Fall 2021 CS 410 Technology Review

By *Brian Betancourt*

Introduction: Word Embeddings

In the field of neurolinguistic programming (NLP) as well as its many sub-disciplines, text must be understood by machines for various uses, such as comparing terms or strings of terms in order to determine their similarity. In order to process text for those uses, it is often convenient to transform text from characters (such as those found in the English alphabet) to numerical data which can convey the same meaning as those characters but can also be used in mathematical calculations.

The term "word embedding" describes the assumption that machines can transform text into vector representations which will carry their embedded meanings as denoted by their similar contexts. Recall that a vector refers to a series of values (scalars), arranged in a set. For example, an example of a vector y comprised of x_m terms can be expressed as follows:

$$y = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

"Global Vectors for Word Representation", or "GloVe" and "Word2Vec" are two popular methodologies for creating vector representations of text.

GloVe: Term Co-occurrence

GloVe was developed by Jeffrey Pennington, Richard Socher, and Christopher D. Manning at Stanford in 2014.[2] The term "global vectors" refers to that this model utilizes vectors produced from the context of the entire corpus, and not just a small "window".[1]

GloVe's methodology is to create a co-occurrence matrix of terms in a corpus which represents the count of each word, relative to another word.

GloVe assumes that the ratios of co-occurrences probabilities are more important than the probability of observing each word on its own. (i.e The ratio of the probability of seeing two related terms should be greater than the ratio of probability of seeing two unrelated terms). This concept is demonstrated in the GloVe paper:

GloVe establishes a matrix of word-word occurrences (X) such that X_{ij} denotes "j occurrences in the same context as i", where both i and j are words in the matrix. Similarly, $P(j|i)$ can be interpreted as the probability of observing term j, given the context of i.[1] This can be calculated in the form of:

$$P(j|i) = X_{ij}/X_i$$

Understanding the relationship between two terms relies on calculating their probability of co-occurrence relative to other terms not expected to be related. The formula above captures this essence in that it can be understood to mean that the probability of observing term j given the context of i is a ratio between their co-occurrence and the occurrence of all terms in the context of term i. While the above formula is derived and refined into further forms, I will leave its contribution to the complete GloVe loss function noted in a later section of this document.

GloVe: Sparse Matrices

A large corpus will produce a sparsely populated term frequency matrix. This is because for a large corpus, there will inherently exist a large number of terms which do not occur frequently (e.g. "Austria" and "Monsoon"). Conversely, terms commonly associated with each other will have very high co-occurrences and will result in a matrix sparsely populated, but containing very large numbers when populated. GloVe's solution to this is to reduce term co-occurrence counts by taking their logarithm.

GloVe: Lexicographical Distance

GloVe combines a term-term co-occurrence matrix with the assumption that consideration should be given to the lexicographical distance between terms, as in the case of local context windows. This is referred to as a "window size" or "context window" in other models. The motivation is fueled by the assumption that two terms which occur frequently in a document may be more closely related if the count of words between them tends to be small. GloVe quantifies this lexicographical distance by weighting each term co-occurrence so that terms closer in distance receive their full "score", while terms lexicographically distant from each other are "discounted" by some weight factor.[1]

It is in this way that GloVe utilizes both established broad approaches for word vector modeling: Global matrix factorization and local context windows. It is important to note that while the concept of lexicographical distance can be employed in this approach, its placement in the cost function is such that it may be interchanged with other weighting functions.

GloVe: Cost Function

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

The complete GloVe loss function can be interpreted as follows:

- V is Vocabulary size, or all the terms in the corpus
- $w_i^T \widetilde{w}_j$ denotes the dot product of vector w_i and $\widetilde{w}_j$, where $\widetilde{w}_j$ is the context or "target" term which relevant in calculating $P(word|w_j)$.
  This form comes after derivation of the ratio P_{ik}/P_{jk} and is described in greater detail in the study.
- b_i and $\tilde{b}_j$ are bias parameters corresponding center and context word vectors. These are included in the formula to balance the equation $w_i^T w_k + b_i + b_k = log(X_{ik})$
- The superscripted square is the embodiment of the "least squares" functionality commonly associated with regression models

- $f(X_{ij})$ is a general weighting function not explicitly defined by GloVe but may be modified in production. For example, a weighting function which emulates the concept of context windows may be used here to decrease the weights given to terms further in distance from the context word

Word2Vec

Word2Vec was created in 2013 at Google as an advancement upon previously established methods for word embedding and vector representation.[5] Rather than a distinct model itself, it defines a broad approach for word embedding which can have various types of execution.

For example, Word2Vec can employ either a continous "bag of words" or "skip-gram" architecture in most approaches. Both approaches also utilize neural networks (as opposed to a global term co-occurrence matrix) in order to process the input corpus and produce word vectors.[4] The mathematical properties of these neural networks is outside the intended scope of this paper, and instead a high-level overview of the "skip gram" representation as understood in the Word2Vec model will be presented.

Word2Vec: Skip Gram

In the Skip Gram model, we take a target word and use it to try to predict the context words which are close in lexicographical distance to the target word.

This is accomplished first by iterating through a corpus and selecting a middle term, then pairing that term separately with adjacent context words of a given "window" or context size. For example, a window size of "2" will identify the two terms adjacent to the left and right of the middle term.

The objective of Word2Vec with the "skip gram" approach is to maximize the probability of predicting context terms, given a target term.[6] This is accomplished by optimizing the following average log probability function over a sequence of terms denoted by w_i:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,j\neq 0}\log p(w_{t+j}|w_t)$$

The probability function above can be explicitly defined in a number of ways and is commonly defined using a "softmax" function, of which there exist several variations. The details of the softmax function are outside the scope of this document

While also outside the scope of this document, it is worth noting that Word2Vec, when constructed using the continuous bag of words (CBOW) approach utilizes a set of context words and attempts to predict a target, or "middle" word.[6]

GloVe and Word2Vec: Comparison

GloVe creates a vector space model based on all of the possible terms occurring in a corpus, hence the "Global" part of its names. This is in direct contrast to the Word2Vec model's use of word "contexts", which are narrow subsets of the larger corpus of varying length. However, the addition of a weighting function embodies the concept of "context" to some extent in that the lexicographical distances from middle terms may be weighted.

GloVe requires computing a term occurrence matrix for every term in a given corpus. For a large corpus, this may be computationally expensive. This should be taken into consideration when deciding between approaches. The GloVe study references GloVe's superior performance with regard to time, relative to Word2Vec specifically.[1] However, this may a function of the term occurrence matrix already having been computed prior to training the model.

Conclusion

The concept of word embeddings is almost a sub-discipline within NLP itself, with varying approaches and semi-recent advances (Word2Vec was discovered in 2013, for example). Both Word2Vec and GloVe offer methods which are promising for the purposes of interpreting text data and comparing their embedded similarities or dissimilarities. This can further be built upon to provide support for sentiment analysis and perhaps even higher levels of understanding, such as understanding intentionality or the motivation behind an actor's communication.

References

1. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.
2. https://github.com/stanfordnlp/GloVe
3. https://code.google.com/archive/p/word2vec/
4. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dea. 2013. Efficient Estimation of Word Representations in Vector Space. (https://arxiv.org/pdf/1301.3781.pdf)
5. T*omas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.*
6. https://www.tensorflow.org/tutorials/text/word2vec