Comprehending Text Meaning through Similarity

Adeel Ahmed [†] , **Imran Amin**[††] and **Muhammad Mubashir Khan**[†††]

Department of Computer Science,
Shaheed Zulfikar Ali Bhutto
Institute of Science and Technology,
Karachi, Pakistan

Department of Computer Science,
Shaheed Zulfikar Ali Bhutto
Institute of Science and Technology,
Karachi, Pakistan

Department of Computer Science & IT
NED University of Engineering &
Technology
Karachi, Pakistan

Summary
This paper addresses how natural language processing (NLP) works with deep learning models to understand meaning of words in text. In this work, vector space models representing words into continuous vector representations are employed for identification of semantic and syntactic similarity between words in text articles. The model is trained and evaluated on unlabeled news articles [29], [30]. The model is implemented with continuous bag-of-words (CBOW) and skip-gram (SG) architectures with negative sampling (NES) and hierarchical softmax (HS) techniques. The model is evaluated on word similarity task, analogy tasks and vector compositionality to identify linear structure of word vectors representations. Computationally, the cost of training time and required memory for two architectures trained with two techniques is compared. It is observed that architectures trained with HS are expensive to train and more memory intensive than NES. Moreover, the findings of the evaluations on different task is presented representing both semantic and syntactic regularities in word embeddings.

Keywords:
word similarity; deep learning; unstructured text; natural language processing;

1. Introduction

The research in NLP has undergone an evolution in computational techniques over a period of several decades. The NLP tasks involving computational cost has reduced manifolds from several minutes in processing of a sentence analysis task to millions of pages being processed in fraction of a second [1]. NLP enables machines to understand natural language text to perform a number of tasks such as text similarity, question answering, question generation, facts extraction, concept extraction, part-of-speech-tagging, parsing, text summarization, language translation, and named-entity-recognition.

Deep learning has been used across vision, pattern recognition and speech applications and produced some state-of-art results. The major decrease in error rates achieved with deep learning models has therefore led researchers to take renewed interest in the execution of NLP tasks [2]. For example, analyzing the text authored by a person can help infer various features such as gender and age of that person.

In contrast, machine learning approaches in NLP have traditionally been reliant more on narrow models such as logistic regression and support vector machines. These approaches are more dependent on human labelled features, which in turn makes the job difficult as labelled data demands both time and is often not complete. Moreover, the approaches use high dimensional and sparse features during training.

Deep learning methods help in learning representation of multilevel features and produce very efficient results. Therefore, the scope of application of deep learning to process different NLP tasks is quite extensive. In recent years, NLP tasks based on neural networks built on dense vector representations produce better qualitative results. Notably, word embeddings [3], [4] and deep learning methods [5] are the underlying basis for such accomplishment.

Word embeddings represented by distributional vectors define each word in text by a vector. The vector essentially embeds the characteristics of surrounding neighboring context words. Since, words with same meanings generally tend to appear with similar context words, resultantly, their corresponding distributional vectors represent some sort of general notion of similarity between words [3], [4].

In statistical NLP, a simple language modeling performed under N-gram model, trained on billions to trillions of words [6] may outperform many complex NLP systems trained on comparatively lesser data with the downside that former models index words only as atomic tokens in vocabulary and do not define the notion of similarity. However, on the other hand with the recent advances resulting in better performance of distributed representation or word embeddings [7], language models based on neural network models trained on large data sets outperform simple N-gram models [8], [9], [10]. Moreover, statistical NLP dealing with complex NLP tasks experience

Manuscript received December 5, 2024
Manuscript revised December 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.12.5

curse of dimensionality problem, which is easily overcome for words existing in low-dimensional space through distributed representation learning.

