Building Your Own Word Embedding For Sentimental Analysis

marc@cinnamon.is, matthew@cinnamon.is

December 2019

1 Introduction

For machine learning to be applicable to text, the word-embedding task concerns with the transformation from word to a numerical representation. Although ASCII or Unicode encoding are already numerical, word embedding machine learning models may have additional requirements:

- Uniform: Same sized words, else the data for learning the representation would be massive and so does computational cost, especially for deep learning-based architecture.
- Semantic representation: The word should enforce and/or contain encoding of information of its semantic or use case.
- Low-label cost: Unsupervised methods or self-supervised are always preferred since the transferring cost between languages and contexts is low.

In this challenge, you will learn about the learning process of word-embedding, of which, the classical methods are:

- Bag-of-words [Zhang et al.2010].
- Skip-gram[Mikolov et al.2013].
- GloVe: Global Vector for word representation [Pennington et al.2014].

Since machine learning methods in general are data-driven, so does word-embedding, a good amount of data is required for the training. **Data is the heart of machine learning model**.

The cleaning and polishing of these data can not be overlooked. Therefore, we have made it an official step in This assignment (See Section 2, 3, and Figure 1). Followed by that, Section 4 focus on running word-embedding on these data. In each sessions, only references will be given, you decide what to do with them (And please ask your mentors if there are any concepts or any difficulties).

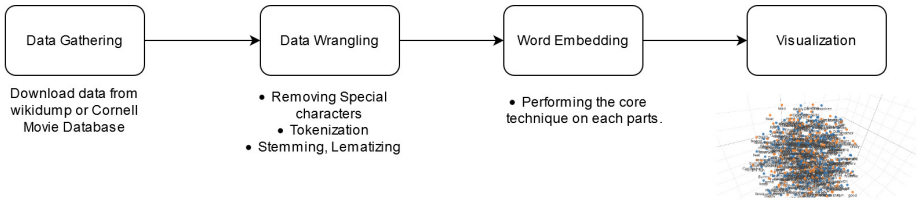

Figure 1: Word Embedding Steps

2 Data Gathering

For now, we will use either:

- Wikidump `https://dumps.wikimedia.org/backup-index.html`
- Cornell Movie Dialog Corpus `https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html`

On either of the dataset, each of the file will contain only a small portion of texts. Your task is providing the clean set amount of text (extracting text from HTML or XML, etc).

3 Data Cleaning

During our treatise on words, engineers tend to find some characters are undesirable: teen code, typo, etc. Indeed, removing them should be counted as a task in word processing. To this:

- First list out all the distinct characters available in the set.
- Identify the noisy characters
- You can either choose to remove only the character, or the whole word.

Then write a tokenizer by splitting each corpus using white space, tabs, any non-alphabet characters. Finally, to reduce number of words, normalize them by either:

- lower casing, stemming, lematizing. (For this part, please see [Perkins2014], chapter 2)
- Use dictionary or WordNet lookup.

4 Word Embedding

Use already available frameworks to perform word embedding on these cases. Before applying the codes, we hope that you would be able to answer the following questions:

- How does each method take into account the context of each word?
- What property of those mathematical models?
- What are the key distinction between the 2 models: Skip gram and Glove?

5 Qualitative Testing

Visualize the the embedded words using either Tensorboard or Scikit learn T-SNE visualizer.

References

[Mikolov et al.2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

[Pennington et al.2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 37, pages 1532–1543, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Perkins2014] Perkins, J. (2014). *Python 3 Text Processing With NLTK 3 Cookbook*.

[Zhang et al.2010] Zhang, Y., Jin, R., and Zhou, Z.-H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52.