Robust Text Classification using Sub-Word Information in Input Word Representations

Mahanti Bhanu Prakash, Priyank Chhipa, Vivek Sridhar, Vinuthkumar Prasan
Samsung R&D Institute India, Bangalore
{me.prakash, p.chhipa, v.sridhar, vinuth}@samsung.com

Abstract

Word based deep learning approaches have been used with increasing success recently to solve Natural Language Processing problems like Machine Translation, Language Modelling and Text Classification. However, performance of these word based models is limited by the vocabulary of the training corpus. Alternate approaches using character based models have been proposed to overcome the unseen word problems arising for a variety of reasons. However, character based models fail to capture the sequential relationship of words inherently present in texts. Hence, there is scope for improvement by addressing the unseen word problem while also maintaining the sequential context through word based models.

In this work, we propose a method where the input embedding vector incorporates sub-word information but is also suitable for use with models which successfully capture the sequential nature of text. We further attempt to establish that using such a word representation as input makes the model robust to unseen words, particularly arising due to tokenization and spelling errors, which is a common problem in systems where a typing interface is one of the input modalities.

1 Introduction

Recent research has demonstrated the success of word based models for NLP problems like Machine Translation (Sutskever, 2014) and Text Classification (Mikolov, 2010). It is well established in literature that the dictionary of words contained in the training corpus have significant bearing on the performance of these models. For example, in the case of language modelling, an unseen word can never be predicted and models also tend to have lower accuracies when predicting words in the vicinity of an unseen word. Models for text classification also suffer from a similar problem wherein one or more unseen words in the input may significantly increase classification error. Alternate approaches using character based models (Zhang, 2015; Kim, 2016) have been proposed to overcome the unseen word problem which ails word-based deep learning networks. However, character based models fail to capture the sequential relationship of words inherently present in texts.

The main contribution of this paper is to establish the suitability and robustness of an input embedded vector which incorporates sub-word information (Bojanowski, 2016) with a recurrent neural network model for sentence classification and also establish the capability of such a configuration to deal effectively with the unseen word problem, especially arising due to word segmentations and spelling errors.

2 Related Work

The input to text based deep learning models is usually a numeric vector representation of text, commonly called embedded vectors. The embedded vector of each word is designed to be indicative of its semantic relationship with other words or characters as available in the corpus in embedded space. This usually constitutes the very first layer of the network. This layer may be initialized randomly or with pre-trained vectors The pre-trained vectors may be static or may also be learned with the network. These pre-trained vectors are typically generated from a large training corpus which is usually not directly related to the problem at hand, but is representative of language as a whole. One of the most commonly

D M Sharma, P Bhattacharyya and R Sangal. Proc. of the 16th Intl. Conference on Natural Language Processing, pages 1–8
Hyderabad, India, December 2019.

used pre-trained embedding is proposed by Mikolov (2010) where a neural network approach is used to generate word vectors based on a 1.6 billion words data set. An alternate approach discussed in Pennington (2014) focusses on whether distributional word representations are best learned from count-based methods rather than prediction-based methods. Since these vectors are pre-trained using large corpora, they contain meaningful semantic representations of even words not seen in the training corpus for the specific problem, which helps deal with the unseen word problem to a certain extent. In addition, models tend to converge faster when pre-trained vectors are used. However, pre-trained word embedding approaches continue to have difficulty with words not in the dictionary of the input embedding. Also, rare words are often not represented as well as more frequently occurring words. Words not seen in the training corpus are usually either marked as unknown (UNK) or excluded altogether from the input. To address these drawbacks, several alternate approaches have been proposed. Zhang (2015) proposed a 9-layer character based CNN model which addresses the unseen word problem in Word based models. However, this CNN based approach fails to capture sequence context features in the text. Kim (2016) proposed an architecture in which the character embedding is input to a CNN, the output of which acts as input to an RNN. In such models, the CNN component captures the n-gram features of text and RNN takes care of sequence context of such features in the text. For character based CNN models, the context or relationship between multiple characters and words are captured by convolution filters or kernels. A set of fixed filter sizes (n-grams) may not completely capture word-level information. Also, capturing longer context is difficult in CNN models. Another alternate approach is proposed by Bojanowski (2016) where each word is represented as a bag of character n-grams and a vector representation is associated to each character n-gram. Words are represented as a sum of these representations. This is found to be especially effective when dealing with morphologically rich languages. This has been used with shallow models for sentiment analysis and tag identification problem in Joulin (2016). However, for more complex problems over a larger number of classes, the higher representational power of deep networks such as RNNs and CNNs may be desired.

