A text classification method of power grid assets based on improved FastText

Feng Zhao[a], Cuiling Jia[b*], Wenjing Li[a], Jinlong Hao[b], Jiangbo Yin[b], Liyun Pan[b], Yang Yang[b]
[a] State Grid Information & Telecommunicaition Group Co., Ltd, Beijing 100089, China; [b] Beijing China-Power Information Technology Co., Ltd, Beijing 100089, China

ABSTRACT

In today's era, information technology is constantly updated, and the degree of social information is getting higher and higher. Power grid enterprises have a large amount of asset data, and There are many types of assets, such as tools, information system and vehicles. These assets need to be managed. Although there are unified codes for the types of assets at present, manual input is still inaccurate, so there will be some inaccurate descriptions, which makes it very difficult to promote the management of power grid assets. In this paper, an improved FastText text classification method is proposed to identify the information of various power grid assets. The purpose is to ensure that the technical object type coding of power grid assets matches the type of power grid assets, and to realize the efficient quality management of power grid asset object type coding data Concrete by using n-gram model information, read between the word and the word order in hidden layer of network by traditional method and the average improvement for the average sum of squares, and the network output layer for the softmax improved hierarchical softmax and negative sampling method, improved rapid extraction and classification of information, and can better meet the automatic identification data grid assets. The experimental study in this paper proves that the classification accuracy of the improved FastText model on the grid asset data set is significantly improved, and the intelligent classification of grid assets can be realized.

Keywords: Grid assets, FastText, text classification, quality management

1. INTRODUCTION

In recent years the company to follow the steps of digital revolution, created a lot of business data However, in practical application process, it is found that there are abnormal situations in the data, and it causes serious consequences. A lot of these data are empty data, wrong data can not be used. For analytical applications built on data, many data quality problems, such as unmatched data, redundant and repeated data, insufficient precision, unidentifiable data, and poor timeliness, have affected the effect to a certain extent. Therefore, it is urgent to carry out data governance, which plays a huge role in further improving application services. At present, data management work mainly depends on manual, there is a large workload, repeated inefficient, dependence on subjective judgment and other problems. To solve the above problems, artificial intelligence techniques such as data mining, knowledge graph and natural language processing can be introduced.

In this paper, the type codes of power grid assets are not uniform. Currently, experts are only required to deal with huge abnormal data, which is time-consuming and inefficient and difficult to maintain continuously. Therefore, it is necessary to use artificial intelligence method to detect anomalies of power grid asset types. Compared with other numerical fields, asset description is more reliable as a multi-text field in predicting the type of power grid assets. The natural language processing algorithm is used to automatically confirm the asset type according to the description fields in the asset master data, and solve the problems of missing asset types and misclassification of assets in the asset master data, which can effectively improve the efficiency of data governance.

2. RELATED RESEARCH

In terms of NLP text processing, the most widely used models and methods include Word2vec model, Convolution neural network (CNN) model, feature extraction Naive Bayes (NB), etc.

(1) Word2vec model

* 1062648009@qq.com

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 1250609
 · 0277-786X · doi: 10.1117/12.2662731

Word2vec model is to transform words into vector form, become data type, through constant length, using autonomous way to learn semantic structure. Word2vec is divided into continuous word bag model (CBOW) or Skpp-Gram model, which analyzes context relations through different inputs and trains word generation vectors. CBOW can be divided into three levels. Firstly, it takes the given word as the input layer, and then combines it with the vector, and classifies the next word by the Huffman tree of the third layer. However, the former is greatly different from the latter. The former uses intermediate words as the input of the model, and the final result is the estimation of its context word vector. The Word2vec model finds such words by estimating the degree of similarity between them.

(2) Feature extraction of convolutional neural network (CNN) model

CNN was applied to text classification tasks in natural language processing in 2014, including convolution layer, linear rectification layer, pooling layer and complete connection layer. Figure 1 is the network structure, the whole process can be described as a different input value through the input layer into numerical matrix, filling the convolution kernel operation characteristic figure, at a certain distance of the set, different convolution kernels can have different characteristics, and then the results are taken as the input of the pooling layer to select its main features. After the matrix is pooled, the parameters will be greatly reduced. Finally, the extracted features are gathered together in the full connection layer.

