Analysis of word embeddings with Graph-Based Context Adaptation for Enhanced Word Vectors

Tanvi Sandhu
School of Computer Science
University of Windsor

Zaid Kobti
School of Computer Science
University of Windsor

Abstract

In the aspect of information storage, text assumes a central role, necessitating streamlined and effective methods for swift retrieval. Among various text representations, the vector form stands out for its remarkable efficiency, especially when dealing with expansive datasets. This paper explores the intersection of data representation in vector form and the heightened performance and accuracy observed in Natural Language Processing (NLP) tasks, employing dynamic embedding models enriched with graph structures. The investigation delves into the merits of vectorized text in NLP, extending to the incorporation of graphs within vectors to enhance overall capabilities in information representation and retrieval. The study employs graph analysis to reveal word relatedness, utilizing a vertex embedding method for generating embeddings. Experimental deployment of this technique across diverse text corpora underscores its superiority over conventional word-embedding approaches. Furthermore, cutting-edge NLP techniques, such as contextual word embeddings from models like ELMo and GPT, are seamlessly integrated to augment text classification. Unlike traditional static embeddings, contextual embeddings consider the specific context in which a word appears, offering distinct representations for words across different contexts. This adaptability to surrounding context addresses limitations in capturing the richness of language semantics present in static word vectors. This research not only contributes valuable insights into advanced word representation methodologies but also sheds light on their implications for text classification tasks, especially within the context of dynamic embedding models. The holistic perspective provided in this paper aims to advance the understanding of optimal information representation and retrieval strategies in the dynamic landscape of NLP.

Introduction

Natural Language Processing (NLP) is a rapidly evolving field that plays a crucial role in various applications, including chatbots, virtual assistants, and language translation services. One of the challenges in NLP is handling the diversity and complexity of human language, including idioms, slang, and cultural variations.

In Natural Language Processing (NLP), the representation of words is a fundamental aspect that bridges the gap between linguistic elements and computational models. Words are typically transformed into numerical vectors, known as word embeddings. These embeddings encode semantic and syntactic information, capturing the relationships between words in a high-dimensional space. The significance of word representation lies in its ability to preserve contextual meaning and nuances, enabling machines to comprehend language based on the surrounding context. This numerical representation facilitates various NLP tasks, including word similarity measurement, sentiment analysis, and language translation.

NLP models like Word2Vec $(Mikolovetal., 2013)$ and Glove $(Pennington, J., 2014)$ were focused on capturing distributed representations of words based on co-occurrence statistics. These static embedding models do not consider the context in which words appear, leading to challenges in capturing nuances and variations in meaning depending on the surrounding words. Whereas, in the contextual word embedding models, the embedding of a word is context-dependent, meaning it varies based on the surrounding words and the overall sentence structure. The model captures the nuanced meanings of words in different contexts. For instance, models like ELMo (Embeddings from Language Models) $(Petersetal., 2018)$ and GPT (Generative Pre-trained Transformer) generate contextual embeddings, capturing word meanings based on the surrounding context in a sentence. This contextual understanding contributes to more accurate language representation.

Unlike traditional word embeddings that assign a fixed vector to each word, contextual embeddings take into account the context in which a word appears within a sentence. ELMo $(Petersetal., 2018)$, for instance, employs deep contextualized word representations by considering the entire input sentence. It utilizes a bidirectional LSTM (Long Short-Term Memory) to capture the context on both the left and right sides of a word. This results in embeddings that vary depending on the specific instance of a word within a sentence, addressing challenges posed by polysemy and context-dependent meanings. Similarly, GPT, a state-of-the-art transformer-based model, adopts a pre-training strategy

to generate contextual embeddings. During the training process, the GPT model takes into consideration the entire context of a word which seizes the intricate patterns and dependencies.

Consider the word "bat" in two sentences:

"He swung the bat and hit a home run"

"At dusk, bats emerge from their caves to hunt for insects."

