GloVe: Technical Review Report

ameetd2@illinois.edu
CS410 Fall 2021

Overview

Understanding the semantics of a word is a crucial requirement for applications that involve Natural Language Processing (NLP) for e.g. query processing in search engines, text classification, chat bots etc. One of ways to capture the semantics of a word is to represent each word as a vector of real values (also known as word embeddings) in some vector space. There are many ways to generate these vector space representations for a word, in this technical report we evaluate the GloVe model and algorithm to generate these word vectors.

The report is organized as follows, the first section provides brief introduction to word embeddings and describes existing approaches to generate word embeddings at a very high level and highlights some of their limitations. In the second section we talk about motivations and the underlying philosophy of the GloVe model. The subsequent sections describes the GloVe model and the training algorithm in detail. We then show GloVe word embeddings in action using some examples. Lastly, we summarize and conclude our review.

Word Vectors and Models

Word vectors (or word embeddings or vector space representation) is a technique of representing each word in a language as a vector of real numbers of fixed dimension. The ideal word embedding should accurately capture the semantics of the word in a language and help with identifying semantically similar (or dissimilar words). However this is easier said than done, as the notion of similar vs. dissimilar is highly subjective and often depends on the context in which the word appears for e.g “apple” is used to describe a fruit in context of food, but refers to a “company” when used in the context of computing.

There are 2 broad families of models to generate word embeddings.

1. Matrix factorisation

   This method has its roots in LSA. In this method we compute a global word co-occurrence count matrix, where each entry (i,j) in the matrix represents the count of the number of times the $\{w_i, w_j\}$ pair have appeared in the entire corpus. Since this matrix can be of very high dimension, lower dimensional word embeddings for each center word w_i is generated by using matrix factorization methods like SVD / PCA.

   This method has the advantage of using all the statistics from the corpus and also a slight performance advantage because we don't need to look at the text corpus after we generate the co-occurrence matrix. But it is limited to capturing the patterns related to computing the similarity of words and also has issues with dealing with word pairs that

have high-occurrence count but have little value in terms of capturing similarity semantics e.g. words pairs containing “the” or “and” as context words.

2. Local context window

   This model only looks at the context of a center word and tries to predict the co-occuring words in the context window using a shallow neural network for e.g. word2vec using skip-gram or CBOW methods.

   This method has the advantage that it not only captures simple patterns related to similarity of the words but also captures other complex patterns for tasks like named entity recognition. The disadvantage is that it does not make use of global statistics and also the time complexity increases with the corpus size when computing probabilities of context words.

GloVe Motivation

Statistics about word co-occurrences in the corpus is the primary source of information for almost all algorithms that generate word embeddings. All algorithms try to learn the meaning of a word from these statistics and more importantly encode this meaning in the word vector representation which can be later exploited using vector arithmetics.

GloVe tries to combine the best of both, counts based model and probabilistic skip-gram model. It is based on a simple principle that relationships between 2 words (w_i and w_j) can be revealed by the ratio of their co-occurrence probabilities using various probe words (w_k).

In the paper Paddington et al. provide a simple example that illustrates this concept. Let P(w|k) be the probability that word w occurs in the context of k. So, the `P(w=”ice”|k=”solid”)` has a higher probability that `P(w=”steam”|k=”solid”)`, similarly `P(w=”steam”|k=”gas”)` is higher `P(w=”ice”|k=”gas”)`. This indicates that the ratio `P(w=”ice”|k=”solid” / P(w=”steam”|k=”solid”)` is much greater than 1, meaning there is a very high probability we will see the word “ice” in the context of “solid” rather than “steam”.

On the other hand if we have probe word like “water” that co-occurs with both ice and steam then the `P(w=”ice”|k=”water”) / P(w=”steam”|k=”water”)` is closer to 1 and we see a similar behaviour for a tortally unrelated probe word e.g. fashion. This indicates that the word embeddings (rather the difference in word embeddings) generated based on these “probability ratios” could encode interesting relationships between these words.

The GloVe Model

GloVe is a bi-linear log linear regression model with weighted least squares objective function. The objective of the model is to encode the information in the ratio of co-occurrence probabilities between 2 words using a probe word into the difference of the vector representation for the 2 words. This is achieved by making the dot product product of the 2 vectors equal to log of the count of co-occurrence of the 2 words in the corpus. The details of how GloVe generates word embeddings is described in the following sections.

Co-occurrence Matrix

The first step in training the GloVe model is computation of co-occurrence matrix X from the corpus. It is V x V dimension matrix (where V is the size of corpus vocabulary), where each element $M_{i,j}$ represents the count of number times the word pair {wi, wj} occurs in the corpus, where the wi is the center word and wj is the context word. The probability of any word wj occurring in the context of wi can then be obtained as $P_{ij} = P(i \mid j) = \frac{X_{ij}}{\Sigma j\, X_{ij}}$.