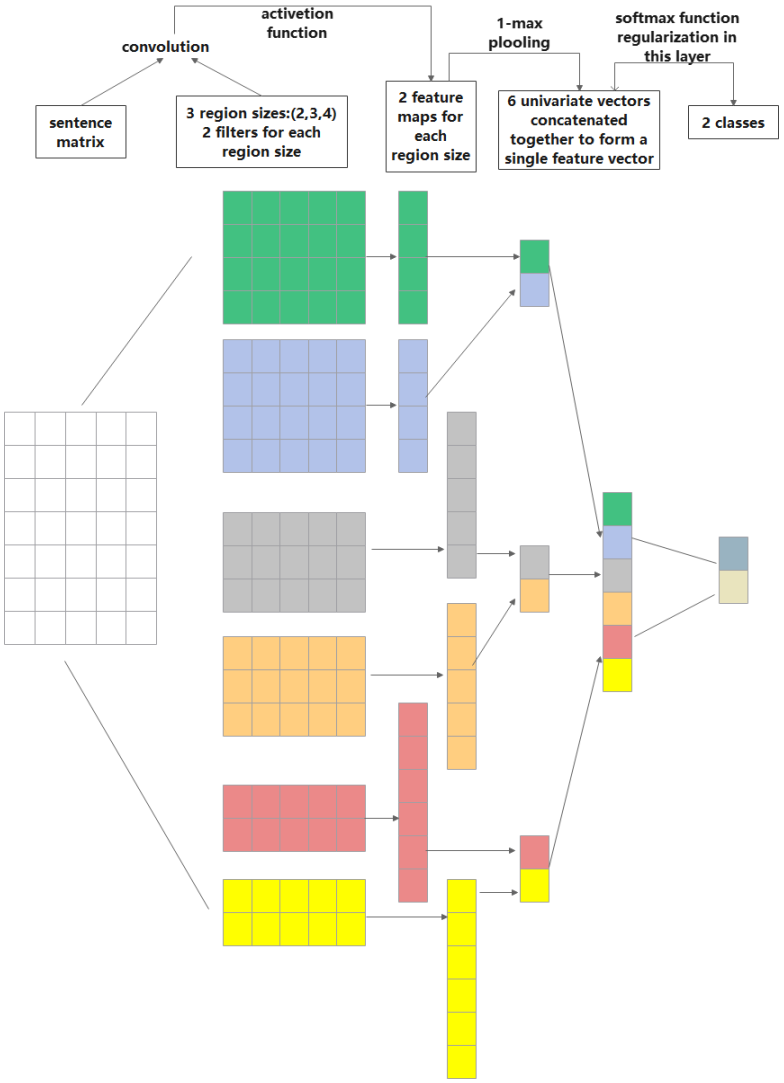

Figure 1. CNN model.

The convolutional layer in CNN model has different convolutional kernels, so the extracted features will be inconsistent. The size of the convolutional kernels determines the distance that CNN can capture features, and then the pooling layer will reduce the dimension of features. As the convolution proceeds, feature vectors are formed, and the full connection layer is classified at the end of the network. However, CNN model cannot capture long-distance features due to its low network structure, and location information will be lost in pooling. To solve the above problems, researchers try different methods to improve CNN, including: increasing network structure so that the convolution kernel can acquire more different features; The expansion convolution method is adopted to cover the intervals; Discarding the pooling layer, adding location coding and so on.

(3) Naive Bayes, NB

Naive Bayes distinguishes the text category on the basis of the relative independence of various samples. If the conditional probability of this text being classified into a certain category is high, it is summed up as the same category[1]. According to this feature, as long as a certain keyword in a certain text belongs to a certain category, it is judged that the text belongs to this category. Therefore, the traditional naive Bayes algorithm can achieve relatively satisfactory results for the decrease of the relevance of contextual semantics. However, the defects are also more obvious, without considering the consequences of similar words, in many cases, the similar words in the keywords will be treated as two independent words, to a certain extent, resulting in poor classification results, often with a large difference from the actual. Equipment category in the table is set on keywords, centrifugal fan, for example, although there are "centrifugal fan" and "fan" of two different expressions, if using traditional naive Bayesian model, it is concluded that the conditional probability, because without considering both actually describe the centrifugal fan in the power grid assets. Another reason is that in the process of setting keywords, the training set is relatively discrete and low concentration.

