Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 3s (2025)

A Review: Word Embedding Models with Machine Learning Based Context Depend and Context Independent Techniques

Kapil Adhar Wagh[1],Dr. Anupa Sinha[2]
[1]Affiliation: Kalinga University, Naya Raipur, Email: kapilwagh2686@gmail.com
ORCID ID:0000-0002-7741-6050
[2]Affiliation: Asst. Professor, Faculty of CS & IT , Kalinga University, Naya Raipur,
Email: anupa.sinha@kalingauniversity.ac.in
ORCID: 0000-0002-0390-4406

Article History:

Received: *02-10-2024*

Revised: *22-11-2024*

Accepted: *01-12-2024*

Abstract:

Natural language processing (NLP) has been transformed by word embedding models, which convert text into meaningful numerical representations. These models fall into two general categories: context-dependent methods like ELMo, BERT, and GPT, and context-independent methods like Word2Vec, GloVe, and FastText. Although static word representations are provided by context-independent models, polysemy and contextual subtleties are difficult for them to capture. These issues are addressed by context-dependent approaches that make use of sophisticated deep learning architectures to produce dynamic embeddings that are impacted by the surrounding text. This review highlights the machine learning underpinnings of these paradigms while examining their development, approaches, and comparative advantages. We examine their benchmarks, applications, and the trade-offs associated with various use cases. The study also identifies future research directions, such as hybrid embeddings and multimodal learning, and highlights contemporary issues, such as scalability and interpretability. The goal of this thorough review is to assist practitioners and researchers in choosing and refining embedding strategies for a variety of NLP tasks.

Keywords: Word Embedding, Supervised learning, unsupervised learning, reinforced learning

1. Introduction:-

Word embedding models have emerged as a key component of natural language processing (NLP) developments in recent years. These models help machines better comprehend and process human language by converting textual data into dense, high-dimensional numerical vectors. Due to their high dimensionality and incapacity to capture semantic relationships, early word representation techniques like one-hot encoding had drawbacks. By embedding words into continuous vector spaces, word embedding models became a revolutionary solution to these problems [1].

The two main categories of word embedding techniques are context-dependent and context-independent models. Word2Vec, GloVe, and FastText are examples of context-independent models that generate static embeddings, in which every word is represented by a single vector independent of its contextual usage. Although these models are good at capturing semantic relationships, they are unable to handle polysemy and context-based meaning variations. Conversely, context-dependent models—such as ELMo, BERT, GPT, and their variations—produce dynamic embeddings that adapt to the surrounding text, providing notable enhancements in tasks that call for contextual comprehension [2].

The efficacy of these models has been further improved by the incorporation of machine learning methods, especially deep learning. In order to evaluate sequential data and extract subtle features, contextual embedding models mainly rely on architectures such as transformers and recurrent neural networks (RNNs) [3].

The goal of this paper is to present a thorough analysis of word embedding models, emphasising their development, underlying principles, and NLP applications. We illustrate the advantages, disadvantages, and applicability of context-dependent and context-independent approaches for a range of use cases. In addition, we provide insights into the future of word representation by talking about the difficulties and new developments in the field [4].

2. Background:-

The creation of reliable word embedding methods has led to a notable expansion in the field of natural language processing (NLP). By converting text into vector representations that reflect the syntactic and semantic characteristics of words, these methods help machines interpret language more accurately and efficiently[2].

At first, text representations were based on frequency-based techniques like term frequency-inverse document frequency (TF-IDF) and sparse techniques like one-hot encoding. Despite their ease of use, these methods suffered from high dimensionality and a lack of semantic comprehension[3,4].

By embedding words into continuous vector spaces, neural network-based models like Word2Vec (Mikolov et al., 2013) signalled a paradigm change and opened the door to richer semantic representation. This foundation was built upon by later models such as GloVe (Pennington et al., 2014) and FastText, which produced embeddings that were more effective and significant.

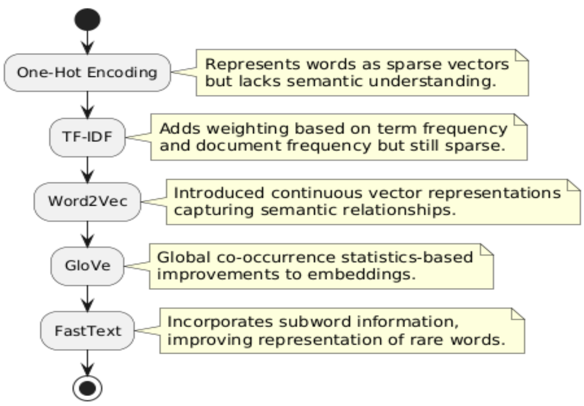

Fig-1: A UML activity diagram showing evolution of text representation techniques.

