INNOMATİCS
RESEARCH LABS

ARTICLE

NLP (Natural Language Processing)

TITLE

Evolution of Language Representation Techniques: A Journey from BoW to GPT

Submitted by:

Payal Kumari

INNOMATICS RESEARCH LABS

Evolution of Language Representation Techniques: A Journey from BoW to GPT

INTRODUCTION:

Language representation techniques have evolved significantly over the years, shifting from simple methods like Bag of Words (BoW) to complex models such as GPT (Generative Pre-trained Transformer). Here's a look at some of these techniques and how they have transformed natural language understanding.

1. **Bag of Words (BoW)**

   - BoW is one of the simplest language representation techniques. It treats text as a collection of words, ignoring grammar and word order. Each unique word becomes a feature, and the text is represented as a vector of word counts or binary indicators. Although effective in some scenarios, BoW has limitations, such as ignoring semantic relationships and creating sparse representations.

2. Term Frequency-Inverse Document Frequency (TF-IDF)

- TF-IDF is an improvement over BoW. It evaluates the importance of a word in a document relative to a collection of documents. Words frequently occurring in a specific document but rare across other documents get higher weights. TF-IDF helps reduce the

impact of common words (like “the” or “is”) but still struggles with semantic understanding.

3. Word2Vec

- Word2Vec, introduced by Google, uses neural networks to learn word embeddings. These embeddings are dense vectors that capture semantic relationships between words. Words that are contextually similar have similar embeddings. Word2Vec is trained using techniques like Continuous Bag of Words (CBOW) and Skip-Gram, which learn to predict a word from its context or vice versa.

4. GloVe (Global Vectors for Word Representation)

- GloVe is another word embedding model, developed by Stanford, that captures word meanings based on global word-word co-occurrence statistics. Unlike Word2Vec, GloVe uses a co-occurrence matrix to learn embeddings, making it efficient in capturing semantic meanings.

5. Contextual Word Embeddings: BERT (Bidirectional Encoder Representations from Transformers)

- BERT, developed by Google, uses the Transformer architecture to create contextual word embeddings. It understands words based on their context in a sentence, considering both left and right surroundings. BERT models are pre-trained using a masked language model and fine-tuned for specific NLP tasks, significantly improving the understanding of polysemy and complex language nuances.

6. GPT (Generative Pre-trained Transformer)

- GPT, created by OpenAI, is an autoregressive model that leverages the Transformer architecture for language generation. Unlike BERT, which focuses on bidirectional context, GPT is trained to predict the next word in a sequence (unidirectional context). The model is pre-trained on a large corpus and fine-tuned for specific tasks, achieving state-of-the-art results in natural language generation.

1. What is a Language Representation and Vectorisation Technique?

- ***Language Representation and Vectorization Techniques***

Language representation and vectorization techniques convert words or sentences into numerical vectors so that they can be understood and processed by machine learning algorithms. These techniques are fundamental for tasks in Natural Language Processing (NLP), enabling models to perform text classification, sentiment analysis, machine translation, and more.

Key Techniques and Descriptions

1. Bag of Words (BoW)

Description: BoW is one of the simplest techniques to represent text. It treats text as a collection of words and creates a vocabulary from all unique words in the corpus. Each document is then represented as a vector based on the frequency of words appearing in it, without considering the order of words or grammar.
- Use Cases: Text classification, information retrieval, and document similarity.

2. Term Frequency-Inverse Document Frequency (TF-IDF)

Description: TF-IDF is an improvement over BoW that measures the importance of a word in a document relative to a collection of documents. It helps reduce the weight of common words and emphasizes rarer words. The term frequency (TF) calculates how often a word appears in a document, while inverse document frequency (IDF) reduces the weight of frequently occurring words across all documents.
- Use Cases: Document ranking, keyword extraction, and search engine optimization.

3. Word2Vec

Description: Word2Vec is a neural network-based model that creates dense word embeddings. It captures semantic similarities between words by learning word vectors that are close in the vector space if the words appear in similar contexts. It has two main training approaches: Continuous Bag of Words (CBOW) and Skip-Gram.