To address similarity of words in text, the representative vectors of similar words should appear in close proximity to each other in the vector space. Such similarity in words is classified as syntactic and semantic similarity, 'Like' to 'likely' is an example of grammatical syntactic similarity, whereas 'man' to 'woman' and 'king' to 'queen' is an example of an analogy related to semantic similarity. Moreover, the vectors may exhibit a number of multiple linguistic degrees of similarity [11]. This notion of similarity can also be applied on title of the text representing the various linguistic features which can be compared through vector representations to find similarity across content in text and identify relevant important linguistic rich concepts which could be used as candidate answers in question answering [33] and question generation [34] research study.

In this paper, we measure the similarity of words and their compositionality of words in text articles [29], [30] through two model architectures CBOW and SG generating continuing representation of quality word vectors representing both syntactic and semantic relationships between words. Both model architectures presented are trained with HS [12] and with NES to address impact of frequent words on word vectors under word2vec model [4], both proposed by Mikolov et. al. The paper also compares how training time of both architectures being trained with two techniques fare with one another. The evaluation of the learned representations is undertaken to identify semantic similarity of words and analogies tasks, the syntactic similarity of analogy task and vector compositionality by means of additive vector algebraic operations.

Later sections of the paper are structured as follows: Section II introduce the background work on deep learning word2vec model along with CBOW and SG architectures including background detail on HS and NES. Section III discuss the overview of the system set up for experimental including problem definition, news article corpus collection and data set preprocessing and preparation. Section IV discuss the results measuring the similarity of words and analogy tasks and their corresponding accuracy along with comparison on how training time varies across different architectures. Conclusively, Section V discuss work being concluded and the extension of current work considered for future work.

2. Related Work

2.1 Distributed Representations

Words represented as vectors in form of distributed representations are key to learning algorithms and they perform better at establishing similarity between words. Keeping in view the guiding hypotheses that meaning may be derived from the distributional information or more simply by the context of the word it appears in is fairly a historical perspective shared by many linguists and authors. The author John Rupert Firth in his linguistic theory describing the importance of context to meaning of a word suggested, "you shall know a word by the company it keeps" [27]**.** The notion of use of distributional representation was introduced back in 1986 by Rumelhert et. al. [13]**.** This was later used in statistical language modeling [8] and a number subsequent tasks in NLP [11], [14], [15], [16], [17], [18], [31], [32].

2.2 Word2Vec Model

Word2vec learns word embeddings as continuous vector representation of words [11]. The model popularized the use of word embeddings by enhancing the accuracy of representing multiple degrees of similarity. Such similarity was outlined for both syntactic and semantic tasks in word2vec model [4]. The proposed model made possible to learn vectors beyond 50 to 100 dimensions commonly implemented across other models, providing decreasing complexity of the model, allowing training on 'a few hundreds of millions of words' [12].

The proposed model by Mikolov et. al. [12] uses simple neural network model to learn distributed representation keeping linear regularities intact and minimizing computational complexity. The model performs better compared to Latent Semantic Analysis (LSA) [19], [11].

Two log-linear architectures proposed under this model are trained by first generating continuous word vectors through simple model and later training neural network language model (NNLM) with the generated word vectors. The resulting architectures result in removal of hidden layer thereby reduce its computational complexity and generate learned word vectors. The first architecture called CBOW predicts a word surrounded by its context while the second architecture Continuous Skip-gram predicts the surrounding context for the given word in the vocabulary [12]. The two architectures are shown in Figure 1 below.

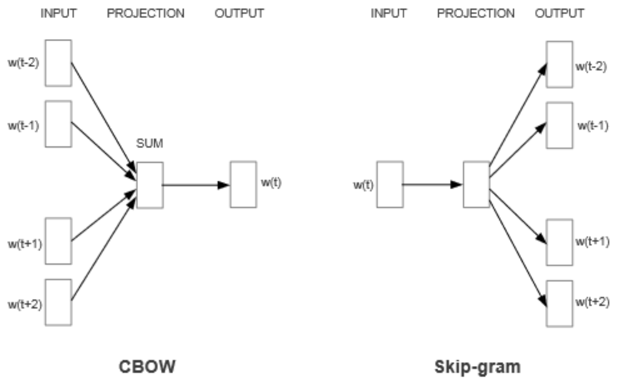

Fig. 1. CBOW and Skip-gram architectures (Figure reproduced from Mikolov et al. [12])

2.2.1 CBOW Architecture

CBOW is similar to feed-forward NNLM except for the hidden layer which is replaced by the average of all the word vectors from the projection layer, appearing before or after the target word in a window. Since the order of the words in the model is irrelevant, hence the model is referred to as 'bag of words'.

