Enriching word embeddings with vectorization of pronunciation featurese

Heeyeon Yoo and Hyun-gwon Cha
{h23yoo, h2cha}@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Abstract

There are researches in linguistics that suggest connections between sound and meaning in words (Perlman, Dale, Lupyan 2015). While humans recognize a word by both of its semantic and phonemic features, these two features are usually modeled separately in machine learning with word embeddings and grapheme to phoneme conversions. Our interest lies in developing an effective method to enrich word embeddings by using both semantic information and pronunciation of a word. In this paper, we propose an all-in-one model that integrates functionalities of both word embeddings and grapheme-to-phoneme conversions.

Introduction

A vector representation of words is an important topic in natural language processing (NLP). Word embedding is an NLP technique for mapping human language to vectors, which are a better computer-understandable format. Word embeddings are widely applied across different research topics, industries and products, such as image captioning system, computer vision and machine translation. The key research focus of word embeddings has been on how to come up with such a mapping, and how to evaluate which is a good one.

Meanwhile, there have been active researches in predicting pronunciation of a word, known as grapheme-to-phoneme(G2P) conversion. G2P conversion is a key component in many deep learning-powered technologies including speech recognition, keyword spotting and text-to-speech synthesis. Unlike word embeddings which models the semantic contexts of a word as vectors, G2P entirely focuses on the phonemic features of a word.

This paper presents a mapping scheme from human language to vectors that contain both semantic context and pronunciation of words by bridging the previous works in word embeddings and G2P. The key algorithms in these methods were analyzed, including CBOW and Skip-Gram from word embeddings and HMM from G2P. In addition, dimensionality reduction techniques such as PCA were analyzed and applied to provide simple fixed-size vectors. The datasets commonly used to train and evaluate word embeddings will also be used in this work for fairness, in addition to phonetic transcription converter from the CMU Pronouncing Dictionary in python NLTK package. We will follow the same evaluation conventions as previous works and use spearman correlation for performance measures.

Related Work

The grapheme to phoneme conversion algorithm is an active research area with many different models proposed such as Hidden Markov Models (Taylor 2005), sequence-to-sequence neural net (Yao and Zweigh 2015) and bi-directional recurrent neural networks (Ni, Shiga and Kawai 2018).

With the rise of deep learning and its application in many research areas, it has also revolutionized successful word vectorization. Neural networks were applied in word embeddings and proved its success in representing semantic and syntactic context of words. The two key algorithms are CBOW (Mikolov, Chen, Corrado and Dean 2013) and skip-grams (Mikolov, Sutskever, Chen, Corrado and Dean 2013).

There are previous researches proposing word embeddings for speech recognition, but they either embed only semantic component (Chung and Glass 2018) or only pronunciations (Bengio and Heigold 2014). There is a research work that proposes the way to improve word embedding using both writing and pronunciation (Zhu, Jin, Ni, Wei and Lu 2018). They experimented with Chinese, English and Spanish by adding word embedding to pronunciation vector. They left it open-ended by noting that this addition can be replaced with other operations. However, this addition can harm the meaning of word embedding. For instance, previous word embedding had the feature of woman+king = queen. Now, because they added pronunciation vector, the semantic components are now distorted.

Data Description

Data Source

To compare the effectiveness of different methods to obtain PWE, the same datasets from [the article] are used to train and evaluate PWE. English Wikipedia Dump is about 15.7 GB big corpus that our PWE is trained on. The corpus is a complete copy of all Wikipedia documents covering the wide range of topics from art to technology. Although word embeddings trained on corpora of certain fields performs better on those fields than the word

embeddings trained on general corpora[some article saying that?], its wide coverage of topics is appropriate for our goal since we aim to train PWE for benchmarking, not for specific area. The corpus can be downloaded from https://dumps.wikimedia.org/enwiki/. The datasets used to evaluate PWE are Mturk-771, Men, WS-353-Sim, and WS-353-Rel. Mturk-771 contains 771 word pairs with scores for relatedness assigned by human participants. Men contains 3000 word pairs with human-assigned similarity judgement. WS-353-Sim and WS-353-Rel respectively have 153 and 200 word pairs along with similarity and relatedness, also respectively, assigned by human. All those four datasets are widely used in benchmarking trained word embeddings.

Data Preparation

English Wikipedia Dump is in XML format. The only part used to train a word embedding is the content of text tag, which is a text representation of hypertext in a Wikipedia document. Thus, it is necessary to pre-process the corpus by removing unnecessary tags containing meta-data and converting hypertext into plain text such that the result is just a long sequence of words. In addition, a sequence of phonetics corresponding to the sequence of words should be created and merged. The datasets for evaluation do not require complex pre-processing steps. Converting them into csv file format to simplify reading in should suffice.

Predictive Models

The word embedding algorithms we chose to expand on are CBOW and Skip-grams. CBOW is a neural net model that takes a bag of words as its input and produces a missing word that is most probable based on input. In contrast, Skip-grams takes a single word as its input and produces most probable surrounding words of the given word. They have a hidden layer in between the input layer and the output layer. During the training procedure, the weight matrices between the layers are adjusted to minimize the error based on the models output and the target. In addition to word embedding produced by CBOW and Skip-grams, we combine the word embedding and the pronunciation embedding, which we will now denote as H-mapping.

As mentioned in the introduction, we used the phonetic transcription converter from the CMU Pronouncing Dictionary in python NLTK package. Suppose p is the pronunciation of word w, vector, v_p represents the phoneme vector of word w and v_w represents the word embeddings of word w. Also,let α and β be weight scalars. let Then, the modified word embedding to include the pronunciation is defined as follows:

$$\hat{v} = [\alpha \cdot v_w \beta \cdot v_p]$$

Previous works either used only semantic vectors v_w, only pronunciation vectors v_p or addition of the two $v_w + v_p$. By simply concatenating the pronunciation embedding to the word embedding, we can preserve both types of information in our result vectors. To see the effectiveness of H-mapping compared to their simple addition, we have set every other experiment environment as identical as possible. Since CBOW and Skip-gram is the model they used in the article, we decided to use the same model.

References

Perlman, M., Dale, R. and Lupyan, G., 2015. Iconicity can ground the creation of vocal symbols. Royal Society open science, 2(8), p.150152.

Taylor, P., 2005. Hidden Markov models for grapheme to phoneme conversion. In Ninth European Conference on Speech Communication and Technology.

Yao, K. and Zweig, G., 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. arXiv preprint arXiv:1506.00196.

Ni, J., Shiga, Y. and Kawai, H., 2018. Multilingual Grapheme-to-Phoneme Conversion with Global Character Vectors. Proc. Interspeech 2018, pp.2823-2827.

Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

Chung, Y.A. and Glass, J., 2018. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. arXiv preprint arXiv:1803.08976.

Bengio, S. and Heigold, G., 2014. Word embeddings for speech recognition. In Fifteenth Annual Conference of the International Speech Communication Association.

Zhu, W., Jin, X., Ni, J., Wei, B. and Lu, Z., 2018. Improve word embedding using both writing and pronunciation. PloS one, 13(12), p.e0208785.