For large corpus, constructing this matrix can take a very long time, but this needs to be done only once and the operation can be parallelized on a cluster or machines.

Cost Function

As mentioned above the starting point for the GloVe is the ratio of co-occurrence probabilities between words using various words and the vector representation of these words should encode the values of these ratios. To achieve this we can think of this ratio as a function F(wi, wj, w'k) where wi, wj are the word vectors when i, j are center words and w'k is the word vector when k is the context word. Technically, there is no distinction between word vectors for center words and context words in the GloVe model and they can be used interchangeably.

So, we need to find F, such that

$$F(wi,\ wj,\ w'k) = \frac{P_{ik}}{P_{jk}}$$

Further, we want the ratio to be encoded in the difference of the vectors wi and wj, so the F can take the form

$$F((wi - wj).w'k) = \frac{P_{ik}}{P_{jk}}$$

By making further assumptions about homomorphism and interchangeability, we get to the following form

$$\frac{F(wi.w'k)}{F(wj.w'k)} = \frac{P_{ik}}{P_{jk}}$$

So the objective for word vectors wi and wj then takes the following form.

$$F(wi\ .\ w'k) = \frac{X_{ij}}{X_i}$$

Assuming F = exp() and after some algebra we have

$$wi\ .\ w'k + log(Xi) = log(X_{ik})$$

We make the equation symmetric; we add biases for both wi and w'k. We also lump log(Xi) into the bias term for wi. We get the final form as:

$$wi\,.\,w'k\ +\ bi\ +\ b'k\ =\ log(X_{ik})$$

The above equation indicates that the dot product of the 2 vectors need to be equal to the log(co-occurrence count) for the pair. So, the final form of cost function is

$$J(\Theta)\ =\ \sum_{i,j} f(X_{ij})\,(wi\,.\,w'k\ +\ bi\ +\ b'k\ -\ log(X_{ij}))^{\ 2}$$

Θrepresents all the wi and w'i i.e the parameters that we need to learn from the co-occurrence matrix.

$f(X_{ij})$is the weighting function to control the overpowering of high frequently occurring word pairs. This optimization function results in interesting linear substructures in the vector space and other just similarity (i.e. similar words placed closer in vector space). For example the difference between different pairs of word vectors that are analogous to each other in similar contexts are the same, this helps the GloVe model to predict analogous words more accurately.

Highlights of The Glove Model

1. GloVe combines both count based matrix factorization to take advantage of global statistics and local context based predictions (e.g. skip-gram model).

2. The word vectors generated by GloVe help in identifying the similar words by using vector operations like euclidean distance or cosine distance between words.

3. Due to the way GloVe cost function is designed, the generated word vectors also exhibit linear substructures that capture interesting relationships/patterns between words based on the vector difference. For example, the vector difference between words {Canada, Ottawa} and {France, Paris} is similar, indicating that word pairs are related to each other in a similar way. This is helpful in discovering word analogies and named entity recognition and entity relationships.

Limitations

1. Training process may require a huge amount of memory for computing the global co-occurrence matrix especially if the training corpus is large, which is mostly the case in most real world applications.

2. Because the algorithm employs an unsupervised learning method the final outcome may be sensitive to word vector initialization.

Glove In Action

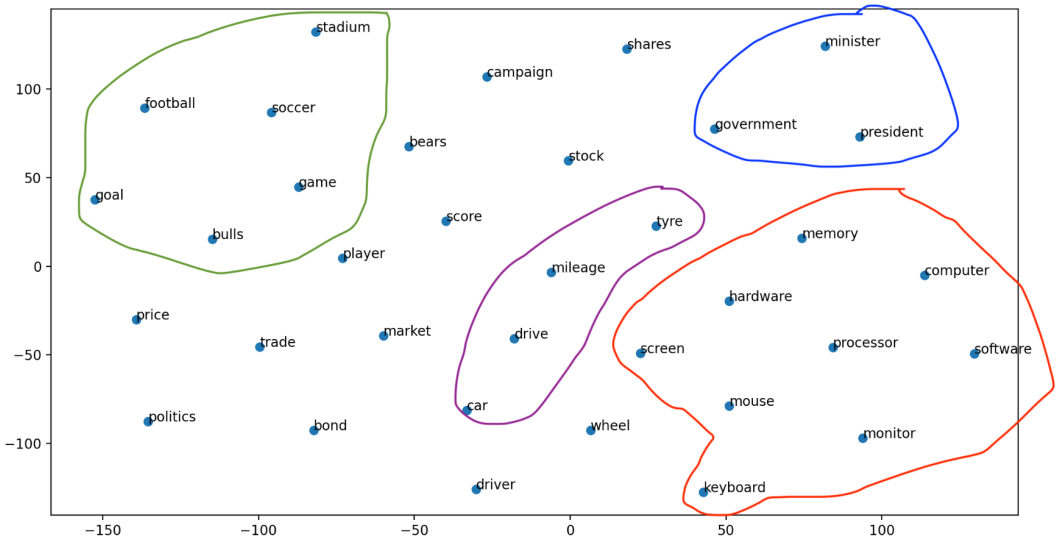
In this section we show GloVe embeddings in action and demonstrate the above mentioned properties of the model from a user/consumer point of view.