2.2.2 SG Architecture

The continuous SG architecture on the other hand acts as inverse of CBOW architecture. It inverts the prediction task by training the word vector representations to predict the target words in the context window for each word in the sentence. The larger the range of context window the higher the quality of generated word vectors. Moreover, due to lesser degree of relatedness of distant context words for the current word in the sentence, distant context words are weighed less in the model [12].

In subsequent work Mikolov et. al. [4] presented extensions of skip-gram model improving vector quality and the training speed through subsampling of frequent words, language understanding through additive compositionality representing linear structure of word vectors enabling mathematical operations to be performed.

Moreover, Mikolov et. al. [12] in his previous work used HS over softmax for normalizing terms and saving computational cost. However, in his later work, Mikolov et. al. [4] introduced NES to reduce the computational complexity associated with HS [20]. NES like previous Noise Contrastive Estimation (NCE) [21] makes it possible to distinguish the data from noise. It takes into account only limited noise distribution samples. A model trained with NES result in updating a limited number of words so as the model can predict "positive" observed word pair i.e. "washing machine" instead of a negative pair such as "washing refrigerator" or "washing air" which are most likely not to exist in text. In this paper, we compare the results of both architectures trained on HS and NES to identify how they perform on word similarity and analogy tasks.

3. SYSTEM OVERVIEW

A detailed overview of the system is provided addressing the problem definition, similarity tasks, working of the model, tools and technologies used, news article corpus collection and data set building.

3.1 Problem Definition

In this paper, the task of finding word similarity in text articles through word embeddings or word vector representations is performed. Since, word vectors generally tend to represent linear regularities both in syntactic and semantic form amongst words, hence this paper investigates if the model so implemented produces such linear regularities for words in news article data set. For this purpose, what role the linear structure of word vectors play in answering similar word pairs in analogy questions and in vector addition also known as vector compositionality is also investigated in the paper.

3.2 Word to Vector Model

The two architectures CBOW and skip-gram from word2vec model are used to learn the word vectors, by predicting target word from its context words in CBOW or otherwise predicting context words for each word from vocabulary in skip-gram[12]. Both architectures are implemented to establish which architecture better performs at capturing multiple degrees of similarity among words [11]. Models are trained with both HS and NES to find how two fare in generating quality of vector-representations by analyzing the results of tasks and in performance in terms of training time or computational complexity of learning vectors [4],[12].

3.3 Similarity Tasks:

In order to evaluate the linear regularities of word vectors, representing multiple degrees of similarity embedded in relative locations of words in form of syntactic and semantic relations, two similarity tasks are used to evaluate the system. The two tasks used in the paper are word similarity task and analogy task. Semantic regularity is evaluated for both tasks, however the syntactic regularity is evaluated only for analogy task.

The first similarity task is a simple semantic word similarity task. It is used to find most similar words for input words to make sense of the similar words intuitively. For e.g. ‘Germany’ is similar to ‘France’ and a few other countries geographically in Europe. In our work, we focus on finding the similar words to countries in news articles.

