# Pronunciation-based Convolutional Networks for Text Classification

**Xiaojie Xu**
261031809

**Dun Yuan**
260964392

**Huiyi Guo**
260827802

## Abstract

The use of pronunciation has been found to improve performance on different natural language processing (NLP) tasks. Inspired by the work of character-based models and the models combining words embedding and pronunciation embedding, and their good performance, this paper proposed a new model which is pronunciation-based. This paper aims to investigate how well a model could perform if it only relies on pronunciation as the input and since it is similar to character-based methods in some way, so we mainly compare our model with the character-based model to measure the efficiency. Based on our datasets for machine learning and deep learning models, we present the optimal combinations of these methods. These combinations should be relevant to a variety of datasets and classifications. On the other hand, our results show that pronunciation has little effect on performance of one dataset. Polyphonic words and limited dictionary, as well as limitations with our experiment, are possible reasons.

## 1 Introduction

In recent years, text classification has become a prominent application of natural language processing learning research due to its commercial benefits. Typically, text classification is a method of analysing text and assigning pre-defined tags or categories to it depending on its meaning. Because of the importance of text classification, researchers have explored a variety of methods to increase its accuracy in the past. In previous works, researchers tend to concentrate on the relationship between writing and its meaning. However, this approach ignores other types of linguistic expression that has intimately linked with meaning.

Inspired by the idea that languages are originally generated by human being in talking instead of writing, we address this unresolved question by focusing on pronunciation, which is another type of linguistic expression, that is inextricably related to meaning (Fitch, 2016), and illustrate how and to what extent it helps to improve text classification. Specifically, we use pronunciation (eg. IPA, a method that describes the sounds of spoken languages) instead of writing for text classification.

We use a variety of datasets, each of which may have different properties (some will use more written terminology, while others will use more colloquial phrases), and text classification by pronunciation may stand out in some data.

Furthermore, we would like to set up a combination of preprocessing methods with pronunciation that works well on our datasets and can be used to other datasets. We design rounds of comprehensive tests to compare and contrast their effects on a model's classification accuracy. The goal of this comparison is to see which strategies are the most effective and which combinations are the best.

## 2 Related work

Text classification is a subject of natural language processing that is currently undergoing extensive research (Christopher D Manning and Schutze, 1999). While the use of character-level data and ConvNet (Zhang X, 2015) and the use of bi-gram alphabet (Elghannam, 2021) has been successfully used to improve text classification; Pronunciation studies are frequently irrelevant to text classification (Sefara et al., 2017). Only a little amount of study has been done on employing pronunciation to improve text classification.

It was found that word embeddings could be improved by using both writing and pronunciation (Zhu, 2018). The authors investigated a method that incorporates speech data into training in order to completely apply both speech and writing to meaning. By using two models (CBOW

Based model and Skip-Gram Based model), integrating pronunciation information increases the performance of the word embedding model. Despite this, it does not display the outcome based just on pronunciation information or utilising various preprocessing approaches.

Moreover, Yang and others have also studied how pronunciation improved word embeddings (Yang, 2021). They also looked at the relationship between word structure and meaning. However, they concentrated on sentiment analysis and exclusively analyzed the Chinese. Their findings did not compare the impact of pronouncing or different preprocessing methods to text classification.

## 3 Method

We propose and investigate the advantages and disadvantages of a collection of preprocessing approaches with pronunciation as the major purpose of this study. We must integrate specific machine learning classifiers and deep learning architectures that can directly reflect the performance of our methods in order to compare and contrast their impacts on final prediction accuracy.

### 3.1 Datasets

We use diverse datasets for better performance and more accurate comparison.

**AG's News:** This dataset has 4 classes, 120,000 train samples, 7,600 test samples.

**DBPedia:** This dataset has 14 classes, 560,000 train samples, 70,000 test samples.

**Amazon Review Full:** This dataset has 5 classes, 3,000,000 train samples, 650,000 test samples.

**Amazon Review Polarity:** This dataset has 2 classes, 3,600,000 train samples, 400,000 test samples.

We also produce an dataset based on Cornell Movie Dialogs Corpus (Danescu-Niculescu-Mizil and Lee, 2011). This corpus contains conversations extracted from movies. We would like to test on this corpus because it is generated from conversation, so it is possible that pronunciation-based models will have better performance on it. However, since we are focusing on text classification task, it is hard to label conversation sentences in this corpus. We tried to label according to whether the movie is "drama" or not by matching the movie genre data on IMDB, but the results is still too poor because of the high difficulty of the task.