3. TEXT CLASSIFICATION METHOD BASED ON FASTTEXT

3.1 Traditional text classification methods

The goal of text preprocessing[2] is to convert text into structured data form, as shown in Figure 2. The main process can be divided into several stages, such as text word segmentation[3, 4], morphological reduction and stem extraction[5], part-of-speech tagging[6], word cleaning, vector space representation, etc.

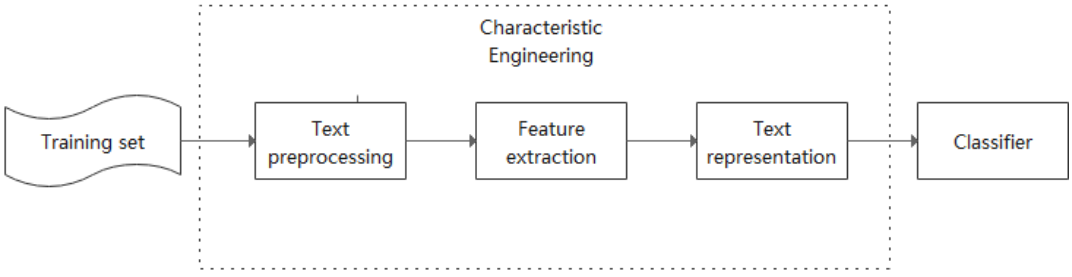

Figure 2. Text classification method.

Feature engineering[7, 8] is data preprocessing, which consumes the longest time in the text classification process and largely affects the performance of the algorithm model. About 70% of the time is spent on data feature engineering processing in daily work. In general, feature engineering is the conversion of literal language into machine language. This part is very important, and if you can split the input text into small enough pieces, you will get more accurate results. The text classifier is a process of summarizing the information that can be understood by the computer into a more concrete, reusable and transferable knowledge base. It needs to adjust the model parameters constantly and strive to obtain results that are infinitely close to the ideal data.

Figure 3 shows the basic architecture of the FastText model.

Word vectors $(x_1, x_2, \ldots x_N)$ as the model input, and the calculation formula of the hidden layer is:

$$\text{hidden} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (1)$$

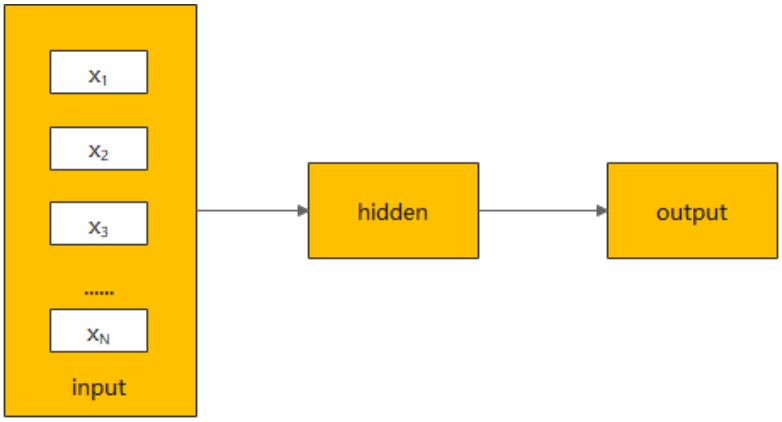

Figure 3. FastText model.

The value obtained after average operation of multiple sets of x_i is taken as the document information, and then the result is assigned to the second model part. Since classification is the final goal of the model, a common multi-classification classifier Softmax is selected to establish a mapping from the second part to the third part, and its loss function is as follows:

$$\text{loss} = -\frac{1}{N}\sum_{i=1}^{N} x_i \sum_{j=1}^{M} 1(y_i = j) \log (y'_{ij}) \tag{2}$$

$$y'_i = \text{softmax}(\text{hidden}_i) \tag{3}$$

Finally, normalization is carried out.

3.2 Improved FastText text classification method

This paper proposes an improved FastText text classification method based on text classification. It does not need to select appropriate feature parameters, and can easily pick out abnormal data from power grid asset data for classification, which greatly improves the operation efficiency.

The following process can be used to identify the authenticity of the text data of power grid assets. Figure 4 is the workflow.