The second similarity task, an analogy task is however a little complex. It tests for both syntactic and semantic regularities to analogies in word vectors. Such questions are evaluated based on the vector offset technique recommended by Mikolov et. al. [11]. An analogy question can be represented as capital - country word - pair question. For example, Pakistan : Islamabad :: Afghanistan : x? Since, the vector space represents relationships between vectors in i.e. ‘Pakistan’ and ’Islamabad’ as vector offset ‘v’, where v = vector[Islamabad] – vector[Pakistan]. Therefore, other such pairs having some relationship would be related by around same vector offset ’v’ in continuous vector space; ‘v’ when added to word vector of ‘Afghanistan’ would yield a new vector say ‘y’ would be the exact answer, i.e. y = v + vector[Afghanistan]. However, such word may not necessarily be present at such position in vector space, where later highest by cosine similarity to ’y’ an equivalent output for X in the given analogy is identified [11].

In this paper, we evaluate both semantic and syntactic regularities of countries to capitals and adjective to adverbs in news articles data set respectively.

3.4 Vector Compositionality

Simple vector algebraic addition of word vectors, also called vector compositionality [4] may yield meaningful but not so obvious side of language understanding. For e.g. in skip-gram model if we add two word vectors i.e. vector[Afghanistan] + vector[India] it may yield a feature vector nearest vector[Bangladesh], because the word Bangladesh would appear frequently in the context of same sentence for words Afghanistan and India. Technically, both the input word vectors represent the context distributions, therefore technically the sum of two input word vectors in actual case represent the relevant product of context distribution of two vectors. In our experiment, we evaluate the vector addition of countries in CBOW architecture.

3.5 Dataset

The word2vec model is trained on daily dawn news data set prepared from extended dawn newspaper corpus [29] as used in [24]. The corpus comprises of 18640 news articles, collected articles from 12 different news subjects shown in Figure 2, which include Blogs, Business, Entertainment, Home, Magazine, Multimedia, Newspaper, Others, Pakistan, Sport, Tech and World. These news articles were published on news website in year 2015 and 2016.

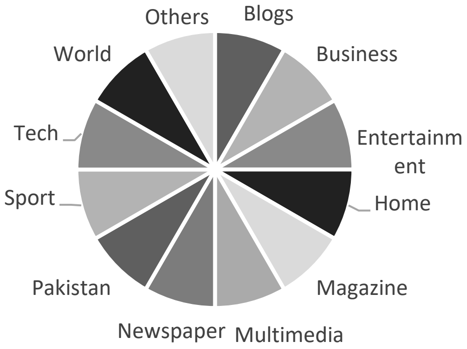

Fig. 2. Categories of News

The data set from news corpus was prepared for all articles. Each article was processed as a collection of sentences. Each sentence was preprocessed to drop any unwanted tokens such as punctuation marks. Finally, a total of 406,228 sentences were produced in the data set. The data set preparation was undertaken in Python, using python natural language toolkit (NLTK) [26]. Additionally, the model is also trained on set of 442 Wikipedia articles [30]., wherein it is used as a feature for establishing linguistic similarity between title and the body of text in such articles for comprehension of similarity from topical point of view.

Following section provides the results produced in respect of the underlying experimental setup performed over dawn news data set and accordingly the findings are presented.

3.6 Experimental Setting

The experiment is implemented in word2vec model in genism, a python library [28]. The vocabulary size of the data set produced is 53,980 unique words i.e. approximately 64K, where each unique word required a minimum frequency of 5 in articles.

3.7 Experimental Results

In this section, we evaluate CBOW and skip-gram architectures trained with HS and NES on word similarity task, linear regularities representing syntactic and semantic characteristics in analogy task and the vector compositionality. Both CBOW and skip-gram architectures are trained with the same hyper-parameters settings on a single Core i-5 machine having 4 cores of 2.2GHz. The hyper-parameters included dimensionality of 300 (size=300), context window size 5 (window=5) to the right and left side of current word and the minimum frequency of words equal to 5. The results of the models on semantic word similarity, semantic country-capital analogy, syntactic adjective-adverb analogy and vector compositionality are presented below.