In the first sentence, "bat" is associated with sports equipment, while in the second, it refers to a flying mammal. Traditional word vectors find it difficult to get the context hence it will not form different vectors for word bat. It will form one static embedding based on the average meaning in both contexts like below:

$$\text{Vector}_{static}\ \text{bat}$$

Here the one static vector of "bat" is formed because "bat" is treated as one point in space irrespective of its meaning in the two sentences. This limitation is addressed by contextual word embeddings, which consider the surrounding context of a word in a sentence. Below are the vectors generated for the word bat:

$$\text{Vector}_{contextual}\ \text{bat}^{sentence1}$$
$$\text{Vector}_{contextual}\ \text{bat}^{sentence2}$$

Each of these vectors has distinct representations capturing the contextual meaning of the word "bat" in the respective sentences. Unlike traditional static embeddings, which provide a single, averaged representation for a word across different contexts, contextual word embeddings, such as those obtained from models like ELMo $(Petersetal., 2018)$, BERT $(Devlin, J., 2018)$, or GPT, consider the specific context in which the word appears.

This allows the model to generate distinct embeddings for "bat" based on its specific usage, enabling more accurate representation and comprehension of the word's varied meanings in different contexts. The ability of contextual embeddings to adapt to the surrounding context makes them valuable in addressing the limitations of static word vectors in capturing the richness of language semantics.

Semantic textual similarity constitutes a great aspect of focus in the field of Natural Language Processing having applications like information retrieval, question answering, and text summarization. The emergence of advanced pre-trained language models, including the Universal Sentence Encoder and RoBERTa (Robustly optimized BERT approach) $(LiuY, 2019)$, reflects significant progress in capturing similarity not only at the word level but also in the semantic alignment of entire sentences or paragraphs. This broader scope provides a more profound comprehension of the relationships embedded within textual content.

In contemporary research, attention mechanisms have garnered notable attention within NLP models. These mechanisms empower models to selectively emphasize specific segments within input sequences, enabling them to assess the importance of different words or subwords dynamically. The adoption of attention-driven strategies enhances a model's capability to handle extended dependencies effectively, resulting in improved performance across tasks such as machine translation and text summarization.

The integration of graph-based representation learning with NLP models extends the capabilities of traditional approaches. The graph helps us to visualize the word embedding in a more structured way. It can also help us handle polysemy or homonymy challenges. The use of weights on edges in a graph-based word embedding model plays a crucial role in capturing the strength or importance of relationships between words.

In conclusion, this paper delves into the dynamic landscape of Natural Language Processing (NLP) and its pivotal role in various applications, ranging from chatbots and virtual assistants to language translation services. The challenges posed by the diversity and complexity of human language, including idioms, slang, and cultural variations, have spurred continuous advancements in the representation of words – a fundamental aspect that bridges linguistic elements with computational models. Traditional static embeddings, as exemplified by models like word2vec $(Mikolovetal., 2013)$ and GloVe $(Pennington, J., 2014)$, face limitations in capturing nuanced meanings due to their lack of contextual awareness. The proposed paper seeks to address these limitations by exploring the realm of contextual word embedding and the broader scope of semantic textual similarity. Ultimately, the proposed paper aims to contribute to the evolving landscape of NLP research by providing insights into overcoming existing challenges and leveraging innovative approaches to enhance language understanding and representation.

Problem Statement

The mathematical problem formulation of the problem statement is as follows:

Task 1: Given a text Corpus C containing n words, find a graphical representation G where each word pair acts as nodes and each node is connected with a weight w. This graph G is derived from matrix M[n][n] after formulating a semantic relationship, where n is the number of words after preprocessing the text corpus C

Task 2: Update the initial weight of edges to get the best of the context.

