Evaluating Semantic Similarity Search: A Methodological Approach to Comparing Language Models and Algorithms

**Abstract**

This research paper introduces a novel methodology for the comparative analysis of various Language Models (LMs) and semantic similarity algorithms in the context of information retrieval. The study aims to create a standard way of evaluating and comparing models on their semantic similarity search capabilities. The LMs used include Word2Vec, MiniLM, and RoBERTa, evaluated with similarity metrics such as cosine similarity, Euclidean similarity, and dot product similarity. The methodology also considers different storage methods, specifically local storage and Pinecone Index, and employs the precision@k methodology with k set to 3. The results are analyzed using a comprehensive scoring formula, considering time taken for retrieval, precision score, and similarity metric score. The findings contribute to the broader field of Natural Language Processing and conversational AI, offering insights into the optimal combinations for efficient and accurate information retrieval.

**1. Introduction**

The vast expansion of unstructured data and the growing need for meaningful interactions between humans and machines have necessitated more refined techniques in information retrieval and natural language processing. In the realm of Natural Language Processing (NLP), the ability to accurately retrieve information based on semantic similarity is a critical task. This involves identifying and retrieving sentences from a text corpus that are semantically similar to a given query sentence. The effectiveness of this process is determined by the choice of Language Models (LMs) and semantic similarity algorithms used.

This research paper embarks on a comprehensive comparative study of various LMs and semantic similarity algorithms to identify a standard approach to evaluate combinations for semantic similarity search. The LMs under consideration include Word2Vec, MiniLM, and RoBERTa, which are among the most advanced models currently used in NLP. These models are evaluated in combination with different similarity metrics, namely cosine similarity, Euclidean similarity, and dot product similarity. The study also explores the impact of different storage methods on the performance of these combinations. Specifically, it compares the use of local storage and Pinecone Index, a vector database designed for efficient information retrieval. The performance of each combination is evaluated using the precision@k methodology, with k set to 3. This means that the top 3 sentences retrieved by each combination are used to compute the precision@3 score, which forms the basis for ranking the combinations using the comprehensive scores.

A unique aspect of this study is the incorporation of the similarity metric score along with the precison@k score into the final comprehensive score. This is achieved through a formula that takes into account the similarity metric score only if the retrieved sentence is semantically similar to the query sentence. The ultimate goal of this study is to provide a comprehensive evaluation strategy towards understanding the performance of different combinations of LMs and similarity algorithms in semantic similarity search. The findings of this study will contribute to the broader field of NLP and conversational AI, providing valuable insights for researchers and practitioners alike.

By focusing on the methodology and the comparative aspect of different models and algorithms, this research paper aims to fill a gap in existing research, offering a robust framework that can be applied to various models and domains. The methodology's comprehensive nature ensures a nuanced and standardized approach, paving the way for further research and potential advancements in the field of semantic search evaluation and comparison.

**2. Literature Review**

The field of Natural Language Processing (NLP) has seen significant advancements in recent years, particularly in the area of semantic similarity. Semantic similarity is a measure of the degree to which two pieces of text carry the same meaning. This concept is critical in various NLP tasks such as information retrieval, text classification, question answering, and plagiarism detection.

**2.1 Evolution of Semantic Similarity -- A Survey [1]:** This paper provides a comprehensive overview of the concept of semantic similarity and its evolution over time. It discusses various methods and techniques used to measure semantic similarity, including knowledge-based methods, corpus-based methods, and hybrid methods. The paper also highlights the challenges and open research problems in estimating semantic similarity in NLP. However, unlike our study, they do not delve into the practical application of these methods in a real-world scenario. Our research aims to fill this gap by applying these methods to a specific dataset and evaluating their performance.

**2.2 A Survey on Semantic Similarity [2]:** This paper presents a detailed survey of semantic similarity of text documents in NLP. It discusses the importance of semantic similarity in various NLP tasks and provides an overview of the different methods used to measure it. The paper also discusses the applications of semantic similarity in fields such as information retrieval, text classification, question answering, and plagiarism detection, but it does not focus on the comparison of different language models and their performance in semantic similarity tasks.

**2.3 COVID-19 information retrieval with deep-learning based semantic search [3]:** This paper presents a semantic, multi-stage, search engine designed to handle complex queries over the COVID-19 literature. The authors use deep learning techniques to extract semantic information from the text and use it to improve the accuracy of information retrieval. The paper provides valuable insights into the application of semantic similarity in a specific domain.

**2.4 Semantic Similarity Computing Model Based on Multi Model Fine-Grained [4]:** This paper discusses the challenges of extracting text in a fine-grained way in NLP. The authors propose a combination of methods, including word embedding, semantic role labeling, and semantic similarity computing, to overcome these challenges. The paper provides a detailed analysis of the proposed methods and their effectiveness in improving the accuracy of semantic similarity computation. However, they do not evaluate the performance of these methods in terms of precision and time efficiency, which is a key focus of our study.