In this work, we apply the method for generating vector representations proposed in Bojanowski (2016) to deep learning networks such as the architecture proposed in Sutskever (2014) and explore the extent to which unseen word problem, especially arising due to misspellings and tokenization errors, is addressed.

3 Proposed Approach

Unseen words are a common occurrence in NLP problems and arise from a variety of situations. The most common reasons for unseen words is simply a lack of exhaustive training data for a specific problem. This problem is largely dealt with by using pre-trained distributions trained on a large corpus. Another common source of unseen words is morphological variance. This is a scenario where the unseen word is close to a seen word both superficially and semantically. Research described in Bojanowski (2016) and Joulin (2016) show that input vectors incorporating sub-word information have proved effective in tackling this problem.

Another source of unseen words are misspellings or incorrect word segmentation. This is a common problem faced in multi-modal applications such as voice assistants wherein one of the input modalities is a typing interface. These types of errors seem to be similar to the UNKs arising from morphological variance wherein the unseen word shares a close superficial as well as semantic similarity with a seen word.

We propose to use pre-trained embedded vectors to deal with the unseen word problem, especially due to misspellings, using word vectors which incorporate sub-word information. An RNN based sentence classifier with an architecture similar to the one proposed by Sutskever (2014) is used and compared with the performance of the distributions described by Mikolov (2013) and Pennington (2014) on standard data sets for text classification. In addition, we intend to simulate the UNK problem due to misspellings and incorrect tokenization by applying rules to the standard data sets. These rules consist of common misspellings such as “ei” instead of “ie”, incorrect double consonants (‘aggressive’ vs ‘aggresive’) and so on.

3.1 Word Representations with Sub-Words

Word vectors are traditionally used to cluster words with a high degree of semantic similarity. These representations help to deal with rare word problem wherein words which are less frequently present in training data are not learnt as well as words which are semantically similar but present more frequently. Word distributions learnt on larger corpora alleviate this problem. Another source of rare or unseen words is morphological variance. This problem is particularly important for languages which have a high degree of inflection. The work by Bojanowski (2016) uses a word representation which incorporates sub-word information to construct word vectors to deal with this problem.

Misspelled words also share several sub-words with the actual word and we therefore propose that a word representation which incorporates sub-word information should be good at dealing with UNKs due to spelling or tokenization errors. This intuition is validated by the clusters shown in the word representations of the misspelling words projected in two dimensions.

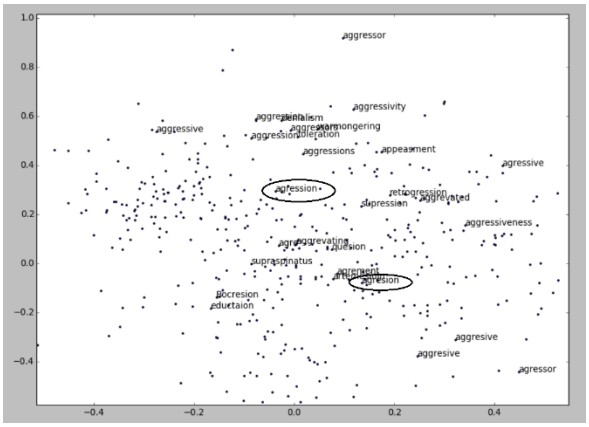

Figure 1: Word clusters for 'aggression'

As seen in Figure 1, various common misspellings of the word 'aggression' are clustered close to the actual word and related words like 'aggresion', 'aggressive' and so on.

Figure 2 shows that another common misspelling, 'ie' instead of 'ei', is dealt with well by the word vectors constructed using sub-words.

Figure 3 shows the incorrect tokenization of the phrase 'remind me' as 'remindme' which is placed in the relevant cluster containing 'reminds', 'remind', 'reminded' and other similar words.

