Question Answering Based on Distributional Semantics

Vladislav Maraev

Department of Informatics, Faculty of Sciences, University of Lisbon, Portugal
vlad.maraev@di.fc.ul.pt

Abstract. An NLP application for question answering provides an insight into computer's understanding of human language. Many areas of NLP have recently built on deep learning and distributional semantic representation. This paper seeks to apply distributional semantic models and convolutional neural networks to the question answering task.

Keywords: question answering, word embeddings, convolutional neural networks, distributional semantics

1 Introduction

In cognitive science there was a long debate regarding how semantic knowledge is organized and used in language understanding and production. Known accounts of semantic representation can be distinguished as three broad families, namely semantic networks, feature-based models and semantic spaces.

Semantic networks [3] reproduce concepts as nodes in a graph whose edges denote semantic relationships between the concepts (e.g. SALMON IS-A FISH). In this case, word meanings can be obtained by collecting relations from the graph. Semantic networks are representations that abstract away from real-world usage as they are constructed manually by subjective modelers.

As an alternative to the network approach, the meaning of a word can be described in terms of *feature lists* [8], that can be obtained by polling native speakers about what features they consider as valuable for the meaning of a word. This approach is limited due to the size of vocabulary and also the number and quality of attributes are highly dependent on time devoted to each word.

A third family of semantic representations is based on the assumption that word meanings are determined by linguistic environment [9]. Words with similar meanings occur in similar contexts and one can talk about the common semantic value of expressions as word co-occurrence among them [6]. *Distributional models* are typically implemented through vector space models, where the semantic representation for a word is a vector in a high-dimensional space. The dimensions stand for context items (for example, co-occurring words), and the coordinates depend on the co-occurrence counts or probabilities. Distributional models can be learned from a corpus in unsupervised fashion. Similarity between words is usually computed using cosine similarity between the respective vectors.

2 Related Work

A widely applied model that learns vector representations by using recurrent neural network (RNN) was proposed by Mikolov et al. [7]. RNN is trained with back-propagation that adjusts the word vectors by walking through huge corpus of texts. This model permits interesting results on semantic and syntactic tests. Test sets were composed of analogy questions of the form "*a* is to *b* as *c* is to _?" aiming at testing different syntactic and semantic relations, such as "*see* is to *saw* as *return* is to *returned*" and "*clothing* is to *shirt* as *dish* is to *bowl*".

When one steps into the domain of sentence semantics, one faces some open problems, namely, the scaling problem, that poses the question on how fixed-length vectors can provide a representation for sentences of arbitrary length and structure in order to obtain fine-grained sentence similarity. Another problem is about representing function words, such as "not" and how they should be encoded in the vector space to adjust the meaning of a constituent [5].

Distributional semantic model can be applied to various natural language processing tasks such as question answering. In my work the results of the paper [2] are planned to be replicated. Authors of the paper are applying distributional semantics approach to the problem of question answering and use neural networks and distributional word representations to detect semantically equivalent questions in online user forums. Authors of the paper [2] define questions as *semantically equivalent* if they can be adequately answered by the exact same answer. In their study they have a goal to predict if two questions are semantically equivalent.

It's important to make a distinction between different tasks related to identification of semantically equivalent questions. Duplicate and near-duplicate sentences, paraphrases and textually similar sentences can be semantically equivalent but the reverse is not true.

Here is a brief description of convolutional neural network (CNN) architecture used in the paper. The input for the network is tokenized text strings of the two questions. In the first step, the CNN transforms words into real-valued feature vectors (word embeddings). Next, a convolutional layer is used to construct two distributed vector representations r_{q1} and r_{q2}, one for each input question. Finally, the CNN computes a similarity score between r_{q1} and r_{q2}. Pairs of questions with similarity above a threshold are considered duplicates. CNN was trained by minimizing the mean-squared error over the training set D. Question-answer pairs were obtained from AskUbuntu StackExchange forum.

The resulting model outperformed SVM (support vector machine) baseline with a significant margin: accuracy 92.9% for CNN vs 82.4% for SVM. And this margin was even more significant on the limited training set.

