How is GloVe different from word2vec?

Answer Request Follow 94 Comments 2 Downvote

5 Answers

Stephan Gouws, Creator of BilBOWA, the multilingual version of word2vec.
Answered Aug 30, 2015

Both models learn geometrical encodings (vectors) of words from their co-occurrence information (how frequently they appear together in large text corpora). They differ in that word2vec is a "predictive" model, whereas GloVe is a "count-based" model. See this paper for more on the distinctions between these two approaches: http://clic.cimec.unitn.it/marco... .

Predictive models learn their vectors in order to improve their predictive ability of Loss(target word | context words; Vectors), i.e. the loss of predicting the target words from the context words given the vector representations. In word2vec, this is cast as a feed-forward neural network and optimized as such using SGD, etc.

Count-based models learn their vectors by essentially doing dimensionality reduction on the co-occurrence counts matrix. They first construct a large matrix of (words x context) co-occurrence information, i.e. for each "word" (the rows), you count how frequently we see this word in some "context" (the columns) in a large corpus. The number of "contexts" is of course large, since it is essentially combinatorial in size. So then they factorize this matrix to yield a lower-dimensional (word x features) matrix, where each row now yields a vector representation for each word. In general, this is done by minimizing a "reconstruction loss" which tries to find the lower-dimensional representations which can explain most of the variance in the high-dimensional data. In the specific case of GloVe, the counts matrix is preprocessed by normalizing the counts and log-smoothing them. This turns out to be A Good Thing in terms of the quality of the learned representations.

However, as pointed out, when we control for all the training hyper-parameters, the embeddings generated using the two methods tend

to perform very similarly in downstream NLP tasks. The additional benefits of GloVe over word2vec is that it is easier to parallelize the implementation which means it's easier to train over more data, which, with these models, is always A Good Thing.

44.7k Views · 221 Upvotes · Answer requested by Nikhil Dandekar

Sujit Pal, search engineer interested in semantic search, text analytics and NLP, machine learning, etc.

Answered Aug 12, 2015

Thanks for the A2A. I believe the two do the same things, the main difference is how they are built. With word2vec you stream through n-grams of words, attempting to train a neural network to predict the n-th word given words [1,...,n-1] or the other way round. The end result is a matrix of word vectors or context vectors respectively. With Glove, you build a co-occurrence matrix for the entire corpus first, then factorize it to yield matrices for word vectors and context vectors.

23k Views · 31 Upvotes · Answer requested by Nikhil Dandekar

Ferhat Aydın, Computer/Software Engineer
Answered Apr 25, 2016

You can find good summaries for both of them;

GloVe: GloVe: Global Vectors for Word Representation

Word2vec: The amazing power of word vectors

The amazing power of word vectors blog post is go over all the papers below

- Efficient Estimation of Word Representations in Vector Space – Mikolov et al. 2013
- Distributed Representations of Words and Phrases and their Compositionality – Mikolov et al. 2013
- Linguistic Regularities in Continuous Space Word Representations – Mikolov et al. 2013
- word2vec Parameter Learning Explained – Rong 2014
- word2vec Explained: Deriving Mikolov et al's Negative Sampling Word-Embedding Method – Goldberg and Levy 2014

David Zornek, Lead Data Scientist (2017-present)
Answered Oct 24

The main insight of word2vec was that we can require semantic analogies to be preserved under basic arithmetic on the word vectors, e.g. king - man + woman = queen. (Really elegant and brilliant, if you ask me.) Mikolov, et al., achieved this through continuous bag of words and/or skipgram models (word2vec comes in both flavors) that are trained to be predictive of how native speakers would intuit

analogies in the real world. Under this approach, all that matters is the distance between two words, and the intrinsic statistical properties of the corpus (which were key to most earlier methods) end up getting lost.

Over at Stanford NLP, they liked the analogy preservation, but disliked losing transparency and direct consideration of the corpus' word occurrence statistics, which they regard as fundamental to the problem. They hypothesized that by identifying a way to achieve analogy preservation under linear arithmetic that uses only these fundamental statistical properties of the corpus as inputs, they would present an improvement in both accuracy and interpretability. The GloVe paper argues that they succeeded.

In practice, the main difference is that GloVe embeddings work better on some data sets, while word2vec embeddings work better on others. They both do very well at capturing the semantics of analogy, and that takes us, it turns out, a very long way toward lexical semantics in general.

1.6k Views · 2 Upvotes

Upvote 2 Downvote

Recommended All

Rahul RM, Text Analytics
Answered May 14

Thanks for the A2A. Already there are good answer by Stephan Gouws. I will add my point.

- In ***word2vec, Skipgram*** models tries to capture co-occurrence one window at a time
- In ***Glove*** it tries to capture the counts of overall statistics how often its appears.

4.8k Views · 1 Upvote