Improving Word Vector Representations using Semi-Supervised Learning

Tom Hamer[1]

Australian National University, ACT, Australia,
tomhamer@live.com,
Home Page: http://www.anu.edu.au/

Abstract. Vector space word representations have found use in a great variety of cases, increasing in popularity following Tomas Mikolov's 2013 paper, which provided an effective method for training them. Mikolov's unsupervised learning approach does not however take advantage of resources that already exist detailing known relationships between words. Our model uses the WordNet 2006 ontology (Miller) to bias the training of a neural network model similar to Mikolov's CBOW so that the trained word representations contain hyponym relationships. Using Wikipedia text to perform the unsupervised portion of the training, we show that the WordNet relationships can be successfully represented as continuous space word vector representations, and the model successfully learns word vectors from the text. While some results are shown, it was not possible with the time constraints to present a rigorous performance test of word vector quality on a benchmark data set.

Key words: Word vector, Ontology, Embeddings

1 Introduction

The use of AI in learning word vector representations has been a hot topic of research for many years. The foundation behind this research is the distributional hypothesis theory, which puts forward that words used in similar contexts usually have similar meanings (Harris, 1954). This suggests that it is possible to learn context vectors from unsupervised text mining, using the words that appear close to each word to establish word context.

Gedeon's 1997 paper showed that it was possible for neural networks to learn atomic context vectors based on co-occurrences of other words in the same document. The method was effective in finding words which were of similar context, however failed in finding synonym relationships. Further, there was no evidence presented in the paper of the representations being able to model specific relationships between words.

In 2013 Tomas Mikolov brought word vectors into frequent use through creating an efficient strategy to learn complex relationships between words. Mikolov's paper presents two models - the CBOW model and the Skip-Gram method. While the Skip-Gram method has been shown to produce excellent results on very large data sets (Mikolov, 2013), in this paper we will focus on the CBOW model as it is easier to train and more effective on smaller datasets. In Mikolov's CBOW model, each word is assigned a random unique one-hot encoded id (Harris). To train, the model takes a corpus of text documents and each word from the text along with its four closest neighbours becomes a training pattern. The network is trained to predict the word's id based on its neighbours using a 300 hidden unit neural network. The hidden layer activations of the trained network for each id are then used as vector representations for each word. The model is purely unsupervised and does not use any labelled training data.

The lack of supervised learning in Mikolov's model means that the model does not take advantage of existing linguistic resources detailing word relationships. A common source of word relationships are ontologies - models that contain a large range of relations between concepts, data, and entities (Harper, 2000). Since Mikolov's paper, attempts have been made to integrate ontologies into word vector learning. Faruqui's 2015 paper retrofits word embeddings according to their closeness in their ontological similarity relationships in the wordnet 2006 ontology. It has also been shown that antonym and synonym relationships can be encoded into vector space representations of words using ontological resources (Mrki et al, 2016). However, neither of these strategies take advantage of directional relationships between words, nor do they accurately represent hypernym/hyponym relationships.

1.1 Dataset

In this paper, two datasets were utilized - the 2006 Wordnet Ontology and a corpus of 100,000 randomly selected wikipedia articles.

Wordnet Ontology The wordnet ontology was first proposed by George Miller in 1995 as a lexical database of English nouns, verbs, adjectives and adverbs. In the 2006 version, the model contains over 100,000 hypernym relationships. Larger ontologies exist, however for this project it was decided that the ontology needed to be kept at a manageable size. The ontology used for this paper was downloaded from the WordNet website in an rdf format. The dataset was then loaded into OntoDB, and exported in a JSON format.