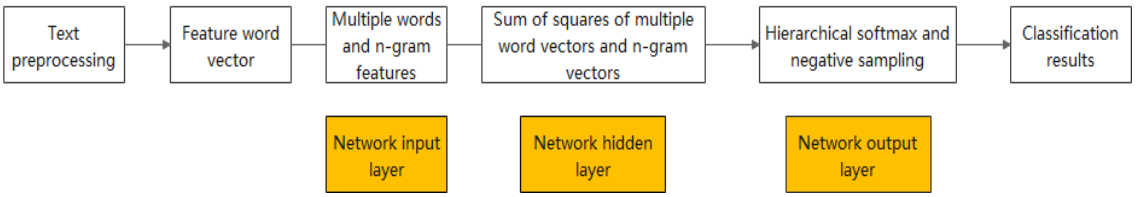

Figure 4. The improved FastText.

(1) Text input

The FastText model inputs a batch of documents, each consisting of a sequence of lexical indexes, and the output layer generates probability values indicating the likelihood that the document will belong to a category. The word and the set of word vectors are projected onto the hidden layer, which is ultimately projected onto the tag. Almost all network inputs rely on simple linear combinations, but they cannot fit all functions, so nonlinear activation functions are used here. The FastText model principle is not much different from CBOW in Word2Vec[9, 10]. The only difference is that the former targets multiple words and word vectors, while the latter targets the context of the word. The whole process is actually word segmentation, deleting redundant words and invalid numbers and so on.

(2) Vector features of generated words

In FastText model, the low-latitude vector is related to each word. As a word bag-based classification model, FastText

captures the context word order relationship in the document text through N-gram features to deepen the understanding of the meaning of the document. In order to solve the problem of large vocabulary and difficult calculation, FastText cleverly adopts the Hash bucket method, which hashes all the features captured by N-Grams into the bucket, and n-Grams in the same bucket share a word vector.

Word2vec allocates a vector to each word in the constructed word set, thus constituting the word vector. Using this process, we can discard its meaning characteristics, for example, "tower" and "towers", Most of them are the same. In other words, they look the same, but understood from traditional word2vec, because they have their own ids, the underlying implications will be ignored. A way to solve this problem is to use n-grams as a word, such as "tower", to treat n as three, so all trigrams are: ""to", "tow", "owe," "wer er >", "", the symbols "<" on behalf of the word formation in front of the composition, " >" on behalf of root formation. Therefore, trigram was used instead of "pole and tower" for description.

(3) Generate classification model

The negative logarithmic likelihood function of the classification model can be expressed by equation (4):

$$-log\ P(\ y\mid x\ ) = -\frac{1}{N}\sum_{n=1}^{N} y_n log\ (f(BAx_n)) \tag{4}$$

In the above equation, x_n is the independent variable, and the multiplication with A is equivalent to extracting the desired word from the thesaurus, and the obtained result is multiplied by A coefficient B. Finally, the classification result can be easily obtained by combining the corresponding word vector and using the characteristics of linear classifier, which is called softmax function.

$$\mathrm{f}(\mathrm{z_j}) = \frac{\mathrm{e}^{\mathrm{z_j}}}{\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{e}^{\mathrm{z_i}}} \tag{5}$$

If *O(kh)* is represented as time complexity, *k* is represented as the number of all categories and *H* stands for the scope of the entire text. In the process of operation, the probability value of all words should be obtained through softmax function, and Max value should be sought on this basis. Due to the large amount of input, the whole process is relatively complex and time-consuming. In order to optimize the training time, hierarchical Softmax method is adopted, and hierarchical Softmax is combined with Huffman algorithm. It is a kind of tree structure, each leaf node represents a word of corpus, a word corresponds to a code value, each coding sequence can be mapped to each sequence of events, used to solve from hidden layer to output layer softmax function the problem of large computation, effectively reduce the classification model to predict the number of labels, makes one category prediction probability calculation, Computing costs have dropped significantly. Compared with other multi-class logarithmic models, hierarchical Softmax functions can greatly reduce the complexity of model training and the time-consuming operation process. According to Huffman's principle of putting the most important features first, words with high frequency in corpus are located closer to the root node than words with low frequency, which reduces path dependence and optimizes operation efficiency.