**2.5 Sentence Similarity Based on Contexts [5]:** This paper discusses the task of quantitatively measuring the semantic relatedness between two sentences. The authors propose a method that takes into account the context of the sentences to improve the accuracy of semantic similarity measurement. The paper provides a detailed analysis of the proposed method and its effectiveness in various NLP tasks.

**2.6 Semantic Textual Similarity Methods, Tools, and Applications [6]:** This paper provides a comprehensive overview of Semantic Textual Similarity (STS), a metric used to measure the semantic similarity between text documents. The authors discuss various methods and tools used to compute STS and provide an overview of its applications in different fields.

**2.7 Semantic Similarity - A Survey [7]:** This paper provides a collection of research papers on the task of Semantic Similarity in NLP. It provides a comprehensive overview of the different methods used to measure semantic similarity and the challenges associated with each method.

**2.8 Semantic Cosine Similarity [8]:** This paper proposes an enhancement of cosine similarity measurement by incorporating semantic checking between dimensions of two term vectors. The authors provide a detailed analysis of the proposed method and its effectiveness in improving the accuracy of semantic similarity measurement.

**2.9 Semantic Search based on the Online Integration of NLP Techniques [9]:** This paper introduces a framework for semantic information retrieval based on the integration of various NLP techniques. The authors discuss the challenges of semantic search and propose a method that combines different NLP techniques to overcome these challenges. This paper provides a valuable perspective on semantic search, but it does not evaluate their performance.

While the existing literature provides valuable insights into the concept of semantic similarity and its applications, there is a gap in the evaluation and comparison of different language models and similarity metrics. Our research aims to fill this gap by providing a comprehensive comparison of different language models and similarity metrics in terms of their precision and time efficiency in semantic similarity tasks.

**3. Methodology**

This section provides a detailed explanation of the methodology employed in this research. The research process involved the creation of a standard evaluation process, use of three different language models ranging from a small sized and baseline model to a large sized and complex language model, three similarity metrics, and two storage methods. The methodology also includes the preparation of the corpus and query sentences, the execution of the experiment, and the evaluation of the results.

**3.1 Similarity Metrics, and Storage Methods**

Three different similarity metrics were used to measure the semantic similarity between the query sentences and the sentences in the corpus: cosine similarity, Euclidean distance, and dot product. These metrics were chosen because they are commonly used in information retrieval and have been shown to be effective in measuring semantic similarity.

Two different storage methods were used: Pinecone, a vector database service, and local storage. The use of Pinecone allowed for efficient information retrieval, while local storage served as a baseline for comparison.

**3.2 Model Selection and Application**

We employ various language models to encode our corpus and query sentences into embeddings. These models have been chosen based on our literature review and include the following:

1. Word2Vec: As a baseline, we use the Word2Vec model to analyze the semantic proximity between sentences. Word2Vec's simplicity and fundamental approach provide a necessary comparison for the more complex transformer models.
2. MiniLM (all-MiniLM-L6-v2): Selected for its size efficiency and competitive performance, MiniLM provides a balance between resource usage and semantic understanding.
3. RoBERTa (all-roberta-large-v1): Recognized for its robust performance, the RoBERTa model is utilized as the biggest and most complex of the models in selection

Three different language models were used in this research: Word2Vec, MiniLM, and RoBERTa. These models were selected based on their sizes and complexity, with RoBERTa being the largest and most complex, followed by MiniLM, and then Word2Vec. Each of these models has been proven effective in various natural language processing tasks, and their use in this research allowed for a comprehensive comparison of their performance in semantic similarity search.

**3.3 Corpus Preparation and Query Generation**

The corpus used in this research was a text file containing sentences about Barack Obama. The corpus was split into individual sentences, and each sentence was tokenized using the Spacy tokenizer.

The query sentences were generated by slightly modifying certain sentences from the corpus. These sentences were chosen such that they had a high semantic similarity with at least one sentence in the corpus. The query sentences were also tokenized using the Spacy tokenizer.

The original sentence from the dataset was: "In 2008, a year after beginning his campaign, and after a close primary campaign against Hillary Clinton, he was nominated by the Democratic Party for president." The input text to the models was: "In the year 2008, after a tightly contested primary against Hillary Clinton and following the start of his political journey a year earlier, the Democratic Party nominated him for the presidency."

**3.4 Experiment Process**

For each combination of model, similarity metric, and storage method, the following steps were performed:

1. The model was used to generate embeddings for the sentences in the corpus and the query sentences.
2. If Pinecone was used as the storage method, the corpus embeddings were added to a Pinecone index.
3. The similarity between each query sentence and the sentences in the corpus was calculated using the chosen similarity metric.
4. The top 3 most similar sentences for each query were retrieved, and their similarity scores were recorded.

The time taken to retrieve the most similar sentences was also recorded for each combination, but it was not included in the final ranking score which is the comprehensive similarity score..