There are significant drawbacks to static embeddings produced by context-independent models. For example, depending on the context, the word "bank" can refer to either a riverbank or a financial organisation. Models like Word2Vec and GloVe, which give a word a single vector regardless of its usage, are unable to capture these subtleties[5].

This restriction was addressed by context-dependent models, such BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018), which dynamically generated embeddings based on the surrounding text. In order to capture rich contextual connections, these models use topologies such as transformers and recurrent neural networks (RNNs), which incorporate bidirectional information [6,7].

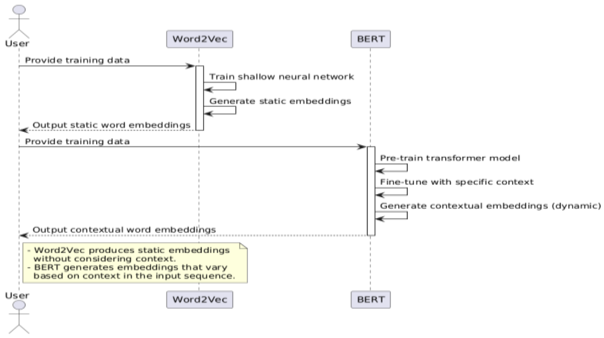

Fig-2: Sequence diagram illustrating the process of generating static embeddings with Word2Vec vs dynamic embeddings with BERT.

The development of word embedding methods has been greatly aided by machine learning, especially deep learning. While context-dependent models use complex deep learning architectures, such as attention mechanisms and transformers, to capture links within sequences, context-independent models usually use shallow neural networks.

3. Context-Independent Word Embedding Models: -

Regardless of a word's context within a phrase, context-independent word embedding models portray each word as a fixed vector. Because of their effectiveness and simplicity, these models have served as a foundation for natural language processing (NLP). They are unable to capture the changing meanings of words in many situations, nevertheless, due to their static character.

1. Word2Vec: -

Word2Vec, first presented by Mikolov et al. in 2013, creates dense vector representations of words using a shallow neural network. Two architectures are available:

- Skip-Gram Model: Given a target word, the skip-gram model predicts the words that surround it.
- CBOW (Continuous Bag of Words):- The Continuous Bag of Words, or CBOW, makes predictions about the target word by looking at the words that surround it.

The key innovation of Word2Vec lies in its ability to capture semantic relationships, such as similarity (e.g., "king" and "queen") and analogies (e.g., "king - man + woman = queen").

Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 3s (2025)

2. GloVe: -

By integrating global co-occurrence information from the corpus, GloVe (Global Vectors for Word Representation), created by Pennington et al. in 2014, enhances Word2Vec. GloVe models the relationship between word pairs based on their co-occurrence probability rather than just local contexts.

GloVe's advantages over Word2Vec include its robustness in smaller datasets and its ability to efficiently capture both syntactic and semantic information.

3. FastText: -

The Word2Vec architecture is improved by FastText, created by Facebook AI Research, which represents words as bags of character n-grams. Because of its ability to collect subword information, FastText is especially useful for out-of-vocabulary terms, unusual words, and languages with complex morphology.

Results: -

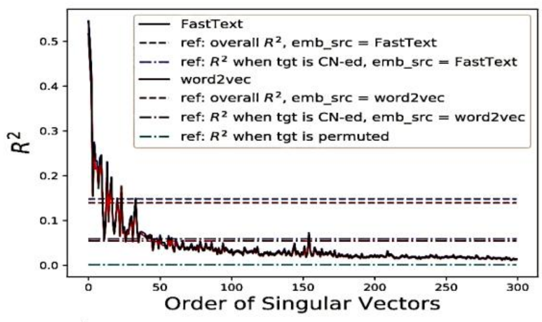

Fig-3: R^2 values of singular vectors from non-distributional word vectors fitted by distributional vectors, plotted by singular value order; GloVe results align closely with word2vec and are omitted for clarity.

Key Features of Context-Independent Models: -

- **Static Embeddings**: Regardless of context, every word has a unique vector representation.
- **High Efficiency**: In comparison to context-dependent models, these models are computationally efficient and require less training time.
- **Limitations**: Incapable of managing changing word meanings and polysemy.

Applications: -

- Retrieval of information and search engines.
- Sentiment analysis of texts that are rather easy.
- Jobs involving document classification where contextual details are not as important.