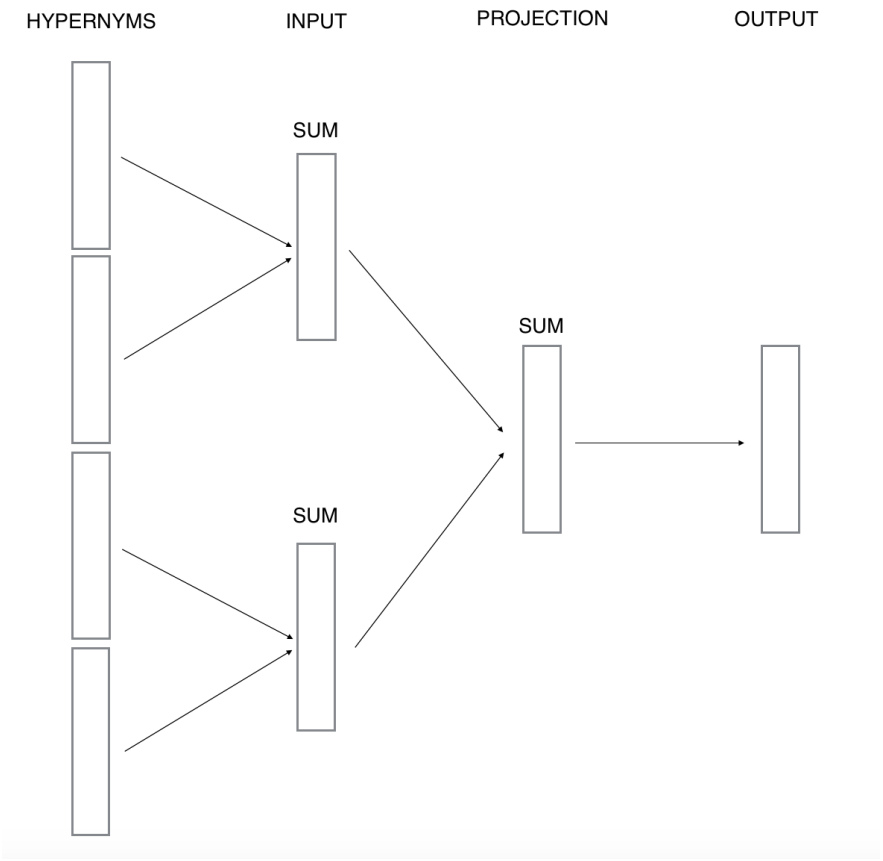
Wikipedia The Wikipedia articles were taken from the 17th file of the Wikipedia XML dump in Feburary 2018. This data can be found at: https://dumps.wikimedia.org/enwiki/20180220/. The data was downloaded in XML format and processed into paragraphs of plaintext. In total, around 124,000 articles were processed.

2 Model

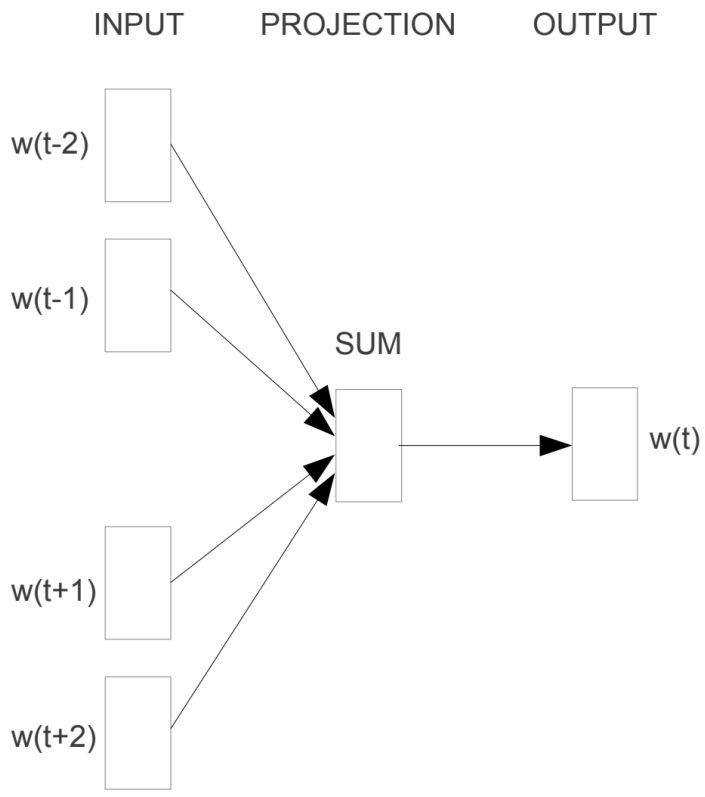
2.1 Neural Network

Similarly to in Mikolov's CBOW architecture, the aim of our implementation was to train a neural network which would take text from documents, and use the words neighbouring each word to predict the word. An example

FIGURE 1



Our Method - words are represented as the sum of their hyponyms



Mikolov's Method - randomly assign word ids

would be the sentence fragment "dog chases cat" - here "dog" and "cat" would be inputted to the network, and the network would be trained to predict "chase". In order to train on these examples, each word must be given an id. In Mikolov's model, these ids are assigned randomly, however we introduce a novel method for encoding the word ids in a meaningful way before training occurs. In our model, each word id is represented as the sum of the ids of its hyponyms. These hyponyms are extracted from the relationships detailed in the WordNet 2006 Ontology.

2.1.1 Encoding Ids

At the beginning of processing, each word existing in the Ontology was given a unique one-hot encoded id (Harris). Hyponym relationships were mined from the ontology, and any word which had hyponyms was represented as the sum of the ids of its hyponyms. The words which had no hyponyms kept their initially assigned id. This enforced that in the training patterns for the neural network model each word was represented as a sum of its hyponyms as found in the ontology. For example, 'sex' is represented as the sum of 'femaleness', 'maleness' and 'androgeny'.

