Classes and examples of embeddings

Goals

1. Exploring the new forms of embeddings
2. Using the doc2vec example to clarify the scope of application of new forms of embeddings

word2vec and beyond

Embeddings expanding word2vec

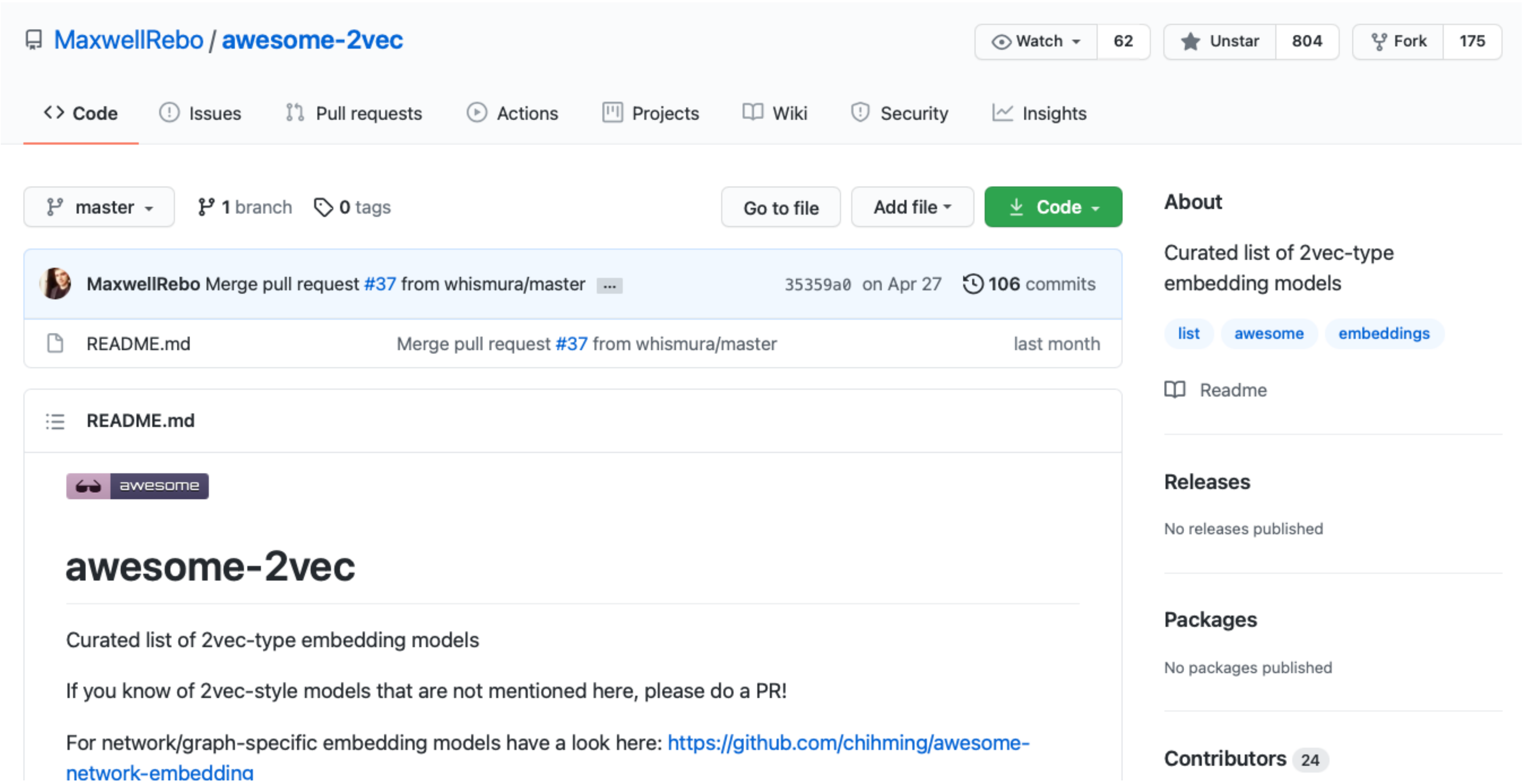
Embeddings designed for entities other than words

- doc2vec
- graph2vec
- tweet2vec
- emoji2vec
- …

Context-aware embeddings

- Distinctive feature: the vector associated with a word can change depending on the context in which the word appears in [e.g., vector('running' a business) vector('running' a marathon)]
- Examples:
  - ELMo
  - BERT

A comprehensive review of *2vec algorithms



Embedding words, phrases, and documents

- doc2vec is one of the most significant extensions of word2vec.
- Mainly, doc2vec aims to represent whole documents as vectors.
- The authors present the algorithm as an alternative to a traditional BoW.

Distributed Representations of Sentences and Documents

Quoc Le QVL@GOOGLE.COM
Tomas Mikolov TMIKOLOV@GOOGLE.COM
Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

Abstract

Many machine learning algorithms require the input to be represented as a fixed-length feature vector. When it comes to texts, one of the most common fixed-length features is bag-of-words. Despite their popularity, bag-of-words features have two major weaknesses: they lose the ordering of the words and they also ignore semantics of the words. For example, "powerful," "strong" and "Paris" are equally distant. In this paper, we propose *Paragraph Vector*, an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents. Our algorithm represents each document by a dense vector which is trained to predict words in the document. Its construction gives our algorithm the potential to overcome the weaknesses of bag-of-words models. Empirical results show that Paragraph Vectors outperform bag-of-words models as well as other techniques for text representations. Finally, we achieve new state-of-the-art results on several text classification and sentiment analysis tasks.

tages. The word order is lost, and thus different sentences can have exactly the same representation, as long as the same words are used. Even though bag-of-n-grams considers the word order in short context, it suffers from data sparsity and high dimensionality. Bag-of-words and bag-of-n-grams have very little sense about the semantics of the words or more formally the distances between the words. This means that words "powerful," "strong" and "Paris" are equally distant despite the fact that semantically, "powerful" should be closer to "strong" than "Paris."

In this paper, we propose *Paragraph Vector*, an unsupervised framework that learns continuous distributed vector representations for pieces of texts. The texts can be of variable-length, ranging from sentences to documents. The name Paragraph Vector is to emphasize the fact that the method can be applied to variable-length pieces of texts, anything from a phrase or sentence to a large document.

In our model, the vector representation is trained to be useful for predicting words in a paragraph. More precisely, we concatenate the paragraph vector with several word vectors from a paragraph and predict the following word in the given context. Both word vectors and paragraph vectors are trained by the stochastic gradient descent and backpropagation (Rumelhart et al., 1986). While paragraph vectors are

A doc2vec application

Predicting the Grammys with data

Published on February 15, 2016

Varun Jewalikar | Follow

496 42 0

Since 1959, the National Academy of Recording Arts and Sciences has awarded a Grammy for Song of the Year, choosing from 5 or more nominees each year.

Wrap-up

Main point

- New forms of embeddings have developed along two arms:
    - Creating vectors for diverse entities (e.g., graphs)
    - Creating more nuanced vector representations (e.g., ELMo)
- Ad hoc embeddings — such as doc2vec — offer distinctive features for statistical or ML post-processing