Literature Review

NLP models like Word2Vec $(Mikolovetal., 2013)$, introduced by Mikolov et al, used neural networks to learn word embeddings, while GloVe $(Pennington, J., 2014)$ employed global statistics to create embeddings. Predicting the next word in Natural Language Processing (NLP) is a tough job because language is complex. Words can have different meanings based on the context in which they are used. Sentence structures vary a lot, and there are many

words that mean almost the same thing. Figuring out the next word or linking two words together is not easy. Also, words often have multiple meanings, adding more complexity. To deal with these challenges, smart models and techniques, like neural language models with attention mechanisms, have been created to understand the patterns and connections in language. Bengio, Ducharme, and Vincent tried to predict the next word in their 2003 paper, 'A Neural Probabilistic Language Model.' They used a neural network with Long Short-Term Memory (LSTM) units to figure out how words work together in sentences. The model assigned an index to each unique word and used a feed-forward network to predict the next word based on previous ones. It was good at understanding long connections between words, but it had some problems, like being computationally complex and needing a lot of examples to learn well.

In the world of NLP, connecting important words is like an evolution story. From basic sparse encoding to advanced contextual embeddings, the journey has changed how we understand and use words. Word vectors have been crucial in solving this problem.

The development of word vectors in NLP has followed a changing path, showing progress in neural network designs. Early methods using sparse vectors like one-hot encoding couldn't capture the rich meaning in language. Then came distributional semantic models like CBOW and skip-gram, which allowed creating dense vectors that understand complex relationships in language. Pre-trained word embeddings like Word2Vec $(Mikolovetal., 2013)$, GloVe $(Pennington, J., 2014)$, and fastText became important, giving efficient representations learned from lots of data. Now, contextualized embeddings like those in BERT $(Devlin, J., 2018)$ have become popular, improving word representations by understanding context and word frequencies in documents. This ongoing evolution shows a continuous effort to have better and context-aware word vector representations in NLP.

Exploring dynamic word embeddings has also caught attention. Papers like 'Dynamic Word Embeddings for Evolving Semantic Discovery' (2016) by Rudolph et al. talk about word embeddings that adapt to changing meanings over time, giving a new perspective for NLP applications.

As the field advances, using graph-based representations has become a new idea. 'Using Graphs for Word Embedding with Enhanced Semantic Relations' by Last et al. introduces WordGraph2Vec, a word embedding algorithm that combines the good parts of existing methods while reducing their problems. This algorithm uses both vectors and graphs in training, showing potential in finding semantic relationships and classifying text. The algorithm relies on several parameters such as p, q, and the choice of word weights (TF or TF-IDF). The sensitivity of the algorithm to these parameters may make it challenging for users to find optimal settings for different types of corpora.

In another notable paper, "Enhancing Word Embeddings with Knowledge Graph Embeddings for Named Entity Recognition" by Wang et al., the focus shifts towards enhancing word embeddings with knowledge graph embeddings specifically for named entity recognition (NER). Named entity recognition is a crucial task in NLP aimed at identifying and categorizing named entities such as persons, organizations, and locations within text. The paper proposes a novel approach that integrates knowledge graph embeddings with word embeddings to improve the performance of NER systems. By leveraging the structured knowledge encoded in knowledge graphs, such as Wikipedia or Freebase, the proposed method enriches word embeddings with semantic information, enhancing their ability to capture the context and semantics of named entities. However, despite the promising advancements in enhancing word embeddings with knowledge graph embeddings for named entity recognition (NER) as presented in Wang et al.'s work, there are notable limitations. One significant challenge is the effectiveness of the method heavily relies on the quality and coverage of the underlying knowledge graph. In cases where the knowledge graph lacks comprehensive information or contains inaccuracies, the performance of the proposed approach may be compromised. Moreover, the model's generalization to diverse domains and languages may be constrained by the specificity of the incorporated knowledge graph. Finding a balance between leveraging external knowledge and addressing these scalability and generalization challenges remains an ongoing area of exploration and improvement in the field of natural language processing.