### 3.2 Converting text into phoneme

After all, we need to classify text based on pronunciation, thus we'll need to convert text to phonemes. We use The CMU Pronouncing Dictionary (Carnegie Mellon University) as the standard to convert English text into the phonemes, which includes 39 phonemes and totally 84 symbols if counting the varia of the lexical stress. Each phoneme will be represented as a one-hot vector in general. Then, to work on text classification tasks, we use just English phoneme as input and use the ConvNet model for training and prediction. We can improve the raw data that will be input to the model by converting text to IPA. In order to make the pronunciation more easily processed, in the code we use ARPABET (Klautau, 2001) used in CMU Pronoucing Dictionary instead of IPA.

### 3.3 Methods of Preprocessing

Preprocessing has long been a crucial step in natural language processing (NLP). It converts text into an a more digestible form, allowing machine learning algorithms to perform better. For example, it decreases the noise level. The goal of preprocessing would be converting data into vectors with one-hot encoding.

**Baseline:** The baseline model entirely follows the preprocessing part described in (Zhang et al., 2016). Therefore, every character will be transformed into an one-hot encoding vector. The alphabet consists of 26 English characters, 10 digits, 33 other characters and the new line character.

**Phonemes:** We followed the method of (Zhang et al., 2016), but replace the preprocessing of characters into the preprocessing of phonemes. The sequence of phonemes generated from previous step will be transformed into a sequence of vectors with a fixed length. Any phoneme or character that is not in the alphabet will be quantized as all-zero vectors. The original alphabet consists of 84 phonemes:

```
AA, AA0, AA1, AA2, AE, AE0, AE1,
AE2, AH, ..., Y, Z, ZH
```

With following techniques or methods applied, the alphabet might be modified.

**Ignore stress:** Stress is frequently appeared in pronunciation rules and it varies from language to language, tone to tone, and context to context. In most circumstances, removing the stress has no effect on interpreting the real meaning of a certain word. However, the stress might sometimes assist the model in making more exact classification. We would like to see how stress affects pronunciation base on the result of experiments. In the ARPABET, number in phonemes represents for stress and it shows if a certain syllable has stress on it. By removing the number from ARPABET phonemes, the phoneme alphabet consists of 39 phonemes:

```
AA, AE, AH, AO, AW, AY, B,
CH, ..., Y, Z, ZH
```

**Add zero vectors between words:** Phoneme, unlike written text, do not have special character to separate words. We need to add zero vectors between the pronunciation symbols of words to better differentiate them. In general, this method should reduce the noise level and enable the model to analyze data more efficiently.

**Add numbers and special characters:** While we can convert words to phonemes, numbers and non-space special characters are still valuable in text classification process, especially when analysing the sentiment of a certain corpus. After adding 10 digits and 33 other special characters, the new alphabet contains 126 items.

```
AA, AE, AH, AO, AW, AY, B,
CH, ..., Y, Z, ZH, 0123456789
-,;.!?:'"/\|_@#$%^&*~`+=<>()[]
```

**Spell check:** In raw corpus, there exists misspelled words. Removing them can reduce the noise level and and improve the model's performance in the vast majority of circumstances. However, due to the dictionary's restrictions, some meaningful terms may be removed. Therefore, we tried to use a spell checker to find the most possible word when we encounter any word that is not in the dictionary. It is possible that the word is not misspelled but an idiom or slang, which cannot be found in dictionary. By using the spell checker, it is still possible to find a word with same or similar pronunciation, and helps keeping the original information of the word.

### 3.4 Model

The focus of this work is not on the model; we choose to use the same Convolutional neural network (CNN) to evaluate the preprocessing methods with prounciation by comparing their classification accuracy.

