Character-level Trigram Natural Language Processing from Scratch

Francisca Mulebya
26703580@sun.ac.za

Abstract

Natural Language Processing is still a developing field and it is useful in many contexts associated with human language (written and spoken). In written cases, we can use a probabilistic model to generate and make predictions using characters, words or something in between. In this work, we use a character-level based approach though the probabilistic model encompasses more than 1 character but not necessarily a word for modelling. Data was obtained from Wikipedia for five languages.

1. Introduction

The goal of Natural Language Processing (NLP) is to generate, generalize and make predictions based on the probabilistic model learned and employed in some validation and test datasets. The usefulness of such field is mainly on machine-human interactions such as translations, chatbots (with multiple or specific purposes), language classification and many others like automatic text generation (Daniel & H.James, 2023). This AI task has greatly contributed to the breakthrough of Large Language Models (LLMs) and Robotics (in Speech case).

1.1. Problem Statement

In NLP Modelling, there are some major problems namely generative process probabilities, semantics and sparsity issue.

To model the data, we need to have the probability of an input unit (either a word, character or subword) which has to be non-0 so that it is possible to generalize on testing data. In practice to find such probabilities, the context of '*Maximum Likelihood Estimate*' is used to compute the classical probability based on counts. In this case, data not in training will have 0 probability which means it will not appear in generalization, generation or prediction.

To counter the above sparsity issue, we can use character-level trigram probabilities in modelling despite the fact that word is the basic unit of meaning. The use of characters alone can be used but the context might suffer semantics issues (meaning).

1.2. Motvation

Since the dataset used comprises of five languages (English, Xhosa, Zulu, Afrikaans and Dutch), it is best to use n-gram characters in languages with rich, inconsistent and complex morphology such as Xhosa and Afrikaans languages as discussed in (Bojanowski et al., 2017). The choice of n depends on the task at hand. Also for multilingual context, it might be beneficial to have a high n value, especially in the case of similar structured languages to improve generalization performance.

Using character-based trigram, the model can perform well even in data subjected to inconsistent formatting and spelling errors. Also in small training set, this approach could be very useful. (Zhang et al., 2015) suggest a character-level as efficient for text classification without considering meaning; although in the modelling uses Convolutional Networks. Also, not normalizing word cases yields better results. The idea of character-level based approach is useful in word segmentation learning and works well even in deep natural language models (Chung et al., 2016).

1.3. Approach

Given a character-level trigram defined as $x_1x_2x_3$; In general, the character-level trigram probability is given as follows:

$$P_{MLE}(x_3|x_1, x_2) = \frac{C(x_1, x_2, x_3)}{C(x_1, x_2)}$$

where C is the count function over the whole corpus.

In summary, to smoothen the character-level trigram probabilities the following approach was considered to avoid *vanishing probabilities*:

Add-k Smoothing

$$P_{MLE}(x_3|x_1, x_2) = \frac{C(x_1, x_2, x_3) + k}{C(x_1, x_2) + k|\mathcal{V}|};$$

whereby $\mathcal{V}$ is the vocabulary/set of word types.

This method is useful in some tasks such as text classification; I considered only this in the end as the unigram of a character (in multilingual context) is not important in learning so it is enough to learn only from the trigram and bigram.

Kneser-Ney Smoothing This is an advanced and complex interpolation approach that used the idea of Absolute Discounting (assigns non-0 probabilities in different weights) with more hyperparameters to tune and learn from less contexts.

2. Data Processing

In this sub-task, text normalization is done in training and test sets based on normalized structure of the validation dataset as seen below.

2.1. Text Normalization

The following are the observed data preprocessing

- Numbers: all digits are transformed to 0
- New Sentence/line: new sentences are separated by \n
- Character cases: the corpus takes lower cases only
- Punctuations: remove all the punctuation marks such as .,;:- were all ignored by replacing with an empty space
- Accent characters
- Contraction: from the language I could understand, I did not find the terms of the form 'isn't' etc.

The resulting normalized corpus was as follows:

3. Language Modelling

As discussed earlier the model will use $n-$gram on a character-level (subwords of 2 and 3 characters) which are taken from the corpus obtained after data processing. Add-k-Smoothing approach is used in calculating the trigram (and bigram as well) probabilities.

4. BPE for Language Similarity

Word segmentation for rare words translation(Sennrich et al., 2015)

5. Results

This results imply that the trigrams proabilities are likely correct as the trigram 'the' has the highest probability of almost 0.538 far off from the other two 'tha' with probability 0.065 and 'the' 0.036.

```
'thn', Probability: 0.0036431325621592873
'thi', Probability: 0.035341045073793376
'tht', Probability: 0.00018614545938040153
'thr', Probability: 0.026778353942294907
'ths', Probability: 0.002579444222842707
'thc', Probability: 0.00023932987634623057
'th ', Probability: 7.977662544874351e-05
'thm', Probability: 0.00034569871027788856
'th-', Probability: 7.977662544874351e-05
'thp', Probability: 0.00013296104241457252
'tha', Probability: 0.06533705624252094
'th'', Probability: 0.0007711740460045206
'thw', Probability: 0.0005584363781412046
'thu', Probability: 0.00188804680228693
'thé', Probability: 7.977662544874351e-05
'thl', Probability: 0.00045206754420954663
'thb', Probability: 7.977662544874351e-05
'the', Probability: 0.5377742321499801
'thd', Probability: 0.0005052519611753756
'thf', Probability: 7.977662544874351e-05
'th ', Probability: 0.08177104108496211
'thã', Probability: 0.00034569871027788856
'tho', Probability: 0.030022603377210477
'thy', Probability: 0.0015689403004919558
```

Figure 1. th results in english training dataset.

The results on perplexity shows the model is not performing well especially with Zulu and Xhosa languages (hundreds) while it at least perform with almost 17 on english language. Due to that, the prediction made on test data classifies the text to be from English language.

6. Conclusion

The work is not 100% complete especially on the generative process part and BPE; also the results obtained in the Perplexity part are not as I expected but with a time limit, this is what I have done so far.

7. Reference

References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5: 135–146, 2017.

Chung, J., Cho, K., and Bengio, Y. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*, 2016.

Daniel, J. and H.James, M. Speech and language processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft, 2023.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

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