- **Use Cases:** Sentiment analysis, named entity recognition, and question-answering systems.

4. GloVe (Global Vectors for Word Representation)

Description: GloVe is another technique to generate word embeddings using a global word co-occurrence matrix. It captures the statistical information of how often words co-occur across the entire corpus, resulting in word vectors that encode semantic meanings.

- **Use Cases:** Word similarity analysis, recommendation systems, and language modelling.

5. Contextual Word Embeddings: BERT (Bidirectional Encoder Representations from Transformers)

Description: BERT is a transformer-based model that creates contextual embeddings. It takes into account the context on both the left and right sides of a word, making it ideal for understanding the meaning of words in various contexts. BERT is trained using masked language modelling and next-sentence prediction.

- **Use Cases:** Question answering, text classification, and natural language inference.

6. GPT (Generative Pre-trained Transformer)

Description: GPT is an autoregressive transformer model used primarily for generating text. It predicts the next word in a sequence using only the left context. GPT models are pre-trained on large datasets and can generate coherent and contextually relevant text. They are widely used in applications like chatbots, content generation, and summarization.

- **Use Cases:** Text generation, dialogue systems, and language translation.

2. Different Types of Language Representations.

Language representations are techniques used to transform text into formats that can be processed by machine learning models. They range from simple, rule-based methods to

complex, deep learning-based models. Here are the main types of language representations:

1. One-Hot Encoding

Description: One-hot encoding is a basic method where each word in the vocabulary is represented as a vector with all zeros except for a single one at the index corresponding to that word. While simple, this technique results in high-dimensional, sparse vectors and does not capture semantic relationships between words.

- Use Cases: Suitable for small vocabularies in simple text classification problems.

2. Count Vectorization (Bag of Words)

Description: Count vectorization, similar to Bag of Words, creates a vector representation based on the frequency of words in a document. It's useful for capturing word occurrence but ignores word order and context.

- Use Cases: Document classification and clustering.

3. TF-IDF Vectorization

Description: TF-IDF vectorization assigns weights to words based on their frequency in a document relative to the entire corpus. This method helps emphasize rare but important words while reducing the weight of common words.

- Use Cases: Search engines and text mining.

4. Word Embeddings (Word2Vec, GloVe)

Description: Word embeddings represent words as dense vectors in a lower-dimensional space. Models like Word2Vec and GloVe learn these vectors from large corpora, capturing semantic relationships, such as similar words appearing closer in the vector space.

- Use Cases: Sentiment analysis, word similarity tasks, and question-answering systems.

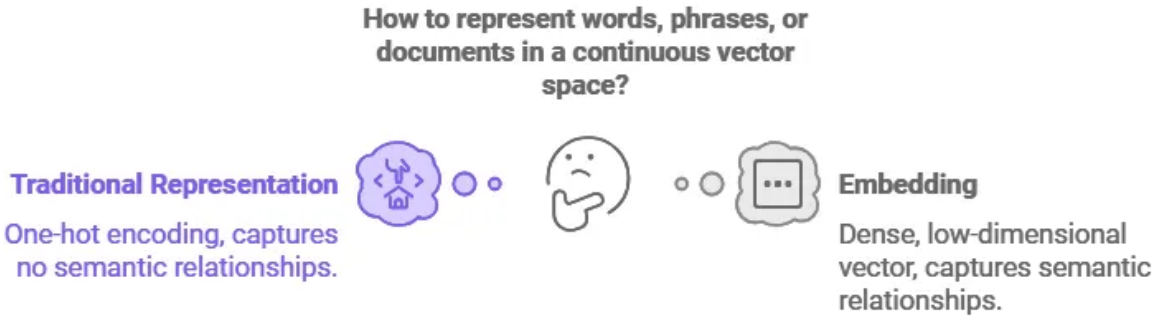
5. **Contextual Embeddings (BERT, GPT)**

Description: Contextual embeddings are generated using deep learning models like BERT and GPT. These models understand the context of a word within a sentence, resulting in different representations for the same word in different contexts. They have revolutionized NLP tasks by capturing polysemy and complex language structures.