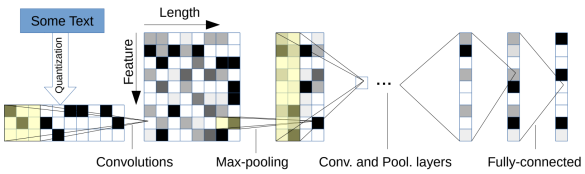
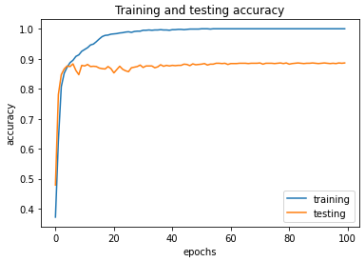



Figure 1: Illustration of the model. Figure adopted from (Zhang et al., 2016)

**Convolutional neural network (CNN):** As shown in Figure 1, this model is nine layers deep with six convolutional layers and three fully-connected layers (Zhang et al., 2016). We used the code of this model implemented by others on Github (Ardalan and Bazarov) as the baseline and modified it to fit our needs, i.e. to be able to take the phonemes as the input.

## 4 Results

In this section, we evaluate the performance of methods. We also present the combination of these methods that performs the best on our datasets.




(a) Accuracy


(b) Loss

Figure 2: Training and testing results of baseline model on AG's news dataset

Figure 2 shows the accuracy and loss value over epochs. The experiment uses baseline model and AG's news dataset. According to Figure 2, it is obvious that the training process should be early

stopped at around 9 epochs. After 9 epochs, although the training metrics continue to improve, decreasing testing accuracy and increasing testing loss imply that the model is becoming overfitted on the training dataset. Therefore, we applied early stopping technique while training to avoid overfitting. For different datasets, the number of epochs is also different.

Table 1: Testing accuracy for all models and datasets. "Ph" stands for "phoneme". "Sp" stands for "Space". "NoS" stands for "No Stress". "Char" stands for "Number and Special Characters"

| Model | AG | DBP. | Amz.F | Amz.P. |
|---|---|---|---|---|
| Baseline | 87.8 | **98.8** | | **95.7** |
| Phoneme | 85.9 | 97.7 | | |
| Ph Sp | 87.2 | 97.9 | | |
| Ph NoS | 86.6 | 97.4 | | |
| Ph Char | 87.7 | 97.8 | 60.8 | 95.3 |
| Ph Sp Char | **88.3** | 98.2 | | 95.5 |
| Ph NoS Char | 86.8 | 97.6 | | |

Table 1 shows the testing accuracy of models on different datasets. The best performance for each dataset is set to bold. Due to limited time and computer power, we cannot finish the experiment on Amazon Review Full and Amazon Review Polarity dataset. However, it is still proved that our model and processing techniques run successfully on these overwhelmingly large datasets.

Since there are 7 kinds of models tested in our experiment, we would like to focus on evaluation of preprocessing decisions and pronunciation-based decision, then we can find the model with best combinations.

### 4.1 Preprocessing Evaluation

We begin by looking at the preprocessing methods that improve the accuracy. Their impacts on the final classification accuracy are in Table 1. Except the no space method, others improve the prediction accuracy of pronunciation classification. In AG dataset, the average of pronunciation processing improvements is approximately 1%. In DBP dataset, the average of pronunciation processing improvements is approximately less than 1%.

### 4.2 Pronunciation Evaluation

Instead of using traditional written text for text classification, we use phoneme for text classification. At the beginning, baseline's value appears to be slightly more accurate than phoneme's (1% or 2% lower than the baseline). However, with the use of preprocessing and combinations of preprocessing, accuracy is enhanced much further.

### 4.3 Best Combinations

We chose the best combination that produce the best classification results on our datasets after comparing all combinations of preprocessing with pronunciation. On a CNN pronunciation classifier, we employ space and number/special characters for preprocessing. It obtains a 98.2% accuracy for DBP dataset and 87.7% for AG dataset and 95.5% for Amazon P. dataset, which is the best among all preprocessing combinations. The excellence performance of this combination highlights the positive influence of space and Number & Special Characters. The success of this combination should not be considered as a coincidence as we apply it to CNN. When we apply these methods to other models, we should get similar results.