The above plots indicate the robustness of a word vector built using sub-words to deal with spelling and word segmentation errors. We attempt

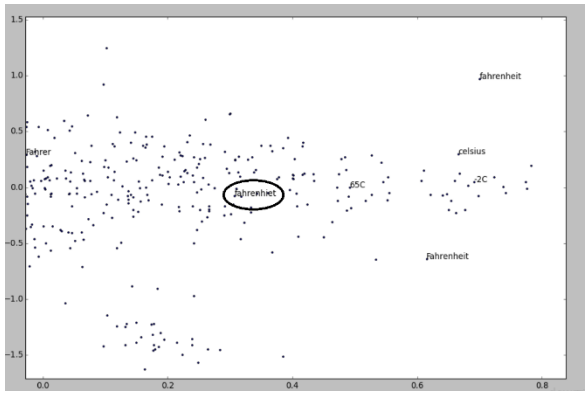

Figure 2: Cluster for 'Fahrenheit'

to establish this conclusively by applying such a word representation to a sentence classification problem using a standard word-based RNN architecture described in the following sections.

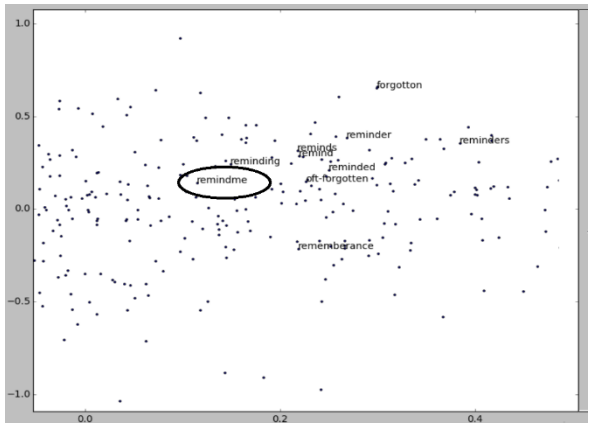

Figure 3: Cluster for tokenization (remind me)

3.2 Sentence Classification RNN Model

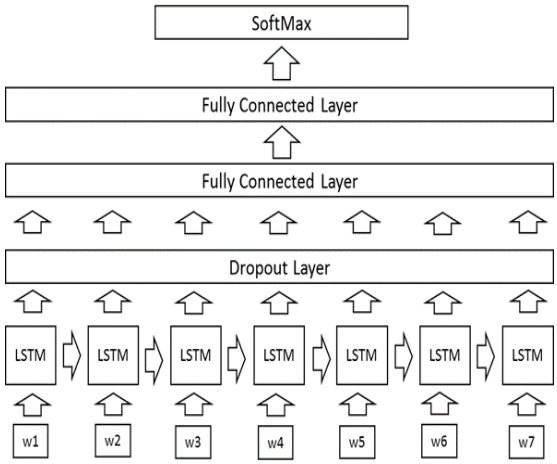

Figure 4: RNN based Sentence Classifier

Sentence classifier architecture used is depicted in Figure 4 and is based on the sequence to

sequence learning architecture for machine translation described in Sutskever (2014).

The input 'wi' is the embedded vector used to represent words in the input text. Experiments were carried out using the word distributions described in Mikolov (2013) and Pennington (2014) and used as reference for comparison.

$$Loss = -\sum_{c=1}^{M} y_{o,c} \log(P_o, c) \quad (1)$$

The model is trained with the 20% dropouts and categorical cross entropy loss function represented by the Equation 1. The rmsprop optimizer is used which is a popular choice for Recurrent Neural networks. A 300-dimension word vector is used as the input.

3.3 Sentence Classification RNN Model with Pre-Trained Embedding containing Sub-Word Information

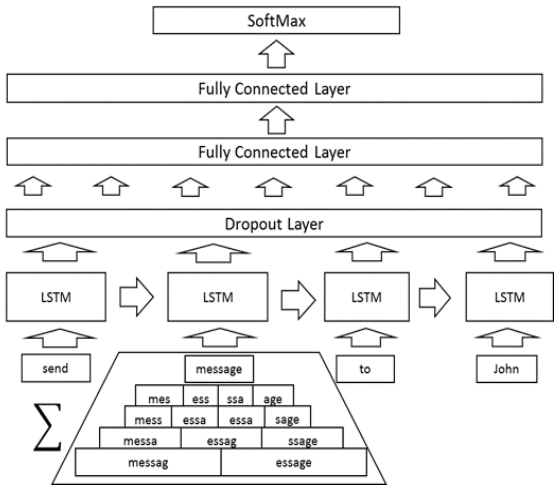

Figure 5: RNN based sentence classifier with sub-word information based embedding

Figure 5 shows the architecture used with the input vector built using a combination of sub-words. One of the configurable parameters while generating the pre-trained embedded vectors is the sub-word length to be incorporated. Sub-words consisting of lengths from 3 to 6 along with the whole word are used to generate the input embedding.