2.1.2 Creating Training Patterns

The Wikipedia text is used to generate training samples as follows. A window is drawn sequentially through the text which takes each word and its two neighbours. Similarly to the CBOW architecture, the ids of the two words either side of each word are used to predict the middle word id. The ids used here are the ones derived in the previous step from word's hyponyms. The input of the network was the sum of the two word ids either side of the word, and the output was the output word. Using the WordNet 2006 ontology, we derived vectors of length 18,949, meaning that there were only 18,949 words with no hyponymss - the remainder of the words could be represented as a combination of these words. To decrease the amount of training data, just 10,823 training examples were chosen from the patterns generated using Wikipedia.

2.2 Neural Network Architecture and Parameters

Using the Wikipedia patterns encoded using the ontology-derived ids, a one hidden layer neural network was trained. The neural network had an input layer of size 18949 and an output layer of 18949. (see figure 1).

Hidden Layers Similarly to Milokov's paper, a single hidden layer of size 300 was used. Reducing this number of hidden units resulted in under-fitting, meaning that there were significant performance decreases in the word vectors. Increasing the number of hidden layer units resulted in increases in computational time, which needed to be minimized.

Mini-Batches To shorten the training time and decrease the required amount of memory, we use batches of size 10. The size of these batches was minimized as much as possible - however, it was important to use batches due to the immense amount of computation required to train the vectors.

Gradient Decent Optimizer The 'ADAM' adaptive stochastic gradient decent optimizer was utilized, to make gradient decent occur within fewer epochs (Kingma & Ba, 2015). Since there was a large amount of data to be processed, it was important optimize convergence of the model, even if this lead to higher likelihood of getting caught in local minima.

Activation Function The model uses the rectified linear unit. This helps to prevent under fitting due to the larger gradient update magnitudes.

Epochs The algorithm trained in very few epochs. This is likely due to the fact that the network is very large, with relatively few training examples for each neuron. Because the loss did not improve significantly after 4 epochs, training was discontinued.

2.3 Tooling

Scikit learn was used to train the word vectors. This is because its implementation hides some complexity that other libraries expose - since the neural network itself was relatively simple this was not required.

2.4 Extracting Word Vectors

The word vector representations were extracted through calculating the hidden layer activations when each word id was inputted into the network. Since the hidden layer was of length 300, the word vectors that were produced were of length 300.

3 Results

The model is able to demonstrate learning of relationships derived both from text and from ontological sources. Unfortunately, showing testing results on a benchmark dataset was not possible due to time constraints.

3.1 Model Successfully Represents Relationships Derived from Text

The model was tested and verified to show understanding of some connections between words that were derived from the Wikipedia articles. Nearest neighbours with the euclidean norm was used as a metric to measure word similarity. We found that for example, 'combat'+'peacetime' was very close to 'unfriendliness', yet 'combat'+'peacetime' was very close to 'combat'+'destroy' was very close to 'shoot' or 'murder'. This suggests that the word representations are capable of understanding complex relationships. Table one contains some examples of very high sentiment, with words such as 'combat' and 'Nazi', as with less training data their word relationships are often more apparent.

Word	Nearest Neighbours
music+product	music, write, **record**
western+Germany	**West_Germany**
combat+peacetime	combat, **unfriendliness**
combat+destroy	reach, bend, **shoot**, **murder**
Nazi+die	sequence, **victory**

Table 1. Examples of successful word relationship representation in the trained model.

3.2 Model Successfully Represents Ontological Relationships

The model was verified to successfully register some of the ontological hyponym relationships encoded into the ids of the words at the start of training. The intended behaviour, that hyponyms sum to their hypernyms is mostly apparent in the word vectors that were extracted (table 2). An important benefit of the ontology use that can be seen here is that knowledge of highly specialized domains has been encoded into the word representations, such as understanding categorization of different forms of fever.

Word Sum	Nearest Neighbour	Ontology Relationship
African+composer+statesman+doctor +athlete+worker	**person**	All summed words are hyponymss of 'person' in the ontology.
news_program+telecast+talk_show+sitcom+episode +television_program+serial	**broadcast**	All summed words are hyponyms of 'broadcast' in the ontology.
elegy+verse+blank_verse+epic_poem+lyric +free_verse+haiku+sonnet+ballad	**poem**	All summed words are hyponyms of 'poem' in the ontology.
Argentine_hemorrhagic_fever +Ebola_hemorrhagic_fever +Marburg_disease	**hemorrhagic_fever**	Ebola_hemorrhagic_fever, Marburg_disease and Argentine_hemorrhagic_fever are hyponyms of hemorrhagic_fever

Table 2. Examples of successful word ontology representation in the trained model.