3.7.1 Word Similarity Task

The word similarity, a first similarity task for given countries is evaluated in CBOW architecture trained with NES and HS. The results of top three, most similar target words for a given country as a context word are shown in Table 1 below:

Table 1: Similar words produced in cbow model trained with negative sampling & hierarchical softmax

Input Words	**Negative Sampling**		**Hiearchical Softmax**	
	Output Words	**Simil.**	**Output Words**	**Sim.**
Pakistan	**Store**	0.93595	**Pakistan’s**	0.85822
	'Bangladesh'	0.93570	'Bangladesh'	0.84979
	'Mobile'	0.93442	'India'	0.84378
Afghanistan	**Turkey**	0.95332	**India**	0.84377
	'Iraq'	0.94823	'Turkey'	0.83876
	'Nigeria'	0.94609	'Syria'	0.82413
India	**Bangladesh**	0.94159	**Bangladesh**	0.87245
	'China'	0.94104	'Australia'	0.85334
	'Afghanistan'	0.93158	'China'	0.84651

For the given input word country in HS, all the resulting similar words produced are countries, which proves the linear regularity of word vector representations in continuous vector space. However, in case of NES, ‘Store’ and ‘Mobile’ target words semantically don’t relate with the given context word ‘Pakistan’ as country. Therefore, semantically the accuracy and the quality of word vector representation produced with HS at the level of experimental setup settings are apparently better, therefore it yields better similar words.

3.7.2 Country-Capital Analogy Task

The country-capital is the first of two analogy tasks and part of the second similarity task, trained with NES and HS is evaluated in both CBOW and skip-gram architectures. The given analogy question, ‘Pakistan is to Islamabad therefore China is to what?’ is evaluated as vector algebraic operation as “vector[Islamabad] – vector[Pakistan] + vector[China]” fed to the cosine similarity to find the nearest cosine similar vector analogous answer word vector. The result of top three analogous answer word vectors in continuous vector space for CBOW and skip-gram architectures are shown in Table 2 & 3 respectively.

Table 2: Semantic analogy prediction through vector arithmetic in cbow architecture trained with negative sampling rate of 5 and hierarchical softmax

Input Words	**Negative Sampling**		**Hiearchical Softmax**	
	Output Words	**Sim.**	**Output Words**	**Sim.**
Pakistan : Islamabad :: Afghanistan: x	**Kabul**	0.97324	**Kabul**	0.81742
	'Turkey'	0.97073	'Turkey'	0.81021
	'Riyadh'	0.96332	'Syria'	0.80940
Pakistan : Islamabad :: Iran : x	**Tehran**	0.99238	**Tehran**	0.87596
	'Turkey'	0.97741	'Riyadh'	0.84454
	'Russia'	0.96750	'Turkey'	0.82674
China : Beijing :: Iran : x	**Tehran**	0.97403	**Tehran**	0.89427
	'Russia'	0.95038	'Moscow'	0.82675
	'Moscow'	0.94696	'Washington'	0.81486

For the given capital-country analogy task word vectors trained with both NES and HS in CBOW perform equally well by predicting the right analogous capital city. This shows that the resulting offset of the vector algebraic operation represents a linear regularity with the cosine similar words in vector space representing semantic similarity with cities or countries.

Table 3: Semantic analogy prediction through vector arithmetic in skip-gram model trained with negative sampling rate of 5 and hierarchical softmax

Input Words	**Negative Sampling**		**Hiearchical Softmax**	
	Output Words	**Sim.**	**Output Words**	**Sim.**
Pakistan : Islamabad :: Afghanistan : x	**‘Kabul'**	0.89159	**'Kabul'**	0.90703
	'Afghanistan'	0.81725	'Afghan'	0.83643
	'Nangarhar'	0.80643	'Helmand'	0.83047
Pakistan : Islamabad :: Iran : x	**'Tehran'**	0.93138	**'Tehran'**	0.95679
	'Riyadh'	0.91533	'Riyadh'	0.85452
	'Vienna'	0.85509	'Vienna'	0.84693
China : Beijing :: Iran : x	**'Tehran'**	0.89163	**'Tehran'**	0.90802
	'Tehran’s'	0.86549	'Moscow'	0.86184
	'Khamenei'	0.85972	"Iran's"	0.81001

Skip-gram model albeit predicts the correct capital analogies for all the given countries with both NES and HS, however one of the third ranked analogy for Iran’s capital is presented as ‘Khamenei’ with NES. Therefore, both the architectures trained with both techniques represent the semantic similarity through mathematical operations on vectors.

