Intro to GloVe

AI Club Mini
Project Task 1

What is word embedding?

To perform any task with a model, we need to encode input as numbers.
In NLP, we definitely need word embedding.
It helps make vector representation of words capture their semantic relationships and contextual information. So vectors closely related in meaning or adjacent in common phrases are closer to each other.

For example, spam detection, topic categorisation, Named Entity Recognition (NER, where it identifies names of organisations and such in text).
Another example is word analogy. Like Queen:King::Woman:Man.
QnA systems - to recommend similar items, cluster certain related documents, etc. (Which we might use in RAG)

Different Types of Word Embeddings

Prediction-based : It generates **fixed** vector representation of a word that captures words closely related to it (Word2vec, GloVe)

Frequency-based : It gives importance to certain words in a text depending on how rare they are in the whole corpus of data and how many times they occur in the document.

Transformers : They generate dynamic vector representations of words that change depending on the context it is used in.

What is GloVe?

The GloVe algorithm is a powerful and efficient approach for learning word embeddings. GloVe doesn't just rely on neural networks to implicitly learn word relationships (like Word2Vec does); instead, it formally defines the mathematical structure (such as co-occurrence probabilities) that leads to meaningful semantic similarities in word embeddings.

The result is a global **log bilinear regression model** that combines the advantages of **global matrix factorization** and **local context window methods** (elaborated later). GloVe efficiently leverages statistical information by training **only on the nonzero elements in a word-word co-occurrence matrix**, rather than on the entire sparse matrix or on individual context windows in a large corpus.

Overview of the embedding process (general)

1. Pre-processing the text (tokenization, removing punctuation)
2. Sliding context window identifies target and context words, for the model to learn word relationships and build co-occurrence matrix.
3. Training to predict words based on context, keeping semantically similar words closer together. Cost function is error in prediction.

Quick Segue on Sliding Context Window

A **context window** is a technique used in natural language processing (NLP) to identify target and context words by sliding a window across a text. This method helps models learn relationships between words.

- **Identifying word pairs:** The GloVe algorithm starts by explicitly counting how many times a word *i* appears in the context of another word *j* within a text corpus. The context window determines what words are considered to be in the "context" of a given "target" word. The size of the window determines the words in the context.
- **GloVe Optimization**: The GloVe algorithm uses the co-occurrence counts obtained within a context window to optimize word vectors by minimizing the difference between the dot product of word vectors and the logarithm of the co-occurrence counts.
- **Sliding Window:** The context window slides across the entire text corpus, processing each word in turn to build the word-word co-occurrence matrix.

Rundown of how GloVe works

Preprocessing: (tokenization, removing stopwords, punctuation, etc.)
Vocabulary Creation: A vocabulary of unique words is created from the corpus, and their frequencies are counted.
Co-occurrence Matrix: A co-occurrence matrix is generated.
Training: It uses a weighted least squares objective to minimize the difference between predicted and actual co-occurrence probabilities.
Gradient descent is used to optimize word vectors and bias terms.
Word Vector Generation: Each word is associated with a dense vector.
This means that every word is mapped to a fixed-length vector where all dimensions contain real-valued numbers (as opposed to sparse vectors, which mostly contain zeros).

The Math behind glove

The core of GloVe is a weighted least squares regression model, defined as:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

Where:

- X_{ij} is the co-occurrence count of words i and j
- w_i and $\tilde{w}_j$ are word vectors
- b_i and $\tilde{b}_j$ are bias terms
- $f(X_{ij})$ is a weighting function
- V is the vocabulary size

Reasoning Behind the Model:

- **The model aims to capture the relationship between word vectors through their co-occurrence statistics.**
- **The dot product $w_i^T \tilde{w}_j$ should relate to the logarithm of the co-occurrence count $\log X_{ij}$.**

Derivation of the Model :-

a. Start with a general form: $F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$

b. Simplify to depend only on vector differences: $F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$

c. Take the dot product: $F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$

d. Enforce homomorphism: $F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$

e. Solve to get: $F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$

f. Take logarithm: $w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$

g. Add bias terms for symmetry: $w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$

Time Complexity

tion of word co-occurrences. In particular, we will assume that the number of co-occurrences of word i with word j, X_{ij}, can be modeled as a power-law function of the frequency rank of that word pair, r_{ij}:

$$X_{ij} = \frac{k}{(r_{ij})^{\alpha}} . \qquad (17)$$

Here, V=Vocabulary - set of unique words.
C=Corpus - group of all words used (even repeated)

"For the corpora studied in this article, we observe that Xij is well-modeled by Eqn. 17 (given above) with α = 1.25. In this case we have that |X| = O(|C| 0.8). Therefore we conclude that the complexity of the model is much better than the worst case O(V^2), and in fact it does somewhat better than the on-line window-based methods (unlike GloVe, which computes the global co-occurence offline pre-training) which scale like O(|C|). "

Sources

- Original paper: https://nlp.stanford.edu/pubs/glove.pdf
- https://becominghuman.ai/mathematical-introduction-to-glove-word-embedding-60f24154e54c
- https://www.geeksforgeeks.org/pre-trained-word-embedding-using-glove-in-nlp-models/
- https://www.youtube.com/watch?v=wgfSDrqYMJ4
- https://youtu.be/EHXqgQNu-lw?feature=shared