Another interesting result was obtained from the experiment that was aimed to figure out the impact of word embedding on the CNN accuracy. Firstly, it was found that word embedding dimensionality improves the performance of the model, when dimensionality was increased from 50 to 100 and from 100 to 200.

Secondly, authors reveal that congruent in-domain word embeddings improve performance of the model even when they have smaller amount of tokens. AskUbuntu corpus containing ≈121M of tokens outperformed Wikipedia corpus with ≈1.6B of tokens (acc. 92.4% vs 85.5%).

3 Plan and Current Progress

In my work it's planned to achieve similar results by replicating experiments made in [2]. The work will be divided in three big steps. First and second steps will consider the replication of results obtained by authors of [2]. On the first step I will construct the word embeddings. On the second step I will build a CNN with the word embeddings obtained on the first step as a first level of neural network. In the third step I will check how CNN could be helpful for non-English question answering namely for Portuguese and for Russian. More detailed description of the work is provided below.

For building the word embeddings I took slightly different data sets as the same data sets were not available at this time. I used AskUbuntu and Meta StackExchange dumps from September 2014 instead of the May 2014 version used in [2]. Corpora that I used for creating word embeddings contained 38 millions of tokens for AskUbuntu dump and 19 millions of tokens for META StackExchange dump. I used Deeplearning4j toolkit [4] for creating word embeddings. I have done basic tests in order to check adequacy of created word embeddings. Data sets include domain-specific questions, thus instead of classical test [7] "king − man + woman = queen" an example "like − likes + contain = contains" was picked. Test was passed for both of the domains.

For the step of building the CNN, I created a tool for randomized set generation of required number of questions: 20K pairs for training, 1K for validation and 4K for testing, where about 30% of each set are marked as duplicates. This tool will be used to make sets for training, testing and validation of CNN. The first level of the neural network will be constructed using domain-specific word embeddings described in the previous paragraph. I will use TensorFlow [1] to build neural network.

On the last step of work I will apply CNN and distributional semantic models to Portuguese and Russian question answering data sets. The Portuguese corpus was collected by selecting the data contained in the database of the PcWizard application, that offers IT hardware and software support by chat, through call-centre where all the interactions with the clients are saved. Only interactions composed by one question and the respective answer were included in the corpus. Resulting corpus is composed of 4000 question and answer pairs.

The Russian corpus was also collected by chat through call-centre of ISP and includes various questions about user account setup and billing. It contains total 24K question-answer pairs taken from dialogs and that should be manually reduced in order to create corpus that will only contain pairs that include one question and the respective answer.

4 Conclusions

The work is going to facilitate question answering by using convolutional neural network and distributional semantic representations. One focus of the work is the replication of the experiment [2], another focus considers applying reported architecture to Portuguese and Russian question answering corpora.

Acknowledgements

The results reported in this paper were partially supported by the Portuguese Government's P2020 program under the grant 08/SI/2015/3279: ASSET-Intelligent Assistance for Everyone Everywhere, and by the EC's FP7 program under the grant number 610516: QTLeap-Quality Translation by Deep Language Engineering Approaches.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `http://tensorflow.org/`, software available from tensorflow.org
2. Bogdanova, D., dos Santos, C., Barbosa, L., Zadrozny, B.: Detecting semantically equivalent questions in online user forums. CoNLL 2015 p. 123 (2015)
3. Collins, A.M., Quillian, M.R.: Retrieval time from semantic memory. Journal of Memory and Language 8(2), 240 (1969)
4. Deeplearning4j Development Team: Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. `http://deeplearning4j.org`
5. Erk, K.: Vector Space Models of Word Meaning and Phrase Meaning: A Survey. Language and Linguistics Compass 6(10), 635–653 (oct 2012)
6. Harris, Z.S.: Distributional structure. Word 10(2-3), 146–162 (1954)
7. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL. pp. 746–751 (2013)
8. Smith, E.E., Shoben, E.J., Rips, L.J.: Structure and process in semantic memory: A featural model for semantic decisions. Psychological review 81(3), 214 (1974)
9. Wittgenstein, L.: Philosophical Investigations. Basil Blackwell, Oxford (1953)