3.7.3 Adjective-Adverb Analogy Task

The adjective-adverb is the second of two analogy tasks and part of the second similarity task, trained with NES and HS is evaluated in both CBOW and skip-gram architectures and the result of top three analogous answer word vectors in continuous vector space for CBOW and skip-gram architectures are shown in Table 4 & 5 respectively.

Table 4: Syntactic analogy prediction through vector arithmetic in cbow model trained with negative sampling rate of 5 and hierarchical softmax

Input Words	**Negative Sampling**		**Hiearchical Softmax**	
	Output Words	**Sim.**	**Output Words**	**Sim.**
recent : recently :: initial : x	**Initially**	0.9291	**initially**	0.8386
	'previously'	0.9250	'already'	0.8090
	'already'	0.9170	'successfully '	0.8022
actual : actually :: recent : x	**Recently**	0.9724	**recently**	0.8989
	'REFEREE'	0.9716	'later'	0.8923
	'never'	0.9666	'ago'	0.8904
interesting : interestingly :: important:x	**Importantly**	0.8032	**importantly**	0.7325
	'variants'	0.7898	'Amazingly'	0.7252
	'politics”'	0.7822	'Sârbu'	0.7117

For the given adjective-adverb analogy task word vectors trained with both NES and HS in CBOW perform equally well by predicting the right analogous adverb city. This shows that the resulting offset of the vector algebraic operation represents a linear regularity with the cosine similar words in vector space representing syntactic similarity with mainly adverbs or other relevant grammatical forms of similar words.

Table 5: Syntactic analogy prediction through vector arithmetic in skip-gram model trained with negative sampling rate of 5 and hierarchical softmax

Input Words	**Negative Sampling**		**Hiearchical Softmax**	
	Output Words	**Sim.**	**Output Words**	**Sim.**
recent : recently :: initial : x	**Initially**	0.8167	**initially**	0.8200
	'PC-1'	0.7992	'Rs300m'	0.7618
	'preliminary'	0.7978	'Inquiries'	0.7567
actual : actually :: recent : x	**Recently**	0.7959	**recently**	0.8155
	'lately'	0.7716	"Hamza's"	0.7908
	'dismayed'	0.7694	'careful”'	0.7859
	'“now'	0.9038	***'worryingly'***	0.7577
interesting : interestingly:: important : x	'failed”'	0.8934	'ineffectiveness'	0.7510
	'flouted'	0.8895	"Ja'afari"	0.7507

However, skip-gram model fails at predicting the same adjective-adverb analogy for ‘important’ adjective with HS and NES. Otherwise the performance with two techniques is found same.

3.7.4 Vector Compositionality

The results of vector addition over pair of countries is evaluated with NES and HS in CBOW. The results for top three words in continuous vector space for the given two vectors from news corpus are shown in Table 6 below:

Table 6: Vector compositionality or vector addition representating linear structure produced in cbow model trained with negative sampling rate of 5 and hierarchical softmax

Input Words	**Negative Sampling**		**Hiearchical Softmax**	
	Output Words	**Sim.**	**Output Words**	**Sim.**
Afghanistan / Pakistan	**Bangladesh**	0.87827	**India**	0.71196
	'Store'	0.87013	'Bangladesh'	0.68241
	'mobile'	0.86659	'China'	0.67539
India / Afghanistan	**China**	0.88613	**Bangladesh**	0.70061
	'Bangladesh'	0.88380	'China'	0.69586
	'Turkey'	0.87686	'Turkey'	0.68572
China / Russia	**Turkey**	0.91956	**Iran**	0.71563
	'Japan'	0.91849	'Turkey'	0.70507
	'Iran'	0.91124	'India'	0.69909

With both techniques, the result shows the vector addition of the context words also produce all countries. This suggests that vector[China] + vector[Russia] are more likely to appear as context words in the same sentence with Turkey than Japan or Iran as the target words. The results of vector compositionality are meaningful in respect of the news articles but not easily or directly understood from language point of few.