The skip-gram model described by Bojanowski (2016) is used to generate the pre-trained input vector representation using One-billion-word benchmark (Chelba, 2013). This is used for comparison with the reference word distributions described in Mikolov (2013) and Pennington (2014) which are commonly used with deep neural network based architectures.

4 Datasets and Experimental Setup

The sentence classification task is chosen for the work described in this paper.

In our first set of experiments, we apply the reference word-based sequence learning architecture of Sutskever (2014) to the sentence classification problem on three standard datasets. The pre-trained embedding learnt through the method proposed in Bojanowski (2016) is used as a static input embedding and is not updated as part of the training for the specific problem.

We compare the performance of these sub-word based input embedded vectors with the popular GloVe and Word2vec pre-trained word representations using this architecture. This is used to establish the suitability of the word representations of Bojanowski (2016) to deep learning architectures.

In our second set of experiments, we apply the reference word-based sequence learning architecture of Sutskever (2014) to the sentence classification problem on three standard datasets which are modified to incorporate misspellings. Two standard misspelling dictionaries are used to generate the misspelled versions of the standard datasets. This is done in order to simulate real-world situations, such as multi-modal smart assistants, where the input to a sentence classification system may be via a textual input interface, and therefore prone to misspellings. The performance of sub-word based word vectors Bojanowski (2016) is compared with reference distributions. This is used to establish that word representations which are constructed using sub-word information are robust and more suitable for use in multi-modal commercial applications than the more popular GloVe and Word2vec word representations. The datasets used for these experiments are described in detail in the following sections.

4.1 Sentence Classification Datasets

The following three datasets are used for benchmarking on the sentence classification problem.

The SUBJ Subjectivity dataset is a two-class dataset where the task is to classify a sentence as subjective or objective. There are 10000 sentences,

we used 5-fold cross validation with 80% of the data for train and the remaining 20% are used as a test set.

The MPQA Opinion polarity dataset has 10606 sentences with two classes. We use 80% of the data for model training and remaining 20% for testing. A 5-fold cross validation result is presented since a pre-defined test and train split is not available.

AG News dataset is a four-class dataset where the task is to classify the sentence into 'world', 'sports', 'business', 'science and tech'. This dataset has two parts: title and description, but we only considered the title for text classification. The dataset contains 120000 training and 7600 test samples.

4.2 Spelling Error Dictionaries

There are two broad sources of misspellings, namely phonetic and typographic. We use the following two reference dictionaries which focus on these two kinds of misspellings.

The Wikipedia[1] misspelling dictionary contains 2,455 misspellings of 1,922 words. This is a list of common misspellings made by Wikipedia editors. This dictionary focuses mainly on the typographic misspellings, but also includes several common phonetic based spelling errors.

The Aspell[2] dictionary contains 531 misspellings of 450 words. This dataset focusses on phonetic misspellings. Aspell begins by converting the misspelt word to its sounds-like equivalent using Metaphone and moves on to find all words that have a sounds-like within one or two edit distances from the original word's sounds-like. These sounds-like words are the basis for the suggestions of Aspell. This is derived by Atkinson[2] for testing the GNU Aspell spellchecker.

5 Results

The reference model of Sutskever (2014) described in the sections above with GloVe and Word2vec embedding vectors used as the input word vectors has been compared with the proposed word embedding on the three standard sentence classification datasets as described in the Table 1.

The main purpose of this experiment is to prove the performance of the proposed word embedding using sub-words with a word-based RNN sequence learning architecture.

[1] http://www.dcs.bbk.ac.uk/~ROGER/wikipedia.dat
[2] http://aspell.sourceforge.net/

The performance of the proposed input word embedding applied to the reference architecture is

Dataset	Test Set Size	Standard Data Set		
		GloVe	Word2vec	Sub-word embedding
subj	1000	85.68	86	84.58
MPQA	1000	89.8	89.8	89.8
AGNews	7600	87.2	87.5	87.5

Table 1: Comparison of reference distributions with proposed approach on standard datasets

comparable to the performance with the Pennington (2014) and Mikolov (2010) input embedded vectors. This illustrates the suitability of the sub-word embedding for use with deep neural networks.

