therefore we adopt a feature extractor that is trained for domain-category analysis, and use it as a neural sentence embedding system for OOD sentence detection. Lastly, the learned representations of ID sentences are used to train an autoencoder that detects whether an input sentence is ID or OOD based on its reconstruction error. To the best of our knowledge, this is the first study that applies neural sentence embedding to solve the sentence representation problem of OOD sentence detection.

The remainder of this paper is organized as follows: In Section 2, we review previous studies. In Section 3, we describe our method in detail. In Section 4, we explain our experimental data, evaluation metrics, and methods to be compared. In Section 5, we show and discuss the experimental results. In Section 6, we conclude this paper.

2. Related work

Previous studies [12,19,31] on OOD sentence detection use sentence representations based on bag-of-words models, which have limitations in representing rare or unknown words; those words are likely to appear in OOD sentences. Lane et al. [12] proposed an in-domain verification (IDV) method, which uses only ID sentences to build domain-wise one-vs.-rest classifiers that generate low confidence scores for OOD sentences, and then uses the scores as evidence that a sentence was OOD. We implemented this method and compared it to our work. Nakano et al. [19] proposed an two-stage domain selection framework, which uses both ID sentences and OOD sentences to build multi-domain dialog systems; the main contribution is to use discourse information to prevent erroneous domain switching, but whenever developers expand the domain of a dialog system they must reassess all OOD sentences because some will become ID due to the change of the boundary between ID and OOD. Tur et al. [31] used syntactic feature and semantic feature for OOD sentence detection; web search queries are used as OOD sentences to eliminate the need to collect OOD sentences, but such queries are *noisy* because some are actually ID, and they cannot be obtained readily without using a commercial search engine. Compared to these studies, the main contribution of this paper is a neural sentence embedding method that can understand rare words and unknown words.

Recently, neural sentence embedding methods have been assessed for their ability to solve the sentence representation problem. Paragraph Vector [13] is a well-known method that uses a large set of unlabeled text to learn sentence representations, but the representations are not optimized for a specific task because they are learned based on *unsupervised* objectives. In contrast, some researchers have worked on *supervised* sentence embedding particularly for natural language understanding using recurrent neural networks (RNNs) [16,33,36] and long short-term memory (LSTM) networks [5,22,35]. However, because we cannot define an objective function based on classification error between ID cases and OOD cases, these methods are not directly applicable to our problem in which only ID sentences are available as a training set. To solve this problem, we exploit the similarity between OOD sentence detection and domain-category analysis (Section 1).

Another important part of OOD sentence detection is one-class classification that uses the training data about a target class to distinguish between target items and uninteresting items. Nearest Neighbor Distances (NN-d) [29] classifies an input item as the target class when the local density[1] of the item is larger than the local density of its closest item. A one-class support vector machine (OSVM) [25] learn a decision function about distinguishment. In this work, we propose to use an autoencoder to detect OOD sentences, and compare the results to those obtained using other methods including NN-d and OSVM.

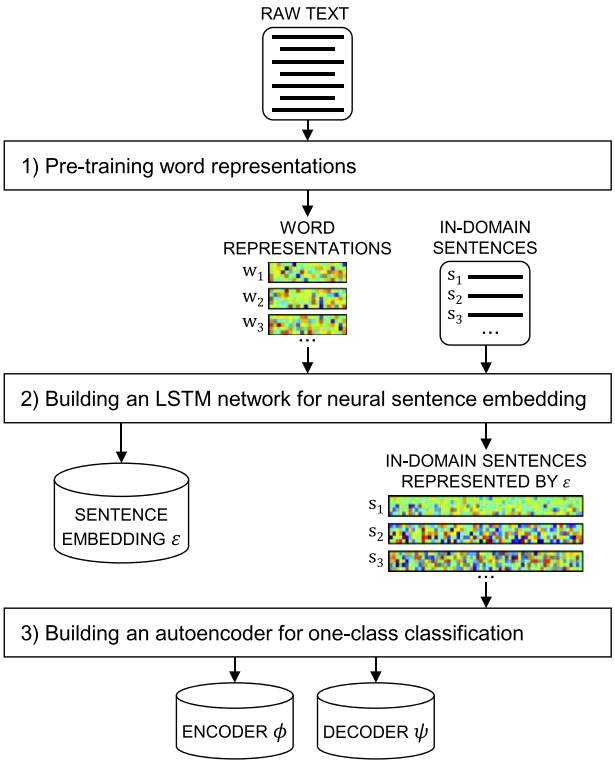

Fig. 1. Overall training process of our proposing method. Components and processes are described in the text.

3. The proposed OOD-sentence detection method

We defined OOD sentence detection $f(x)$ as a binary classification problem (Section 1). However, unlike most other binary classification problems, we assume that only ID sentences are available as training data. With these ID sentences, domain-category analysis $g(x) = d \in D$ can be built under another assumption that the domain category for each ID sentence is given.

When we represent sentences in m-dimensional continuous vector space, we take sentence embedding $\varepsilon(x) \in \mathbb{R}^m$ from an LSTM network trained with $g(x)$ as the supervised objective. Then, we build an autoencoder that consists of an *encoder* ϕ that takes sentences represented by $\varepsilon(x)$ and maps them onto a different space, and a *decoder* ψ that reconstructs their original representations. Finally, we use the learned autoencoder (ϕ, ψ) to detect OOD sentences of which reconstruction errors are greater a threshold θ as:

$$f(x) = \begin{cases} ID, & \text{if} \|\psi(\phi(\varepsilon(x))) - \varepsilon(x)\|^2 < \theta, \\ OOD, & \text{otherwise.} \end{cases} \quad (2)$$

The details of the proposed method (Fig. 1) are presented in the remainder of this section.

[1] The local density of an item is the distance between the item and its closest item in the training data.