(4) Negative sampling rule

The negative sampling algorithm considers the positive path, that is, the final output target words should be captured and retained by the model, and the wrong output words should also be selected as negative samples to optimize model parameters. Because different words are interfered by many factors, it is impossible to determine in which range they will appear. Therefore, the influence of this part is studied, and the weighted value of each word is calculated to ensure that the high-frequency words will be selected as negative samples with a greater probability. In the process of model parameter updating, part of corpus is randomly selected for training, which saves the training time. For known positive samples(context (w), w), to maximize the $\mathrm{g(w)} = \prod_{\mathrm{u}\in\{\mathrm{w}\}\cup\mathrm{NEG(w)}} \mathrm{p(u \mid context\ (w))}$, The conditions are as follows:

$$\begin{aligned} p(u\mid \text{ context }(w)) &= \begin{cases} \sigma(X_w^{\top}\theta^u), & L^w(u)=1 \\ 1-\sigma(X_w^{\top}\theta^u), & L^w(u)=0 \end{cases} \\ &= [\sigma(X_w^{\top}\theta^u)]^{L^w(u)}[1-\sigma(X_w^{\top}\theta^u)]^{1-L^w(u)} \end{aligned} \tag{6}$$

w is the background word of the positive sample, context (w) is the center word of the positive sample, $\sigma(X_w^{\top}\theta^u)$represents the probability of predicting the center word w when the context is context (w). $X_w^{\top}$ is the sum of vectors of each word, and T represents the rank of transformation; θ^u is the vector given by this word, the untrained variable; $u \in w \cup NEG(w)$; L^w is the label of the word w, that is, the label of the positive sample is 1 and the label of the

negative sample is 0.

Therefore, the final loss function is $L = log\ \prod_{w \in C}\ g(w) = \sum_{w \in C}\ log\ \ g(w)$, under the rules of negative samples are:

$$\mathrm{P(w)} = \frac{[\text{counter (u)}]^{0.5}}{\sum_{\mathrm{u} \in \mathrm{D}} [\text{counter (u)}]^{0.5}} \tag{7}$$

Through negative sampling, the central word of the word order can be extracted and replaced by other words, so as to increase the number of negative samples in the corpus. In this way, the frequency of positive samples can be increased and the frequency of negative samples can be reduced. In the training process, the weight of the hidden layer of the network changes constantly, but the negative sampling only changes the weight of the low. This idea can reduce the difficulty of reducing the computational complexity of gradient.

3.3 Model to evaluate

The improved FastText has obvious advantages, which is mainly compared by equations (8)-(10). The following three indicators are Precision, Recall and F1-score. So far, most studies have used generality indicators. We abbreviate them as P, R, and F, where TP means positive for correct judgments, TN means negative for correct judgments, FP means positive for wrong judgments, and FN means negative for wrong judgments.

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN} \tag{9}$$

$$f = \frac{PR}{2(P+R)} \tag{10}$$

4. EXPERIMENTAL TEST AND RESULT ANALYSIS

4.1 The data set

The data set includes 360,000 samples of 360 types of power grid equipment. 80% of the data is used as a training set for training, 10% as a validation set, and another 10% as a test set.

4.2 Experimental results and analysis

The traditional FastText method and the improved FastText method are used to carry out experiments respectively, and the experimental results obtained by the two algorithms are compared. The experimental results are shown in Tables 1 and 2.

It can be seen from Figure 5, the average accuracy of FastText algorithm is 87.53%, the average recall rate is 85.77%, and the average F1 value is 85.75%. The average accuracy of the improved FastText algorithm is 89.13%, the average recall rate is 88.84% and the average F1 value is 88.76%, Therefore, this test is proposed to improve the sum-average method of FastText mapping layer to sum-average method of squares, and improve softmax to hierarchical softmax and negative sampling in the output layer. The test results show that this method can greatly improve the results of text classification.

Table 1. Test results of FastText.

	Precision	Recall	F1-score	Number of test data
Voltmeter	0.7624	0.8953	0.8235	100
Voltage regulator	0.5909	0.7471	0.6599	100
Voltage transformer	0.8571	0.9767	0.913	100
Voltage monitor	0.7615	0.954	0.8469	100
...	...	...	...	...
Weighted avg	0.8753	0.8577	0.8575	36000

Table 2. Test results of improved FastText.