The accuracies shown in Table 1 above are used as benchmarks for our further investigation into the capacity of the various types of input embedded vectors to deal with misspellings and tokenization errors.

Dataset	Misspelling Dictionary : Wikipedia			
	Changed Data	GloVe	Word2vec	Sub-word embedding
subj	964/1000	79.72	78	81.3
MPQA	401/1000	66.62	66.66	87.3
AGNews	2201/7600	83.3	82.7	86.3

Table 2: Comparison of reference distributions with proposed approach on standard datasets with misspellings from Wikipedia

The Wikipedia misspelling dictionary mainly focusses on typographic misspellings. The different datasets are also differently prone to spelling errors. In the case of Subj dataset, 964 out of 1000 test sentences are modified, but a majority of these misspellings are words like 'the' and 'and', which are typically less likely to affect the classification result. However, 40% of the MPQA test set is modified and about 29% of the AG News data set is modified by the Wikipedia misspelling dictionary. This serves to illustrate the need to deal with misspellings as part of any commercial application.

The results in Table 2 show that the proposed approach is always better than the reference embedded vectors at dealing with the UNK problem arising due to spelling errors. In certain cases, the improvement is marginal (Subj: ~1%)

whereas in the best case (MPQA), a huge improvement of over 20% is observed. These results indicate that the proposed approach is significantly better at dealing with UNKs arising due to typographic misspellings.

The Aspell misspelling dictionary mainly focusses on phonetic misspellings. Similar to the discussion above, the different datasets show varying susceptibility to spelling errors. Applying the misspellings from the Aspell dictionary results in 12% and 14% of the MPQA and AG News test sets being modified respectively. In the case of the Subj dataset, a much larger 86% of the 1000 test sentences are modified.

Dataset	Misspelling Dictionary : Aspell			
	Changed Data	GloVe	Word2vec	Sub-word embedding
subj	859/1000	81.75	80.56	83.8
MPQA	145/1000	70.69	70.6	89
AG News	970/7600	84.5	83.3	86.79

Table 3: Comparison of reference distributions with proposed approach on standard datasets vs misspellings from Aspell

The results in Table 3 show that GloVe and Word2vec word representations show a drop in performance due to misspellings for all 3 datasets ranging from 3% in the case of AG News to 19% in the case of MPQA dataset and that the proposed approach is better than the reference embedded vectors at dealing with the UNK problem arising due to phonetic spelling errors in all cases.

Datasets	Best Accuracy (without misspelling)	Proposed Approach (Wikipedia misspelling)	Proposed Approach (Aspell misspelling)
SUBJ	86	81.3	84.58
MPQA	89.8	87.3	89
AG news	87.5	86.3	86.79

Table 4: Comparison of proposed approach on misspelled data with best accuracy on original datasets

Table 4 compares the performance of the proposed approach on the datasets modified with misspellings with the best accuracy out of any of the three word representations on the original dataset without misspellings. The purpose of this comparison is to measure the extent to which the proposed approach addresses the problem of spelling errors.

In most of the cases, only a minor drop in accuracy is observed ranging from 0.7% to 2.4%. The only outlier is the Wikipedia dictionary modified SUBJ dataset where a significant drop of over 4.7% is seen. Detailed analysis shows that the most common spelling modifications in this dataset are the words 'the' and 'and' which the sub-word based representation doesn't deal with well as the number of sub-words for very short words are too less to have a significant impact in generating the word vectors. Some more specific situations which are not handled well by the proposed approach are discussed in the following section along with the direction our future work will take to address these problems.

Overall, the performance of the proposed approach is close enough to the performance on the original datasets without misspellings to indicate that the proposed approach is not only comparable to the state-of-the-art when applied to deep learning architectures but also solves the UNK problems arising due to typographic and phonetic misspellings to a significant extent.

6 Discussion

It is seen that 7% of the attendees of the TOEFL[3] examination, a test of English, tend to make spelling errors, even in an environment where the sole focus is correctness of grammar and language. Our study of internal data from a Voice Assistant applications indicates that in excess of 25% of all data input using a typing interface contains errors in spelling and word breaks. This illustrates the need for a method to handle spelling errors gracefully and reliably, especially for more natural AI applications.