Methodology

There are 2 main steps considered in the paper:
Graph construction using adjacent matrix,
and the Graph Enhancement.

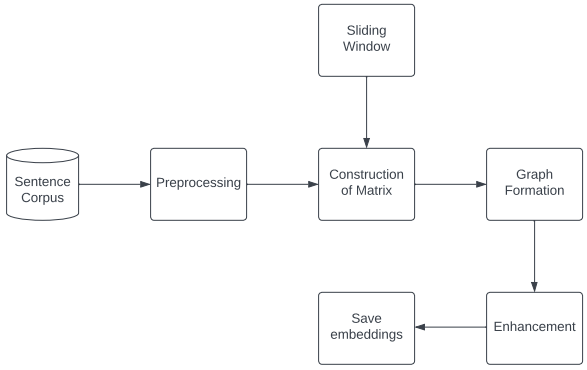

Figure 1: A Flowchart of the methodology

The first step of the formation of the graph is one of the important step as it will determine the semantic relationship of the text. Graph is basically the structural representation of the input dataset. Graph has three components, namely nodes, edges and weight. The nodes of this graph is comprised of the vocabulary generated. This vocabulary is generated with the input dataset of sentences. Figure 1 shows the flow of the methodology. First, the text corpus containing sentences is broken into a list of words which is then

preprocessed to a tokenized and lemmatized format. Stopwords are removed, and a vocabulary is constructed. The edges of the graph is the relationship among these words acting as nodes. The relationship is defined as a semantic relationship. Here sliding window concept is used to define the relation.

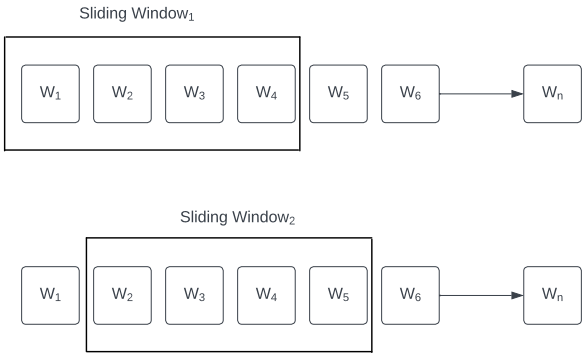

Figure 2: Sliding Window Concept

A sliding window is basically the sliding of a fixed window over the text corpus to find the local features or patterns between words. As the window slides on the data, the text within the window is considered as an input text for calculating the frequency of that word pair. This frequency is symmetrically populated into the matrix. This frequency acts as a weight in the graph. Hence after the graph formation, there are no redundant edges in the graph if there are no relationships among the words.

In the above Figure 2, W $_1$, W $_2$, W $_3$, W $_4$, and so on represents a set of words which are derived from the corpus. A sliding window of size 4 is shown on the initial four words. The words in this window act as an input to calculate the relationship which will be updated in teh matrix. This window then slides from W $_1$ to W $_2$ in the next iteration. The co-occurrence matrix is required as a base for the graph generation. Lets consider co-occurrence matrix M, where V is the vocabulary size of corpus C. Hence the matrix generated is M(V×V).

Graph enhancement is the next step in the process. There are many preexisting steps to enhance a graph like Node2Vec, Word2Vec $(Mikolovetal., 2013)$ and many more. The technique used here is Deepwalk $(PerozziB., 2014)$ and LINE $(TangJ., 2015)$. DeepWalk's emphasis on preserving local structure through random walks aligns with the inherent intricacies of semantic relationships among words in the input dataset. By treating the vocabulary as nodes and simulating random walks, DeepWalk $(PerozziB., 2014)$ effectively learns embeddings that encapsulate the contextual meaning of words. On the other hand, LINE $(TangJ., 2015)$ complements this approach by optimizing both local and global structures of the graph, offering a comprehensive representation of semantic associations. The efficiency of LINE $(TangJ., 2015)$ in handling large-scale graphs is particularly advantageous, ensuring scalability and performance even with extensive vocabularies. The combination of DeepWalk $(PerozziB., 2014)$ and LINE $(TangJ., 2015)$ thus contributes to a robust methodology for enhancing the initially constructed graph, providing a refined semantic representation that captures the subtle relationships among words in the input text corpus.