3.7.5 Comparison by Training Time

Both architectures are trained with HS and with NES at four different sample rates with the recommended range of 5-20[4]. A summary of training performance of two architectures is presented in Table 7 and Figure 3 below:

Table 7: Training comparison of negative sampling and hierarchical softmax in cbow and skip-gram architectures

Method	**CBOW**		**Skip-gram**	
	Training Time (secs)	**Memory (~MBs)**	**Training Time (secs)**	**Memory (~MBs)**
NEG-5	147.529s	156.542	503.295s	156.542
NEG-10	206.609s	156.542	934.717s	156.542
NEG-15	262.504	156.542	1343.478s	156.542
NEG-20	302.508s	156.542	1725.677s	156.542
HS-S	366.809s	232.114	2225.559	232.114

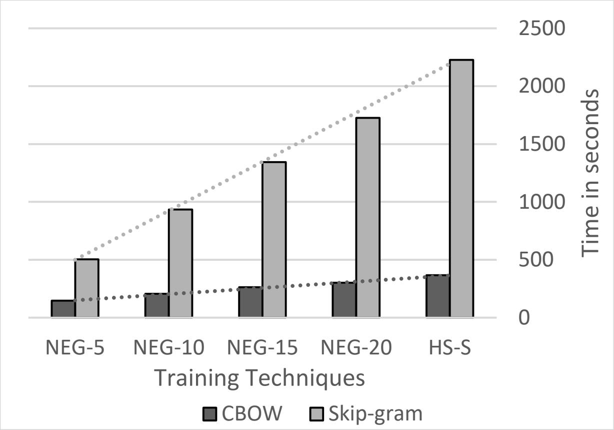

Fig. 3 Training Comparison of CBOW and Skip-gram Architectures

The results show that computational complexity in terms of required memory for training and the training time in both architectures for NES irrespective of sampling size, are far lower than HS, the same was reported by Mikolov et. al. in [4]. However, the training time of NES is faster than HS by a factor of 1.21 to 2.48 and by a factor of 1.28 to 4.42 in CBOW and skip-gram architectures respectively.

4. Conclusion

We presented how word vector representations generated in deep learning word2vec model based on CBOW or skip-gram architectures can be used in NLP for establishing word similarities between words in dawn news articles. The word vectors represented both semantic and syntactic similarities. The findings of word similarity task of countries evaluated in CBOW, country-capital analogy task evaluated in both CBOW and skip-gram models showed semantic similarity. Similarly, adjective-adverb analogy task showed syntactic similarity in both architectures. Finally, vector compositionality or vector addition of pair of countries task evaluated in CBOW showed semantically meaningful similarity in vector space. NES technique is faster, less computation intensive and less memory intensive than HS. Therefore, deep learning enables harnessing voluminous data and extensive computation with either no or minor engineering by hand [22]. As future work, the relationship between dimensionality and accuracy of prediction needs to be undertaken and find an optimal converge point for embedding dimensions. Moreover, a comparative study of other models such as GloVe[23] and FastText[5] would be undertaken.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] E. Cambria and B. White, “Jumping NLP curves: A review of natural language processing research,” IEEE Computational Intelligence Magazine, vol. 9, no. 2, pp. 48–57, 2014.

