Misspelling Oblivious Word Embeddings (MOE)

Technology Review, Submitted by Vijayaragavan Selvaraj (VS27)

Introduction

This technology review is about Misspelling Oblivious Word Embeddings, in short, MOE. It is a novel method to learn word embeddings that are resilient to misspellings and malformed words, in other words, Out-Of-Vocabulary (OOV) words. Most popular word embedding techniques like Word2Vec and GloVe have a drawback that they cannot provide word embeddings to OOV words. In real-world applications, the input text is often generated by people and misspellings are frequent that it eventually diminishes the quality of the downstream applications. To overcome this, Facebook introduced this new method to train word embeddings by combining FastText with subwords and a supervised task that embeds misspellings close to their correct variants.

Prior to MOE

Word2Vec is a very popular model that is effective in training models on a huge text corpora efficiently. It uses skip-gram with negative sampling model (SGNS) to train word embeddings. Skip-gram uses context words to predict surrounding words in order to learn word embeddings whereas negative sampling is picking a false case for the skip-gram training.

FastText, another popular model that is introduced by Facebook, extends the Word2Vec SGNS architecture and applies subword-level features to train word embeddings for OOV words. In the SGNS model, a word w_i is represented by a single embedding vector v_i equivalent to the input vector of a simple feed forward neural network trained by optimizing a loss function L_{FT}. In addition to optimizing the loss function L_{FT}, FastText uses a scoring function S_{FT} that is the dot product of the input vector associated with a word w_i and the output vector of the contextual word w_c. FastText embeds subwords (character n-grams) in order to derive the scoring function S_{FT}. For example, the subwords (or) n-gram set for a word “banana” is ban, ana, nan, bana, anan, nana, banan, anana and the word “banana” itself. Several experiments depict that FastText improves over the original Word2Vec skip-gram model.

Loss Function: $L_{FT} := \sum_{i=1}^{|T|} \sum_{w_c \in C_i} [l(s(w_i, w_c)) + \sum_{w_n \in N_{i,c}} l(-s(w_i, w_n))]$

Scoring Function: $s_{FT}(w_i, w_c) := \sum_{v_g, g \in G_{w_i}} v_g^T u_c$

MOE Model

Though FastText captures morphological aspects of a text, it may not be resistant to misspellings. This results in the new idea, MOE that takes the fundamentals of Word2Vec and FastText but explicitly gives importance to OOV words. Here, the loss function L_{MOE} is the weighted sum of two loss functions: L_{FT} and L_{SC}. L_{FT} is the loss function of FastText which captures semantic relationships between words and L_{SC} is the spell correction loss aimed to map embeddings of misspelled words close to the embeddings of their correctly spelled variants in the vector space.

Misspelling Oblivious Word Embeddings (MOE)

Technology Review, Submitted by Vijayaragavan Selvaraj (VS27)

Spell Correction Loss Function: $L_{SC} := \sum_{(w_m,w_e) \in M} [l(\widehat{s}(w_m, w_e)) + \sum_{w_n \in N_{m,e}} l(-\widehat{s}(w_m, w_n))]$

Where M is the set of pairs of words (w_m,w_e) in which w_e is the expected word and w_m is its misspelling. $N_{m,e}$ is the set of negative samples. L_{SC} makes use of the logistic function *$l(x) = log(1 + e^{-x})$.*

Scoring Function: $\widehat{s}(w_m, w_e) = \sum_{v_g,\, g \in \widehat{G}_{wm}} v_g^T v_e$ where $\widehat{G}_{w_m} = G_{w_m} \setminus \{w_m\}$

Complete Loss Function: $L_{MOE} = (1 - \alpha) L_{FT} + \alpha \frac{|T|}{|M|} L_{SC}$

Optimizing both loss functions L_{FT} and L_{SC} is not a straight-foward task since it involves two different datasets, text corpus T and misspellings M. Optimization should be agnostic to the sizes of T and M in order to prevent results from being affected by those sizes. So, we scale L_{SC} with the coefficient |T| / |M|. The α is the hyperparameter that sets the importance of spell correction loss L_{SC} with respect to L_{FT} thus making MOE a generalization of FastText.

Performance

To analyze the performance of MOE, FastText is used as the baseline since it can also generate word embeddings for OOV words. The datasets used for conducting the experiments are Text Corpus T and Misspellings M. Text Corpus T is obtained by scrapping the wikipedia pages. The size of the Text Corpus is 4,341,233,424 words and the vocabulary is 2,746,061 words after applying deduplication and frequency threshold as 5 on the corpus. For the misspellings dataset, a script that is based on a simple error model, injects misspellings on the words collected by mining query logs of a popular search engine and identifying cases where a query was manually corrected by the searcher. The size of the misspellings dataset is 20,068,964 pairs.

Both Intrinsic and extrinsic tasks are run to evaluate the performance of MOE against FastText.

<u>Intrinsic Tasks:</u>

- Word Similarity - In this task, the evaluation is done on how well word embeddings generated by MOE can capture semantic relationships between words. Two different datasets are used for this purpose and different values for the hypermeter α are tried. The result shows that MOE performs better than the baseline FastText on both the datasets.
- Word Analogy - This task is to measure how good the embeddings is at preserving the relationships between words. There are two types of relationships that are tested here: (i) syntactic, related to the structure of words; and (ii) semantic, related to their meanings. The result of this experiment also shows that MOE outperforms FastText.
- Neighbourhood Validity - This task checks where in the neighborhood of a misspelling the correct word is situated which is the objective of MOE. For a pair of misspelling and correction, the task first picks k nearest neighbors of the misspelling in the embedding

space using cosine similarity as a distance metric. Then, it evaluates the position of the correct word within the neighborhood of the misspelled using two metrics, MRR and neighbourhood coverage. The result shows that MOE is more likely to surface the correct words than the FastText baseline.

Extrinsic Tasks:

- POS Tagging - This task is to evaluate MOE on Parts-Of-Speech tagging. In order to do this, three different datasets are used. First dataset has no misspelled words, the second dataset has 10% of words misspelled and the final one contains 100% misspelled words. A state-of-the-art POS tagger consisting of a Conditional Random Fields (CRF) model is used in this task. The results show that MOE attains a sensitive improvement and also does not reduce effectiveness of CRF POS Tagger.

Conclusion and Future Work

A most important issue of word embeddings is that they cannot deal with malformed words and it is a huge drawback in the real-world applications. This new model, MOE, is introduced to solve that problem; generating high quality, semantically valid embeddings for misspellings. From all the experiments, it is clearly evident that MOE is very effective in mapping embeddings of misspellings close to the embedding of the corresponding correctly spelled word. Also, it does not affect the effectiveness of the POS Tagger in the case of correctly spelled words and improves sensitively the quality of the POS tagger on misspellings. In the future, the model can be used in different ways of training embeddings for misspellings including the extension of the same technique to multilingual embeddings.

References

- B. Edizel, A. Piktus, P. Bojanowski, R. Ferreira, E. Grave and F. Silvestri. Misspelling Oblivious Word Embeddings. 2019

- T. Mikolov, G. Corrado, K. Chen and J. Dean. Efficient Estimation of Word Representations in Vector Space. 2013