	Precision	Recall	F1-score	Number of test data
Voltmeter	0.713	0.8283	0.7664	100
Voltage regulator	0.5372	0.6633	0.5936	100
Voltage transformer	0.9167	1	0.9565	100
Voltage monitor	0.8364	0.9293	0.8804	100
...	...	...	...	...
Weighted avg	0.8913	0.884	0.8876	36000

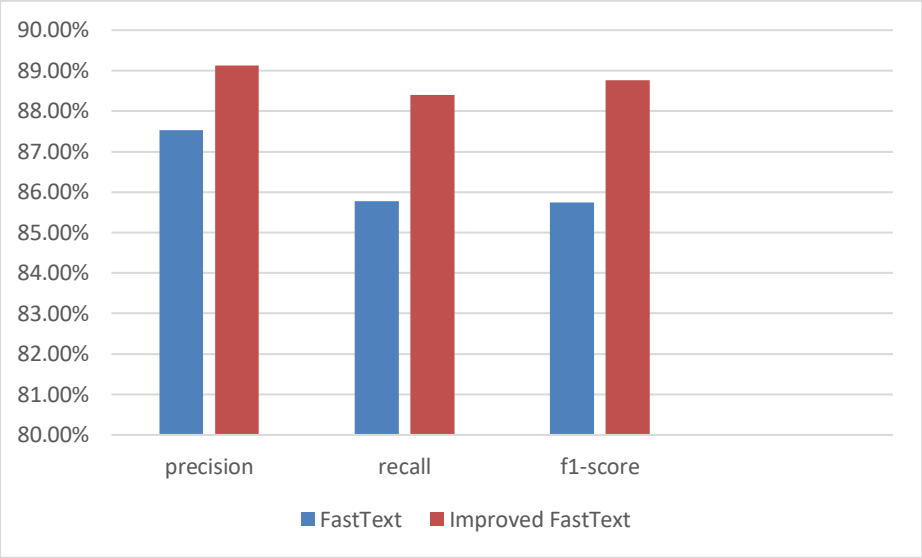

Figure 5. Comparison of experimental results.

5. CONCLUSION

In view of the traditional FastText method, we made an improvement on the two aspects of network hidden layer and output layer. The input value is obtained by the results of average sum again after the first as the input values in square sum then average operation effect is good, and the output layer adopts hierarchical softmax and negative sampling method, reduces the computation complexity. It saves the operation energy, and finally uses the values of P, R and F to compare the traditional and improved effects through the bar chart. The conclusion is that both have been greatly improved. Although the results obtained in this study are generally satisfactory, it is still necessary to combine with other models for comparison and strive for a higher level of research on the classification of state grid asset data.

REFERENCES

[1] Zhang, W. and Zhang, H. X., "Naive Bayesian ensemble classifier with attribute weighting," Computer Engineering and Applications, 46(29), 144-146(2010).

[2] Li, M. J., "Research on method of feature selection in text combined with word frequency in class," Application Research of Computers, 31(7), 2024-2026(2014).

[3] Chen, J. H., [Research on Feature Selection Method for Chinese Text Classification], Northwest Normal University, Lanzhou, Master's Thesis, (2012).

[4] Zhang, C. Y. and Wang, J., "A new weighted naïve Bayesian classification algorithm," Microcomputer Information, 26(10), 222-224(2010).

[5] Ma, S., [Research on Key Technologies of Web Text Data Warehouse Preprocessing], Xidian University, Xi'an, Master's Thesis, (2011).
[6] Song, G., Ye, Y., Du, X., et al., "Short text classification: A survey," Journal of Multimedia, 9(5), 200-212(2014).
[7] Hung, L. C., Lin, H. P. and Chung, H. Y., "Design of self-tuning fuzzy sliding mode control for TORA system," Expert Systems with Applications, 32(1), 201-212(2007).
[8] Bahdanau, D., Cho, K. and Bengio, Y., "Neural machine translation by jointly learning to align and translate," arXiv:1409.0473, (2015).
[9] Mikolov, T., Kopecky, J., Burget, L., et al., "Neural network based language models for highly inflective languages," IEEE Inter. Conf. on Acoustics, Speech and Signal Processing, 4725-4728(2015).
[10] Mikolov, T., Kombrink, S., Burget, L., et al., "Extensions of recurrent neural network language model," IEEE International Conference on Acoustics, Speech and Signal Processing, 5528-5531(2012).