The major motivation to conduct the investigations presented in this work was to come up with a technique to deal with the misspelling problem which is inherently present in multi-modal voice assistants where the primary input paradigm is speech, which is not prone to misspellings at all, and the secondary modality is a typing interface which is quite prone to spelling and word segmentation errors. The goal was to use a technique wherein the models trained on well-formed data are robust to errors in spelling rather than to implement a relatively clumsy rule-based preprocessing module which would attempt to

[3] https://www.researchgate.net/figure/Average-percent-of-misspelled-words-per-essay-by-NS-NNS-and-score-panel-A-GRE-data_fig3_277584335

correct misspellings but would tend to be unreliable by nature.

A study of the misspelling dictionaries and the substitutions made to the standard datasets using these dictionaries gives further clarity on the various types of spelling errors commonly seen. The first level classification of the types of spelling errors is typographical and phonetic. Typographical errors consist mainly of omission, addition or swapping of characters. All these three cases seem to be handled reasonably well using the sub-words approach to construct word vectors. The second major category of phonetic misspellings consists mainly of replacement of characters by other similar characters such as 'destruction' vs 'distruction' and so on. Other common errors of this kind are incorrect usage of double consonants, 'ei' instead of 'ie' and so on. The majority of these cases are also handled well by the sub-word based word representations. One of the observations while analyzing the drop in accuracy of the models on the misspelled datasets is that misspellings towards the middle of the word are not dealt with as well as misspellings near either end of the words since more number of sub-words are affected in this case. We are currently working on some improvements to the word representations to overcome this problem.

7 Conclusion

The work in this paper demonstrates that input embedded vectors which incorporate sub-word information and are learnt through a shallow network are well-suited for use with sequence aware deep learning networks. It also showcases the effectiveness of such a configuration in dealing with various common types of spelling errors arising due to both typographic as well as due to phonetic reasons. The results showing that the accuracy of the proposed configuration on the standard datasets with misspellings is comparable to the best performance on the misspelling free datasets indicate that the proposed configuration almost entirely solves the problem of spelling errors.

The proposed work is especially suited for use is multi-modal applications as it not only seamlessly handles spelling errors but performs as well as state-of-the-art systems on correctly spelled inputs. One example of such a real-world application is a multi-modal voice assistant which allows textual input in addition to speech input. Modern multi-modal voice assistants attempt to support a very wide range of complex functionality for which deep learning networks are a natural choice and will greatly benefit from an input embedding which seamlessly handles misspellings. Moreover, the approach used to construct these input embedded vectors also handles morphological variance and is applicable across languages.

This work also establishes the similarity in nature between morphological variance and spelling or tokenization errors wherein the unseen word is both semantically and superficially similar to an actual seen word, and therefore improvements made in dealing with one are likely to be beneficial in dealing with the other. This opens up the possibility of a wide area of research as this work proves a significant overlap between two problem statements which were hitherto perceived to be different.

8 Future Work

Our future work will focus on proving the applicability of the proposed approach across languages by extending the experiments conducted here to more languages.

Another line of research we are pursuing focusses on improving the method of selecting sub-words in order to deal better with certain kinds of morphological variance and spelling errors, such as omission of a character in the middle of a long word, which this proposed approach doesn't deal with well in some cases.

We also intend to improve this approach to be robust to other variance arising from other forms of textual input such as text messages, tweets and so on.

References

Bojanowski, P. et al., 2016. Enriching word vectors with subword information. *Computing Research Repository*, arXiv:1607.04606. Version 2.

Chelba, C. et al., 2013. One billion word benchmark for measuring progress in statistical language modeling. *Computing Research Repository*, arXiv:1312.3005. Version 3.

Kim, Y. et al., 2016. Character-Aware Neural Language Models. *Association for the Advancement of Artificial Intelligence*, pages 2741-2749

Kim, Y. 2014. Convolutional neural networks for sentence classification. *Computing Research Repository*, arXiv:1408.5882. Version 2.

Joulin, A. et al., 2016. Bag of tricks for efficient text classification. *Computing Research Repository*, arXiv:1607.01759. Version 3.

Mikolov, T. et al., 2010. Recurrent neural network based language model. *Eleventh Annual Conference of the International Speech Communication Association*.

Mikolov, T. et al., 2013. Efficient estimation of word representations in vector space. *Computing Research Repository*, arXiv:1301.3781. Version 3.

Pennington, J. et al., 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532-1543.

Sutskever, I. et al., 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, pages 3104-3112.

Zhang, X. et al., 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, pages 649-657.