Applications of word2vec

1 Introduction

Earlier many NLP systems treat every word as an atomic unit and each word is represented as an unique word without the notion of any kind of relation or similarity between the words. The most useful concept in NLP techniques nowadays is to use the distribution of words. This representation of words is in the form of continuous vector known as embedding vector with a modest dimensionality. It is generated using the context of words in which it is used, thereby we get almost similar embedding vectors for the similar words.

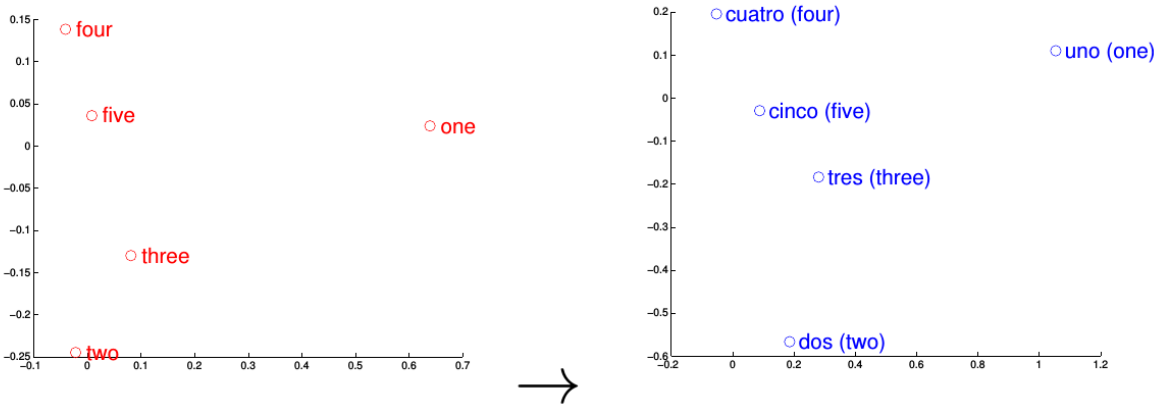
This paper (Mikolov, Chen, Corrado, & Dean, 2013) proposed two novel techniques for learning of the embedding vectors namely Continuous Bag-of-Words(CBOW) and Skip-gram Model. These models are basically the simpler version of the earlier Neural Net Language Model(NNLM) with the non-linear hidden layer is removed. The major advantage of the proposed techniques is that it is very efficient in terms of computational complexity. So it can be trained on very huge data sets with billion of words thereby generating high-quality word vectors.

The CBOW technique is computationally faster than skip-gram so it is used when huge datasets are present while skip-gram is better for small datasets with higher accuracy. The extended version of skip-gram technique is presented in the paper (Mikolov & Dean, 2013) to make the algorithm much more efficient. We will discuss some very useful applications of word2vec technique in many fields.

2 Applications

2.1 Machine Translation

The paper (Mikolov, Le, & Sutskever, 2013) develops a method to automate the process of generating dictionaries and phrase tables for translation between different languages using the word2vec technique. It is also useful in translating the missing words using the semantic and syntactic knowledge of the word given by embedding vector. It can be used with the languages that are substantially different because it mainly deals with the meaning of the word.



It can be visualized from the figure that English words from one to five have similar geometric arrangements with corresponding spanish words uno to cinco.

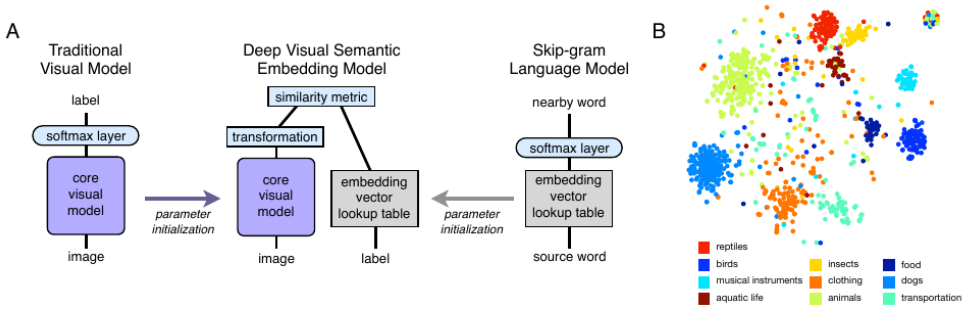
The proposed method consists of two steps. First, the two monolingual models(representation of words as a vector) are build for the source and target languages. Secondly, it uses a small bilingual dictionary to learn a linear projection between the languages. Since all common languages share concepts that are grounded in the real world so there will be almost similar geometric arrangements in the vector space for similar words is the reason behind working of the above model. So the relationship between these vectors can easily be captured by linear mapping(rotation and scaling). It can be visualized by the figure above.

For the first step, skip-gram and CBOW models are used to represent the words in the form of embedding vectors. To find the translation matrix between the vectors representation of the languages we use the given small bilingual dictionary as the training set. This translation matrix can be found using Stochastic Gradient Descent method by minimizing the square error loss. At the prediction time the model map the vector representation of source language word to the vector representation of target language using translation matrix. And then we find the word whose representation is closest to output vector in target language space using cosine similarity as the distance metric.