When compared to results derived from the Gensim (Radim & Sojka) implementation of Mikolov's Skip-Gram model (table 3), we see the benefits of the model our paper proposes. In the word sums on table 3, some words are omitted due to the fact that they do not appear in the vocabulary of the Gensim implementation. Howevever, the difference is still clear - rather than the summed words adding to equal their hyponym, in some cases there are instead somewhat random words from the same context domain.

Word Sum	Nearest Neighbours
African+composer+statesman+doctor +athlete+worker	musician, politician, jazz_musician poet_Aime_Cesaire
news_program+telecast+sitcom+episode+serial	episodes, Watching_Ellie, ep, sitcoms
elegy+verse+lyric	lyrics, **poem**
ebola+Argentine+Marburg	Ebola,Marburg_fever, Marburg_virus, **hemorrhagic_fever**

Table 3. Examples of hyponyms representation in Mikolov's generic model.

4 Discussion

4.1 Improvements to Word Vector Averaging

One of the most common uses for continuous space word representations is in word vector averaging, where a sentence or sentence fragment is represented as a sum of its word vectors. Using the representation presented in this paper, where words adding to each other are represented as a hyponym relationship, is beneficial in this context as it creates the structure where more specific words from the same catagory sum to a more general word. For example, where there is a search engine with a user who searches for "where can I find a bed and table", the bed and table would average to their common hypernym "furniture", and results matching all furniture could be retrieved.

4.2 Limitations

Window Size In Mikolov's model, the context vectors included four neighbouring words, however, to reduce training time, in our method we only include two. This may have limited the amount inter-relationships the network learnt.

Hyponym Magnitude An additional problem is that because of the way that hyponym ids are constructed using the sum of the hypernyms, it leads to certain words having a larger magnitude than others. This is a potential limitation, as it can create a large distance between specific and general words, even if they are strongly related. However, this problem may be rectified in the future through normalizing the vectors.

5 Conclusion

The results so far are promising. Despite the fact that the algorithm has been trained on relatively few patterns, it is already beginning to reveal complex relationships. The ontology is providing a much needed structure to the learning to make it easier to learn word concepts. With more text and a larger ontology, the model could potentially be able to achieve better performance than current solutions.

5.1 Future Work

Test on a Benchmark Dataset Once the model has been trained on more data, it should be tested against other models to quantify possible improvements of the semi-supervised approach. Such a data set is Microsoft's Semantic-Syntactic Word Relationship test(Mikolov, Yih & Zweig, 2014).

Encode Additional Types of Ontological Relationships Antonym, synonym and many other word relationships can be found commonly in ontologies. Researching the way that these relationships can be encoded in vector space representation, and then including these relationships in the biasing of the ids would improve the word vectors.

Implement Skip-Gram Model In his paper, Milikov also presents an additional model for training the network, known as the Skip-Gram model. This model uses the word id to predict the surrounding words. This model is more difficult to train, but can provide better results than the CBOW architecture in a large enough data set. Implementing the biasing presented in this paper for this model could improve on the state of the art in word embeddings.

6 Bibliography

Harper, D (2000). Online Etymology Dictionary. Avaliable online: [https://www.etymonline.com/word/ontology]

Miller, A. G. (2006). WordNet: A Lexical Database for English. *Communications of the ACM* Vol. 38, No. 11: 39-41.

Harris, Z. (1954). "Distributional structure". Word. 10 (23): 146162.

Gedeon, T. D., Bustos, R. A., Briedis, B. J., Greenleaf, G., & Mowbray, A. (1997). Word-concept clusters in a legal document collection. In International Conference on Computational Intelligence (pp. 329-336). Springer, Berlin, Heidelberg.

Harris, David and Harris, Sarah. (1953) Digital design and computer architecture (2nd ed.). San Francisco, Calif.: Morgan Kaufmann. p. 129. ISBN 978-0-12-394424-5.

Faruqui M, Dodge J, Jauhar S, Dyer C, Hovy E, Noah A. (2015). Retrofitting Word Vectors to Semantic Lexicons. *Computation and Language*. Avaliable online: [https://arxiv.org/abs/1411.4166]

Kingma B, Ba J. (2015) Adam: A Method for Stochastic Optimisation. *ICLR*. Available online: [https://arxiv.org/pdf/1412.6980.p

Mikolov T, Yih W, Zweig G. (2013), Linguistic Regularities in Continuous Space Word Representations. Available online: [https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/rvecs.pdf]

Radim R, Sojka P. (2010) Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. pp 45–50. Avaliable online: [http://is.muni.cz/publication/884893/]