4. Context-Dependent Word Embedding Models: -

Word representations produced by context-dependent word embedding models are dynamic and vary according to the words that surround them in a particular context. These models are especially helpful for comprehending the subtleties of language that static embeddings miss and for capturing the polysemy of words (words with several meanings). Modern models for a variety of NLP tasks, such as sentiment analysis, question answering, and machine translation, are now built on context-dependent embeddings.

1. **Language Model Embeddings (ELMo):**

One of the first context-dependent models, ELMo was first presented by Peters et al. in 2018. It generates word representations based on the full sentence, improving over static word embeddings. Based on a language model objective, ELMo employs a bidirectional long short-term memory (BiLSTM) network. The context of words in relation to their preceding and following words is captured by the model.

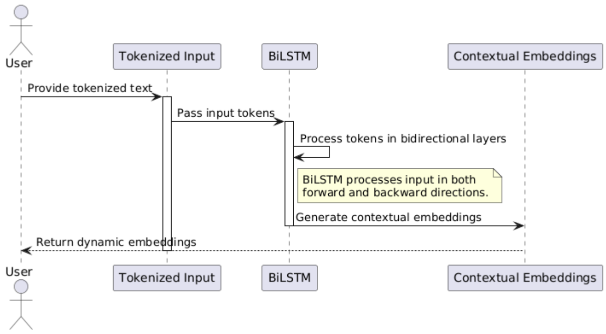

Fig-4: A UML sequence diagram illustrating ELMo's architecture

2. **BERT (Bidirectional Encoder Representations from Transformers)**

Because it makes use of the transformer architecture, BERT, which was suggested by Devlin et al. in 2019, represented a major advancement in NLP. BERT uses a bidirectional attention mechanism, which enables the model to simultaneously capture context from the left and right of a given word, in contrast to ELMo, which uses LSTMs. Two activities are used to pre-train BERT on a sizable corpus:

The Masked Language Model (MLM) predicts the masked words based on context after randomly masking some of the input's words.

The Next Sentence Prediction (NSP) feature trains the model to determine whether a sentence makes sense after another.

Numerous sophisticated NLP models are built on top of BERT, which has been optimised for a range of downstream applications, including named entity recognition, text classification, and reading comprehension.

3. **GPT (Generative Pre-trained Transformer): -**

Another transformer-based paradigm that produces context-dependent embeddings is GPT, created by OpenAI. GPT is autoregressive and analyses text from left to right, in contrast to BERT, which is primarily intended for bidirectional context understanding. Text generating tasks like language modelling and dialogue systems benefit greatly from this unidirectional processing.

Important GPT Features:

- One word at a time, the Autoregressive Model creates text and uses the words it has already created to forecast the subsequent word in the sequence.
- Transformer Architecture: Although it only makes use of the decoder portion of the transformer, GPT, like BERT, depends on the transformer model to capture word associations.
- Pre-training for Generation: GPT has been pre-trained on extensive text corpora and optimised for particular uses, including text generation, chatbots, and summarisation.

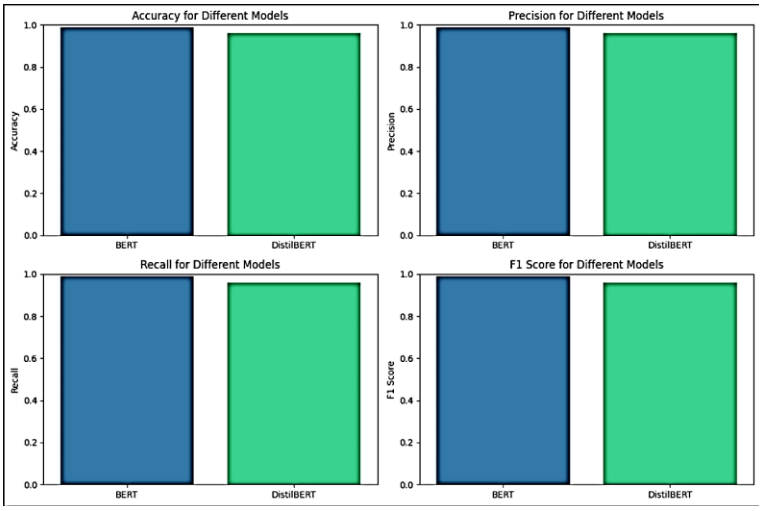

Fig-5: The above chart presents combined results of BERT & DistilBERT (Alternative of ELMo) Context-Dependent word embedding models which depicts accuracy, precision, recall & F1 score

Applications: -

- BERT: Sentiment analysis, text categorisation, question answering, and named entity recognition.
- ELMo: Text summarisation, sentiment analysis, and named entity recognition.
- GPT: Language modelling, dialogue systems, and text generation.

