Using GloVe embeddings for news article classification

1st Simon du Toit
Applied Mathematics Department
Stellenbosch University

***Abstract*—In this project I use GloVe embeddings to create a topic classifier for news articles. I show that the embeddings capture semantic relationships between the words and the model successfuly classifies articles.**

I. Introduction

I first describe how I process the data, construct my GloVe word embeddings and train a topic classifier. I then explain my implementation in detail and give the tuned hyperparameters of my models. Finally, I present the results of embedding the words and applying the topic classifier to the test set. I also discuss the implications of these results.

II. Design

In this section I describe how the data is processed, how words are embedded using GloVe (Global Vectors for Word Representation) and how these embeddings are used to classify news articles into topic categories.

A. Data Processing

The data consists of the AG News dataset, a collection of single sentence news article descriptions divided into four topic categories: World, Sports, Business and Sci/Tech. The dataset is provided by Xiang Zhang et al. [2].

I extract and normalize the article description sentences. I replace full stops and hyphens with spaces. This is also done for backslashes, which appear between some concatenated words in the data. I lowercase all characters, apply NFD normalization and replace all digits with 0. Finally, I remove all characters which are not alphanumeric and remove leading and trailing spaces.

B. GloVe Embeddings

The unique words in the normalized training data are stored as the vocabulary. I create a vector embedding of each word in the vocabulary using GloVe [1]. The GloVe algorithm proceeds as follows:

1) Slide a window over each sentence of the text data.
2) Count the occurrence of every word pair (w_c, w_o), where w_c a central word and w_o is a context word of the window.
3) Initialize an embedding matrix $\mathbf{u}$ for the central words and $\mathbf{v}$ for the context words. Also initialize a bias vectors b and c for central and context words respectively.
4) Train the embedding matrices and biases with backpropagation on the loss function defined as:
$$J(\theta) = \sum_{c=1}^{V} \sum_{o=1}^{V} h(C_{c,o}) \left(f_\theta(c,o) - \log C_{c,o}\right)^2$$
Here $h(x)$ weighs word pairs by their counts and $f_\theta(c,o)$ is:
$$f_\theta(c,o) = \mathbf{u}_c^\mathsf{T}\mathbf{v}_o + b_c + c_o$$

Words can then be embedded using either of the two embedding matrices $\mathbf{u}$ and $\mathbf{v}$, some combination of the two.

C. News Article Classification

I perform article classification using a multilayer perceptron. The neural network has three hidden layers with ReLU activation and 256, 512 and 256 nodes respectively. For a given article description, I convert it to a vector by averaging the embeddings of the words in the sentence and feed this as input to the model. Words which do not appear in the training vocabulary, and thus do not have embeddings, are simply ignored. The model outputs softmax probabilities over the four classes corresponding to the different news categories and is trained using cross-entropy loss.

III. Implementation

In this section I explain how GloVe embedding algorithm and article classification network are implemented in Python.

A. GloVe Embeddings

Before applying the GloVe algorithm, the description sentences are turned into lists of words. I store the vocabulary words in a list, but I also construct a dictionary for the vocabulary index of each word for fast indexing. I use a window of size 7 (that is, a central word surrounded by the three nearest words on each side). The word pair counts are stored in a dictionary. The embedding matrices and biases are trained using PyTorch. Training these matrices is computationally expensive, so I adjust the hyperparameters with manual tuning on a validation set made from 20% of the training set. I settle on a vector embedding size of 100 and train for 10 with the Adam optimizer. I employ minibatching during training and shuffle the batches on each epoch. Training appears to be faster with large batches, so I use a batch size of 1024. For actually embedding words, I get the best results by adding the $\mathbf{u}$ and $\mathbf{v}$ embeddings as suggested in the original GloVe paper [1].

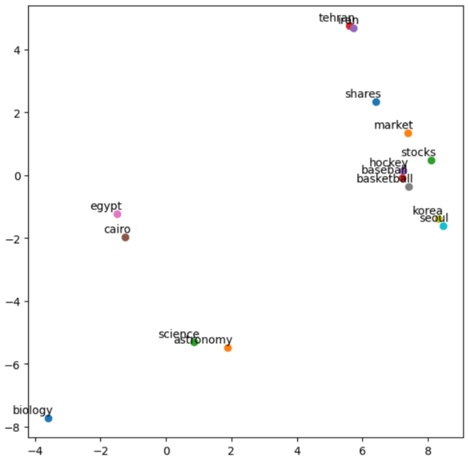

Fig. 1. GloVe Word Embeddings

B. *News Article Classification*

The classification network is implemented in PyTorch. As for the embeddings, the hyperparameters are adjusted with manual tuning on the validation set. The network is trained roughly until the loss stops decreasing, which occurs after 3 epochs. I use the Adam optimizer and a batch size of 64.

IV. Results and Insights

In this section I visualize my word embeddings and present the results of my trained classification model.

A. *GloVe Embedding*

A handful of word embeddings are visualized in a two-dimensional plane using t-SNE (t-distributed stochastic neighbor embedding). This visualization is presented in figure 1. It is clear that the embeddings manage to capture some semantic information about the words. Words relating to science, sports and finance form distinct clusters in the embedding space. The embeddings also put countries close to their capital cities. I also measure the distances in the high dimensional space. For example, the distance between science and astronomy is 7.36, compared to a distance of 7.83 between astronomy and market. Overall the distances reflect the relative proximities in the visualization.

B. *News Article Classification*

The final model achieves a classification accuracy of 81.1% on the validation set and 81% on the test set. This shows that the simple classifier is able to make good predictions based on the word embeddings and generalizes well even though unseen words are ignored. The confusion matrix is shown in figure 2. By far the most confusion is between the Science/Tech and Business topics, perhaps due to the presence of articles describing the impact of new technology on the financial world.

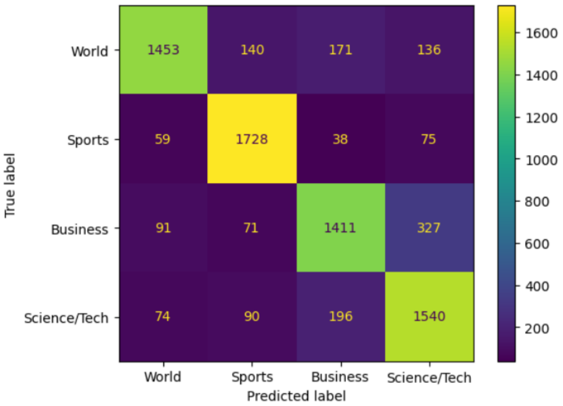

Fig. 2. Confusion Matrix on Test Set

V. Conclusion

In this project I used GloVe embeddings to create a topic classifier for news articles. I showed that the embeddings capture semantic relationships between the words and that the model successfully classifies articles. The final model achieves an accuracy of 81%.

References

[1] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[2] Xiang Zhang, Junbo Zhao, and Yann LeCun. *Character-level Convolutional Networks for Text Classification*. 2016. arXiv: 1509.01626 [cs.LG].