2.2 Computer Vision

The paper (Frome et al., 2013) presents the model trained to identify visual objects using the both labelled image data as well as semantic information got from unannotated text. This model matches state-of-art performance on the 1000-class ImageNet dataset and the semantic knowledge improves

zero-shot predictions (achieving correct prediction for those images that are not available in training data). The main objective behind the proposed approach is to transfer the knowledge learned from the text domain of labelled images to a model trained for visual object recognition. The following image describes the architecture of DeViSE model.



DeViSe model is initialized from two pre-trained neural network models. First, the embedding vectors learned by the word2vec language model are normed and used to map label terms into target vector representations. These embedding vectors are used to train the model for visual recognition. Next, it uses a visual model architecture but instead of the softmax prediction layer, there is projection layer and a similarity metric. Projection layer transform the output of the top layer of core visual network for the given image into the vector similar to embedding vectors.

The objective is to train the matrix of trainable parameters in the linear transformation layer which uses the training set. At the test time, for a new image one first computes the vector using visual model and transformation layer, then one searches for the nearest labels in the embedding space.

2.3 Deep Walk

Social representation are the hidden features of the vertices that capture neighborhood similarities and community membership. Deep Walk (Perozzi, Al-Rfou, & Skiena, 2014) is an algorithm that learns social representation of a graph vertices using a stream of random walk. Each vertex has its social representation encoded in the form of continuous vector. The most important advantage of DeepWalk is that one can integrate the graphical features with any simple machine learning algorithm.

Random Walks in network is the connection to local structures and is taken analogous to the short sentences in language modelling. It is a stochastic process and the the next vertex from any vertex is choosen at random from the neighboring vertices. Each vertex has its random walk.

It treats each vertex like a word and each random walk as a sentence

and then uses the skip-gram technique to learn the social representation of every vertex. In this way we can find the similarities between different vertices. This technique is based upon skip-gram model and on random-walk. Skip gram is implemented with the hierarchichal softmax to reduce the time complexity of calculating the probability function from $O(|V|)$ to $O(log|V|)$. In hierarchical softmax the vertices are assigned to the leaves of binary tree which converts the prediction problem into maximizing the probability of aspecific path in hierarchy. The probability function is maximized by Stochastic Gradient Descent.

2.4 Dependency Parsing

In (Bansal, Gimpel, & Livescu, 2014) the embedded vectors are used as the parsing features to generate dependency parsing for the given sentence. The gains are seen after performing a hierarchical clustering of embedded word vectors and then using features based on the hierarchy.

The main improvement is seen after the tailoring the model with two modifications.These are a smaller context window and conditioning on syntactic context(dependency link and label). The reason behind small windows size is that with small w, grouped words tends to share same POS tag. For the syntactic context the skip-gram model is trained on dependency context rather than linear context. As it expects that embedding will be helpful when words that have similar parents and children are close in the embedding space. The accuracy is similar to the Brown Clustering method but in a fraction of training time.

There are two kinds of indicator features to find dependency parsing features from continuous representation features. In Bucket features, for both parent and child vectors in a potential dependency it fire one indicator feature per dimension of each embedding vector, where the feature consists of dimensional index d and a bucketed version of embedding value in that dimension. In Cluster bit string features, To take into account all dimensions simultaneously, the algorithm perform agglomerative hierarchical clustering of the embedding vectors. Out of thesese two bit string features lead to more improvement in accuracy as compared to bucket features.

2.5 Sentiment Analysis

The work (dos Santos & Gatti, 2014) proposes a deep convolutional neural network that exploits from character-to-sentence level information to perform sentiment analysis on small texts. There are two hidden layers involved in the neural network to extract information from words and sentences. The

first layer is used to transform words into embedding vectors using morphological, syntactic and semantic information about the words. World-level embeddings capture the semantic and syntactic information and Character-level embeddings capture morphological and shape information. The second hidden layer is used to extract a sentence level representation and computes a score for each sentiment label.

To perform word-level embeddings the model uses the word2vec technique. Word-level embeddings play a very important role in the CharSCNN architecture. Word Initializing word-embeddings using unsupervised pre-training gives an absolute accuracy increase of around 1.5 when compared to randomly initializing the vectors.

3 Application on Data other than language

The DeepWalk algorithm (Perozzi et al., 2014) is the very good application of word2vec technique in data mining of graphical networks.

The skip-gram algorithm found very useful in learning the continuous distribution of biological sequences in the paper (Asgari & Mofrad, 2015). This paper proposes an unsupervised data-driven distributed representation for biological sequences. The skip-gram technique proves to be very successful in recommendation systems as well in the publication (Barkan & Koenigstein, 2016).

References

Asgari, E., & Mofrad, M. R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, *10*(11), e0141287.

Bansal, M., Gimpel, K., & Livescu, K. (2014). Tailoring continuous word representations for dependency parsing. In *Acl (2)* (pp. 809–815).

Barkan, O., & Koenigstein, N. (2016). Item2vec: Neural item embedding for collaborative filtering. *arXiv preprint arXiv:1603.04259*.

dos Santos, C. N., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Coling* (pp. 69–78).

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems* (pp. 2121–2129).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining* (pp. 701–710).