[2] T Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria, Recent Trends in Deep Learning Based Natural Language Processing. (2018). arXiv:arXiv:1708.02709v5

[3] T. Mikolov, M. Karafi´at, L. Burget, J. Cernock`y, and S. Khudanpur, “Recurrent neural network based language model.” in Interspeech, vol. 2, 2010, p. 3.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in neural information processing systems, 2013, pp. 3111– 3119.

[5] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts et al., “Recursive deep models for semantic compositionality over a sentiment treebank,” in Proceedings of the conference on empirical methods in natural language processing (EMNLP), vol. 1631, 2013, p. 1642.

[6] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning, 2007

[7] G.E. Hinton, J.L. McClelland, D.E. Rumelhart. Distributed representations. In: Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations, MIT Press, 1986.

[8] Y. Bengio, R. Ducharme, P. Vincent, A neural probabilistic language model. Journal of Machine Learning Research, 3:1137-1155, 2003.

[9] H. Schwenk, Continuous space language models. Computer Speech and Language, vol. 21, 2007.

[10] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, J. Cˇ ernocky, Empirical Evaluation and Combination of Advanced Language Modeling Techniques, In: Proceedings of Interspeech, 2011

[11] T. Mikolov, W.T. Yih, G. Zweig. Linguistic Regularities in Continuous Space Word Representations. NAACL HLT 2013.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

[13] David E Rumelhart, Geoffrey E Hintont, and Ronald J Williams. Learning representations by backpropagating errors. Nature, 323(6088):533–536, 1986

[14] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. In Journal of Artificial Intelligence Research, 37:141-188, 2010.

[15] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM, 2008.

[16] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three, pages 2764–2770. AAAI Press, 2011.

[17] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In ICML, 513–520, 2011.

[18] Peter D. Turney. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. In Transactions of the Association for Computational Linguistics (TACL), 353–366, 2013.

[19] A. Zhila, W.T. Yih, C. Meek, G. Zweig, T. Mikolov. Combining Heterogeneous Models for Measuring Relational Similarity. NAACL HLT 2013.

[20] F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In R. G. Cowell and Z. Ghahramani, editors, Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pages 246–252. Society for Artificial Intelligence and Statistics, 2005.

[21] Gutmann, Michael U. and Hyvärinen, Aapo. Noisecontrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. J. Mach. Learn. Res., 13(1):307–361, February 2012. ISSN 1532-4435. 00247.

[22] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning. Nature, vol. 521, no. 7553, pp. 436–444, 2015

[23] Pennington, Jeffrey, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empiricial Methods in Natural Language Processing (EMNLP).

[24] Adeel Ahmed, Syed Saif ur Rahman, “DBpedia based Ontological Concepts Driven Information Extraction from Unstructured Text” International Journal of Advanced Computer Science and Applications(ijacsa), 8(9), 2017.

[25] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[26] Natural Language Toolkit: https://www.nltk.org/

[27] Firth, JR, 'A Synopsis of Linguistic Theory, 1930-55.

[28] Python Gensim Library: http://pypi.org/project/gensim/

[29] The Dawn Newspaper: http://www.dawn.com/

[30] P. Rajpurkar, R. Jia, and P. Liang, “Know what you don’t know: Unanswerable questions for SQuAD,” ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 2, pp. 784–789, 2018

[31] Khashabi, Daniel, "Reasoning-Driven Question-Answering For Natural Language Understanding" (2019). Publicly Accessible Penn Dissertations. 3271. https://repository.upenn.edu/edissertations/3271

[32] Ferrone L and Zanzotto FM (2020) Symbolic, Distributed, and Distributional Representations for Natural Language Processing in the Era of Deep Learning: A Survey. Front. Robot. AI 6:153. doi: 10.3389/frobt.2019.00153

[33] L. Song, Z. Wang, and W. Hamza, “A Unified Query-based Generative Model for Question Generation and Question Answering,” 2017.

[34] L. Pan, W. Lei, T.-S. Chua, and M.-Y. Kan, “Recent Advances in Neural Question Generation,” no. 3, 2019