- Use Cases: Machine translation, text summarization, and conversational AI.

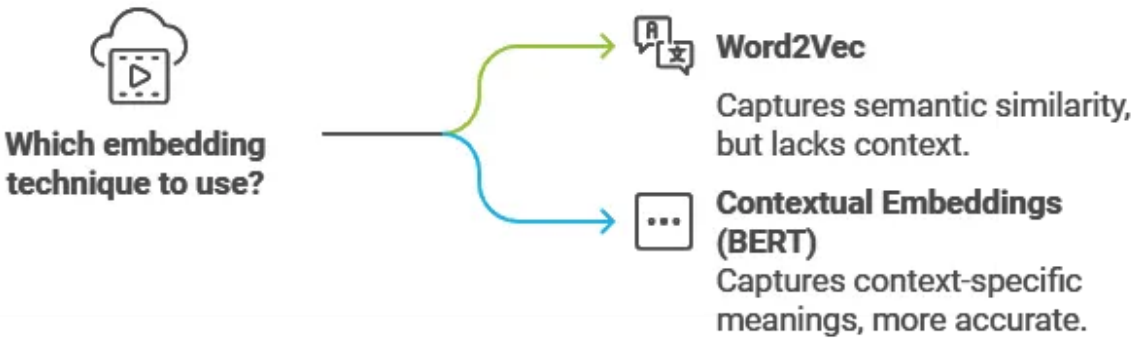
3. What is an Embedding?

Definition: An embedding is a representation of words, phrases, or even entire sentences as continuous-valued vectors in a fixed-dimensional space. Embeddings capture the semantic meaning of words or phrases, enabling machine learning models to understand and process natural language effectively.



Embeddings are typically learned using neural network-based models that process a large corpus of text and map words into a vector space where relationships are encoded. For example:

- **Word2Vec:** Embeddings capture semantic similarity (e.g., “king”—“man” + “woman” ≈ “queen”).
- **Contextual Embeddings (like BERT):** Embeddings differ for each occurrence of a word depending on its surrounding context.



Key Concepts of Embeddings;

1. **Dense Vectors:** Unlike traditional sparse representations (like one-hot encoding), embeddings are dense vectors that contain fewer dimensions and non-zero values. Each dimension captures specific features or relationships, making the representation more efficient and meaningful.

2. Semantic Similarity: Words that are semantically similar or often appear in similar contexts are placed closer together in the vector space. For example, the words "king" and "queen" or "dog" and "puppy" will have similar embeddings.

3. Learning from Data: Embeddings are typically learned from large text corpora using models like Word2Vec, GloVe, or transformer-based models like BERT. These models analyze the context of words to generate vectors that capture their meanings.

4. Applications: Embeddings are widely used in NLP tasks such as text classification, sentiment analysis, machine translation, and more. They help improve the performance of models by providing richer representations of language.

4. Difference between Word2Vec, BERT and GPT approaches.

1. Word2Vec

Description: Word2Vec is a shallow, two-layer neural network that produces word embeddings. It comes in two models: Continuous Bag of Words (CBOW) and Skip-Gram. Word2Vec focuses on predicting a word given its context (CBOW) or predicting the context given a word (Skip-Gram), capturing semantic relationships in dense vectors.

- **Strengths:** Simple and efficient for creating meaningful word embeddings. It captures semantic similarities between words.
- **Limitations:** It generates static embeddings, meaning the vector representation of a word does not change based on context.

2. BERT (Bidirectional Encoder Representations from Transformers)

Description: BERT is a transformer-based model that generates contextual embeddings. It processes text bidirectionally, meaning it considers the context on both sides of each word. BERT is pre-trained using tasks like masked language modeling and next-sentence prediction, which allow it to capture the meaning of words based on their context.