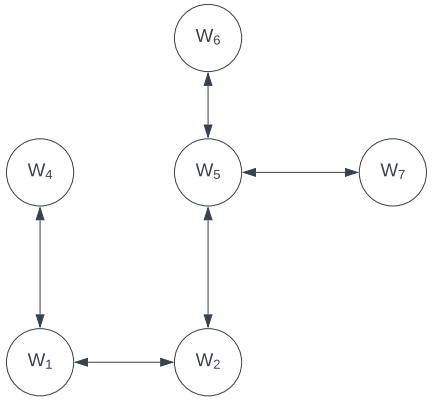

Figure 3: Graph Generated

Figure 3 depicts the graph generated from the dataset where words W $_1$, W $_2$, W $_3$, W $_4$, and so on represent nodes of the graph. The edges are defined only for the nodes having a weight in the matrix, hence no redundant edges are present in this graph.

Datasets

SimLex-999 $(FelixH., 2015)$ has emerged as a widely utilized resource for monitoring word similarity. This benchmark dataset focuses on assessing the similarity between words, distinct from relatedness or association. It comprises 666 noun-noun pairs, 222 verb-verb pairs, and 111 adjective-adjective pairs. The dataset serves the purpose of evaluating word similarity, involving an exploration of the relationship between two words based on the algorithm used and subsequent comparison with human-annotated references. The performance of the task is measured using the Spearman Correlation as the evaluation metric.

The SemEval-2012-Task2 $(DavidA, 2012)$ dataset stands out as a significant asset in the realm of natural language processing, specially crafted for the assessment of Word Analogy. This dataset is centered around appraising the level of similarity between pairs of words, presenting a varied collection of lexical items for a thorough evaluation. Covering a range of tasks related to word similarity across diverse languages and domains, the dataset comprises annotated word pairs, with human raters assigning similarity scores. The evaluation of this task relies on the use of Spearman Correlation to gauge the alignment between model predictions and the similarity scores provided by human annotators.

Baselines

DeepWalk ($PerozziB., 2014$) is engineered to acquire continuous vector representations, known as embeddings, for nodes within a network. These embeddings are adept at encapsulating both the structural intricacies and interconnections among nodes in the network, effectively placing them within a continuous vector space. In the context at hand, DeepWalk ($PerozziB., 2014$) is applied to pre-saved embeddings, playing a pivotal role in social network analysis. The algorithm undertakes random walks on the graph, employing a Word2Vec ($Mikolovetal., 2013$) like methodology to learn distributed representations for each node. Consequently, these node embeddings encapsulate both the structural layout and contextual nuances of the nodes in the graph.

LINE ($TangJ., 2015$), as a network embedding algorithm, surpasses the limitations of exclusively capturing local proximity. Its objective is to safeguard both first-order, signifying direct connections, and second-order, indicating common neighbor relationships, proximity among nodes. This is accomplished through the formulation of an objective function that seeks to maximize the likelihood of observing existing edges while minimizing the likelihood of non-existing edges.

Results

Models like Deepwalk ($PerozziB., 2014$) and LINE ($TangJ., 2015$) were used to generate node embeddings, capturing semantic information about the words in the corpus. These embeddings are then saved for further use to compare these with the other advanced language models like BERT ($Devlin, J., 2018$), XLNET ($Yang, Z., 2019$) and GPT-2. The datasets SimLex999 ($FelixH., 2015$) and Semeval ($DavidA, 2012$) are used in the experiments to compare the word embeddings. The Table 1 shows the value comparison of the results with Spearmann Correlation as the comparison metric.