## 5 Discussion and conclusion

In conclusion, this method is straightforward. We convert the written text into phoneme, then each preprocessing approach is applied on all datasets. It also evaluates every feasible subset of the preprocessing approaches offered in order to determine their overall contribution.

Even though text classification with pronunciation did not perform as well as standard text classification at first, accuracy improved after a series of preprocessing steps. Most preprocessing strategies (except no space method) improve the pronoucication classifier's performance in some way. Although the accuracy of pronunciation classification of DBP and Amazon Polarity datasets after preprocessing do not exceed the accuracy of the baseline text classification, the pronunciation categorization remains legitimate. Because the two values are still quite comparable, and we believe that adding further preprocessing would enhance accuracy even more.

However, with Amazon Review Full and Amazon Review Polarity, we were unable to collect complete accuracy results. This is due to the limitation of our available computer power and resources, as well as the fact that the Amazon data is too large for us to adequately train these two datasets multiple times.

Moreover, the yelp dataset did not perform as

we expected, and the data's accuracy barely improved after epochs during the training, and with the training time being very long and resources being limited, we did not be able to training multiple times to investigate the causes, but based on the results we had, we think the potential reason might be the learning rate too large that the gradient jumped back and forth from one side to another side around the minimum point, or too small that the gradient updated so slow making the accuracy staying almost unchanged.

Our research used a variety of datasets to gain a better understanding of the effects of preprocessing with pronouciation on text classification. The decision to combine some of these methods is correct for most of the cases, and our proposed best combinations for similar text classification problems are strongly encouraged. However, actual data and result evaluations for various application domains, as well as the particular form of their textual data, should be more motivating.

### 5.1 Limitations and Potential Improvements

**Limited dataset** As discussed in Part 3.1, we try to produce a dataset from conversation corpus but failed at finding suitable classes and labels for the training data. In future work, experiment on a larger variety of datasets is needed. It is highly possible that pronunciation models will have much better performance on corpus that is related to speech or conversation.

**Limited model** We only tried one model in this project. We could try other models (e.g. Naive Bayes, Logistic Regression, etc.) and compare them as a part of out research.

**Other language** Since the pronunciation of each language carries the meaning of that language, this method should be applicable to other languages as well. We could potentially apply these techniques to other languages. For example, we could transform words into pinyin (a type of phoneme comparable to IPA) before letting the model interpret and analyse them in Chinese. In the same way, we can utilise romanization in Japanese.

**Preprocessing in written text** We can do preprocessing in written text before converting the it into phoneme to improve its accuracy. For example, ignoring irregular token. However, in light of the pronunciation dictionary, we should avoid using preprocessing such as lemmatization, and other approaches that change the word directly because they may also change the pronunciation.

## 6 Statement of contributions

All team members worked jointly during the literature review process and contributed equally.
Implementation:

Xiaojie takes setup & model.
Dun takes finding datasets & preprocessing.
Huiyi takes classifiers & draft of the report.

Final review and formatting were done jointly by all team members.

## References


Ardalan and Evgeny Bazarov. Baseline model.

Carnegie Mellon University. The cmu pronouncing dictionary.

Christopher D Manning Christopher D Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Fatma Elghannam. 2021. Text representation and classification based on bi-gram alphabet. *Journal of King Saud University - Computer and Information Sciences*, 33(2):235–242.

W. Fitch. 2016. Sound and meaning in the world's languages. *Nature*, 539:39–40.

Aldebaro Klautau. 2001. Arpabet and the timit alphabet. *URL: https://web. archive. org/web/20160603180727/http://www. laps. ufpa. br/aldebaro/papers/ak_arpabet01. pdf*.

Tshephisho Joseph Sefara, Madimetja Jonas Manamela, and Thipe Isaiah Modipa. 2017. Web-based automatic pronunciation assistant. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, pages 112–117.

Xie-H. Cheng G. et al Yang, Q. 2021. Pronunciation-Enhanced Chinese Word Embedding. *Cogn Comput*, 13:688–697.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification.

LeCun Y Zhang X, Zhao J. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems.*, 28:649–57.

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