- **Strengths:** Generates different embeddings for the same word depending on its context. Highly effective in understanding sentence structure and polysemous words.
- **Limitations:** High computational cost and complexity.

3. GPT (Generative Pre-trained Transformer)

Description: GPT is a unidirectional model that uses the transformer architecture to generate text. It predicts the next word in a sequence by processing text from left to right. GPT is pre-trained on a large corpus and fine-tuned for specific tasks, making it powerful for text generation.

- **Strengths:** Excellent at generating coherent and contextually relevant text. Efficient in autoregressive tasks.

- Limitations: Less effective than BERT in understanding the full context, as it only looks at previous words in a sequence.

CONCLUSION:

- Language representation and vectorization techniques have evolved significantly, transforming the way machines process and understand natural language.
- Starting with simple models like One-Hot Encoding and Bag of Words (BoW), which represent words without any understanding of context or meaning, these techniques have laid the foundation for more sophisticated methods.
- TF-IDF improved on BoW by introducing a weighting mechanism to prioritize important words, but it still struggled with semantic relationships.
- The advent of word embeddings like Word2Vec and GloVe revolutionized NLP by providing dense vectors that capture semantic similarities, such as grouping similar words together in the vector space.
- The development of contextual embeddings through models like BERT and GPT marked a significant advancement.
- BERT's bidirectional processing allows it to generate context-aware embeddings, making it powerful for tasks requiring an understanding of word meaning in context.
- On the other hand, GPT excels in generating coherent text by leveraging its unidirectional architecture, processing input sequentially to predict the next word.

Overall, these techniques highlight the continuous progress in natural language understanding, from simple frequency-based models to complex transformer-based architectures.

- Each method has its strengths and limitations, and the choice of model often depends on the specific NLP task and its requirements.
- As research continues, future advancements will likely push the boundaries of what is possible in language representation and understanding, bringing even more sophisticated models that mimic human-like language comprehension and generation.

REFRENCES:

1. Pennington, J., Socher, R., & Manning, C. D. (2014).
"GloVe: Global Vectors for Word Representation." Conference on Empirical Methods in Natural Language Processing (EMNLP).
- This paper introduces GloVe, a word embedding model that uses global word-word co-occurrence statistics to learn dense representations of words.
Link to Paper:

2. Jurafsky, D., & Martin, J. H. (2021).
Speech and Language Processing (3rd ed.). Pearson.
- A comprehensive textbook on natural language processing, which covers various language representation techniques, including embeddings, TF-IDF, and neural network models like Word2Vec, BERT, and GPT.
Link to Book Website:

3. Goldberg, Y. (2016).
"A Primer on Neural Network Models for Natural Language Processing." Journal of Artificial Intelligence Research (JAIR)*, 57, 345–420.
This article provides an overview of neural network models for NLP, discussing embeddings, RNNs, and the development of deep learning approaches for language understanding.
Link to Paper:

4. Le, Q., & Mikolov, T. (2014).

"Distributed Representations of Sentences and Documents." International Conference on Machine Learning (ICML).

This paper extends the concept of Word2Vec to sentences and documents, introducing the Paragraph Vector model to generate embeddings for larger text units.

Link to Paper:

5. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018).

This paper introduces ELMo (Embeddings from Language Models), an approach that generates deep contextualized word representations using bidirectional language models.

Link to Paper:

Appreciation

I would like to extend my heartfelt gratitude to **Innomatics Research Labs** and **Kanav Bansal** for their unwavering support and for providing me with the incredible opportunity to expand my knowledge and explore emerging technologies. Their guidance and encouragement have been invaluable in this learning journey.

——————————————————————Thank You——————————————————

Thank you so much for reading my article! I hope it has offered you valuable insights and sparked curiosity about the exciting world of technology. Your support is deeply appreciated, and I'm thrilled to have you as part of this journey. If you have any thoughts, questions, or feedback, please don't hesitate to share—I'd love to hear from you!

Thank you once again, and happy learning!