	Simlex-999	SemEval
BERT	0.0143	0.012
GPT	0.034	0.019
XLNET	0.080	0.022
Graph+DEEPWALK	0.114	0.097
Graph+LINE	0.053	0.062

Table 1: Comparison of Graph generated embedding with other model using Spearmann Correlation results for evaluation

The Table 1 presents a comprehensive comparison of word embedding models, assessing their performance in word similarity tasks across diverse datasets—SimLex-999 and SemEval. The models under scrutiny include prominent ones like BERT, GPT, XLNET, and two innovative approaches integrating graph-based context adaptation (Graph+DEEPWALK, Graph+LINE). The evaluation metric is Spearman Correlation, measuring the alignment between model predictions and human-annotated similarity scores. Higher correlation values indicate better proficiency in capturing semantic relationships among words. Notably, Graph+DEEPWALK outshines other models in SimLex-999 datasets, showcasing its efficacy in enhancing word vector's semantic understanding. GPT and XLNET also demonstrate competitive performances. However, the graph-based models, which leverage structural information from semantic graphs, exhibit a nuanced understanding of word relationships, particularly evident in SemEval. The table underscores the importance of graph-based context adaptation in refining word embeddings, offering insights into model's abilities to comprehend the intricacies of semantic connections.

Conclusion

In conclusion, this paper significantly adds to the dynamic landscape of Natural Language Processing (NLP) by presenting an innovative method for word representation, achieved through the creation and refinement of semantic graphs. This approach, which synergizes the capabilities of graph-based representation learning and advanced embedding techniques, outperforms existing models in word similarity tasks. Moreover, the comparison with established models vividly underscores the effectiveness of the proposed approach. It underscores the crucial aspect of capturing contextual nuances for precise language representation. This emphasis on context-driven understanding positions the proposed methodology as a noteworthy advancement in NLP. Ongoing advancements in NLP, as exemplified by this study, continuous progress strives to navigate the intricate challenges of human language, fostering the development of more precise and adaptable language models with applications across a multitude of fields and domains.

References

Aminul Islam and Diana Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (TKDD), 2(2), 1–25, (2008).

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

Marco Baroni and Alessandro Lenci, 'How we blessed distributional semantic evaluation', in Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, pp. 1–10, (2011).

Romanova, A. (2021). Semantics graph mining for topic discovery and word associations. International Journal of Data Mining and Knowledge Management Process, 11(04), 01–14. https://doi.org/10.5121/ijdkp.2021.11401

Kun Jing and Jungang Xu. A survey on neural network language models', arXiv preprint arXiv:1906.03591. (2019).

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, (1972).

Pennington, J., Socher, R., and Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics 41(4):665–695.

David A Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. Association for Computational Linguistics, pages 356–364.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved from https://arxiv.org/abs/1810.04805

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. Retrieved from https://arxiv.org/abs/1906.08237

Perozzi B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk: Online Learning of Social Representations. Retrieved from https://arxiv.org/abs/1403.6652

Reimers, N., and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Retrieved from https://arxiv.org/abs/1908.10084

Tang J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). LINE: Large- scale Information Network Embedding. Retrieved from https://arxiv.org/abs/1503.03578

Ken Lang, Newsweeder: Learning to filter netnews, in Machine Learning Proceedings 1995, 331–339, Elsevier, (1995).

Rudolph, M., Blei, D. M., and Griffiths, T. L. (2016). Dynamic Word Embeddings for Evolving Semantic Discovery. Retrieved from https://arxiv.org/abs/1703.00607

Last, M., Klein, T., and Kandel, A. (2019). Using Graphs for Word Embedding with Enhanced Semantic Relations. Retrieved from https://arxiv.org/abs/1911.01678

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D. and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Zuckerman, M., and Last, M. (2019, November). Using graphs for word embedding with enhanced semantic relations. In Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13) (pp. 32-41).