We downloaded the pre-trained word embedding from the glove project page (https://nlp.stanford.edu/projects/glove/). Specifically we downloaded the word embedding generated from Wikipedia 2014 + Gigaword 5 (http://nlp.stanford.edu/data/glove.6B.zip) which has 6 billion tokens.

Word Similarity

First, we experimented with the similarity of words. We created a collection of words from various topics like sports, politics, computing etc and retrieved the word embeddings for each of them from `glove.6B.50d.txt`, which contains 50 dimensional word vectors for each word.

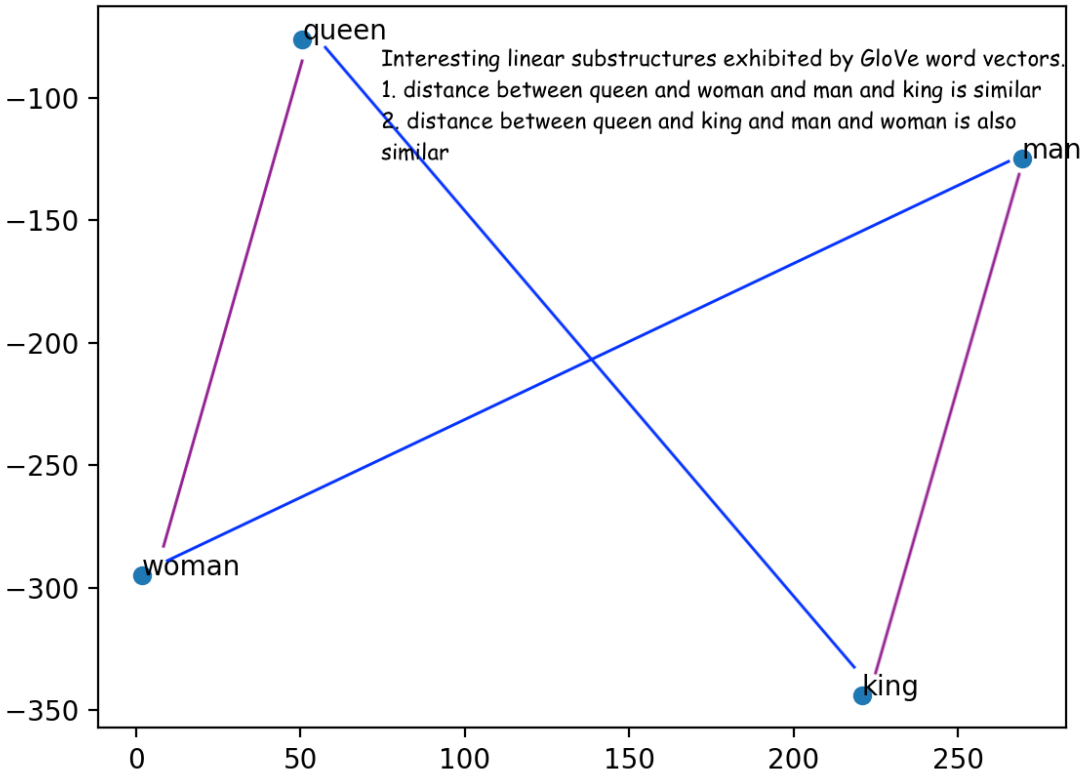
We then used the t-SNE library to plot these vectors in 2 dimensions space. The result clearly shows clustering of similar words in the plot as indicated in the following image from matplotlib scatter plot.



Linear Sub-structures

Next we demonstrate the linear sub-structures/patterns in the vector space representations generated by GloVe. We plot the vectors for {“man”, “woman”, “king”, “queen”} using the same procedure mentioned above. As seen in the image below we see distances between man and woman and king and queen are quite similar, in fact we see similar behaviour between {man,

king} and {woman, queen}



Conclusion

In this report we reviewed the GloVe model for generating vector space representations (or word embeddings) for a word in a text corpus. To start with, we provided an overview of what word embeddings are and their usage in NLP tasks, we then described 2 broad family of models i.e count based and local context models for generating these word embeddings. We then went on describe the GloVe model in detail and also highlighted some of the unique properties of the GloVe model. We also provided a sneak peak into usage of GloVe from a user's point of view.

References

1. GloVe project page (https://nlp.stanford.edu/projects/glove/)
2. GloVe paper (https://nlp.stanford.edu/pubs/glove.pdf)
3. Word2vec paper (https://arxiv.org/pdf/1310.4546.pdf)
4. https://github.com/stanfordnlp/GloVe
5. Lectures on NLP (https://www.youtube.com/playlist?list=PL3FW7Lu3i5Jsnh1rnUwq_TcylNr7EkRe6)