5. Comparative Analysis: Context-Independent vs. Context-Dependent Word Embedding Models

Feature	Context-Independent Models (e.g. Word2Vec, GloVe, FastText)	Context-Dependent Models (e.g. ELMo, BERT, GPT)
Embedding Type	Fixed, Static vectors for each word	Dynamic Vectors based on context
Contextualization	No context-same representation for a word in all contexts	Contextualized-representation changes based on surrounding words
Capturing Polysemy	Poor-same vector for words with multiple meanings	Excellent-distinct representations for polysemous words based in context
Architecture	Shallow neural networks or matrix factorization	Deep learning (e.g. LSTMs, transformers)
Model Complexity	Simple, computationally efficient	Complex, computationally expensive
Training Data	Can be trained on smaller datasets	Requires massive datasets and pertaining on large corpora
Memory and Storage	Low memory requirements, compact models	High memory consumption, large model sizes

6. Conclusion: -

The way we represent and comprehend natural language in computational problems has been completely transformed by word embedding models. Significant progress has been made in the study of how machines process, understand, and synthesise human language, starting with the early days of context-independent models like Word2Vec and GloVe and continuing with the emergence of context-dependent models like ELMo, BERT, and GPT. Despite being easier to use and more computationally efficient, context-independent models have the drawback of not being able to represent how dynamic language is. They give words static representations, which can cause problems like the incapacity to deal with polysemy (words that have several meanings). They are still useful tools in many applications, though, where context is less important.

However, by dynamically creating embeddings depending on surrounding words, context-dependent models have revolutionised Natural Language Processing (NLP) by providing more accurate representations and improved context understanding. These models have demonstrated exceptional performance in a variety of tasks where context is crucial, including named entity identification, sentiment analysis, and machine translation. However, these models' intricacy and resource needs pose difficulties, particularly in settings with limited resources. In the end, the particular task, the computational resources at hand, and the requirement for contextual detail will determine whether to use context-independent or context-dependent models. Context-dependent models such as BERT and GPT are increasingly regarded as the norm for cutting-edge NLP applications, even though context-independent models are still useful for easier, less resource-intensive tasks. It appears that even more advanced hybrid techniques that combine the advantages of both static and dynamic representations are the direction that word embedding models are headed. Context-dependent models are probably going to become more common as computing power increases, pushing the limits of what robots can comprehend and produce in human language.

References

[1] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

[2] Introduced Word2Vec and its architectures (Skip-Gram and CBOW).Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543.

[3] Introduced the GloVe model, emphasizing co-occurrence statistics.Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135–146.

[4] Discussed FastText, emphasizing subword information.Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. arXiv preprint arXiv:1802.05365.

[5] Proposed ELMo for generating context-sensitive embeddings.Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

[6] Introduced BERT, a game-changer for many NLP tasks.Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-training. OpenAI.

[7] Kokane, C. D., & Sachin, D. (2021). Babar, and Parikshit N. Mahalle." Word Sense Disambiguation for Large Documents Using Neural Network Model.". In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE.

[8] Chandra, T., Kaur, M., Rakesh, N., Gulhane, M., & Maurya, S. (2024). Novel blockchain-based framework to publish, verify, and store digital academic credentials of universities. International Journal of Information Technology, 1-9.

[9] Gulhane, M., & Kumar, S. (2024). Fusion of Various Machine Learning Algorithms for Early Heart Attack Prediction. Optimized Predictive Models in Healthcare Using Machine Learning, 229-244.
[10] Nemade, Bhushankumar, and Deven Shah. "An IoT-Based Efficient Water Quality Prediction System for Aquaponics Farming." Computational Intelligence: Select Proceedings of InCITe 2022. Singapore: Springer Nature Singapore, 2023. 311-323.
[11] Choudhary, S., Gulhane, M., Kumar, S., Rakesh, N., Maurya, S., & Tandon, C. (2025). Integrating Machine Learning for Personalized Kidney Stone Risk Assessment: A Prospective Validation Using CLDN11 Genetic Data and Clinical Factors. Genomics at the Nexus of AI, Computer Vision, and Machine Learning, 59-85.
[12] Nemade, B., & Shah, D. (2022). An efficient IoT based prediction system for classification of water using novel adaptive incremental learning framework. Journal of King Saud University-Computer and Information Sciences, 34(8), 5121-5131.