**3.5 Evaluation**

The performance of each combination of model, similarity metric, and storage method was evaluated using the precision at k ranking methodology, with k set to 3. This means that the top 3 most similar sentences retrieved for each query were used to calculate the precision score.

A matched sentence was considered semantically similar if its similarity score was greater than 0.5 and if it was in fact semantically similar after human verification, The matched would then be given a precision score by dividing the number of truly semantically similar sentences by k ( 0.333 per correctly matched sentence). Otherwise, if it was considered not semantically similar, it will be assigned a precision score of 0. This implies that a model exhibiting a precision score of 0.999 has demonstrated optimal performance, a score of 0.666 indicates moderate performance, a score of 0.333 suggests marginal performance, and a score of 0 signifies poor performance.

The final ranking score for each combination was calculated using the novel formula:

***Comprehensive score = (The sum of correct matched sentence similarity scores \* precision@k ) / k***

This score, referred to as the comprehensive score, factored in the semantic similarity scores, precision scores, and the value of k. It provided a holistic measure of the performance of each combination, taking into account both the quality of the matched sentences (as indicated by the semantic similarity scores) and the quantity of correct matches (as indicated by the precision scores).

**3.6 Results Comparison**

The results of the experiment were compared and analyzed based on the comprehensive scores of the different combinations. These scores provided a measure of how well each combination performed in terms of semantic similarity search, taking into account both the quality and quantity of the matched sentences.

The results comparison allowed for a comprehensive evaluation of the performance of the different combinations of models, storage methods, and similarity metrics. This evaluation provided valuable insights into the effectiveness of these combinations in semantic similarity search, and it formed the basis for the conclusions drawn in this research**.**

**4. Results and Analysis**

The experiment was conducted using three different language models (Word2Vec, MiniLM, and RoBERTa), three similarity metrics (Cosine Similarity, Euclidean Similarity, and Dot Product Similarity), and two storage methods (Local and Pinecone Index). The results were evaluated using a newly introduced comprehensive score which uses similarity scores and the precision at k ranking methodology, with k set to 3. This means that the top 3 matched sentences by the combination of model, storage method, and similarity metric were used to create the precision@3 score which was used in ranking them.

The results of the experiment are presented below in a series of tables and graphs. These results provide a comprehensive overview of the performance of each combination of model, similarity metric, and storage method.

**4.1 Comprehensive Scores of Each Model and Each Similarity Metric Under Pinecone Storage Method**

The bar graph below shows the comprehensive scores of each model and each similarity metric under the Pinecone storage method. The comprehensive score is calculated using the formula:

***Comprehensive score = (The sum of correct matched sentence similarity scores \* precision@k ) / k***

This score takes into account the semantic similarity scores, precision scores, and the value of k, providing a holistic measure of the performance of each combination.

## 

Figure 1

**4.2 Comprehensive Scores of Each Model and Each Similarity Metric Under Local Storage Method**

The bar graph below shows the comprehensive scores of each model and each similarity metric under the local storage method. As with the Pinecone storage method, the comprehensive score is calculated using the same formula.

## 

Figure 2

**4.3 Comparison of Average Time Taken by the Three Models Across All Three Similarity Metrics Under Pinecone Storage vs Local Storage**

The graph below compares the average time taken by the three models across all three similarity metrics under Pinecone storage vs local storage. This comparison provides an insight into the efficiency of the two storage methods in terms of retrieval speed.

## 

Figure 3

Table 1: Word2Vec Model Performance

| Storage Method | Similarity Metric | Time  ( Secs ) | Result |
| --- | --- | --- | --- |
| Pinecone Index | Cosine | 0.2943 | 0.1098 |
| Pinecone Index | Euclidean | 0.3074 | 0 |
| Pinecone Index | Dot Product | 0.2673 | 0 |
| Local Storage | Cosine | 0.0121 | 0.1098 |
| Local Storage | Euclidean | 0.0054 | 0 |
| Local Storage | Dot  Product | 0.0011 | 0 |

Average time -

1. Pinecone index - 0.2896
2. Local Storage - 0.0062

Table 2: MiniLM Model Performance

| Storage Method | Similarity Metric | Time  ( Secs ) | Result |
| --- | --- | --- | --- |
| Pinecone Index | Cosine | 0.2713 | 0.6972 |
| Pinecone Index | Euclidean | 0.8683 | 0.3451 |
| Pinecone Index | Dot Product | 0.3004 | 0.6972 |
| Local Storage | Cosine | 0.0028 | 0.6972 |
| Local Storage | Euclidean | 0.0054 | 0 |
| Local Storage | Dot Product | 0.0036 | 0.6972 |

Average time -

1. Pinecone index - 0.4800
2. Local Storage - 0.0040

Table 3: RoBERTa Model Performance

| Storage Method | Similarity Metric | Time  ( Secs ) | Result |
| --- | --- | --- | --- |
| Pinecone Index | Cosine | 0.0315 | 0.7449 |
| Pinecone Index | Euclidean | 0.0134 | 0.3234 |
| Pinecone Index | Dot Product | 0.0139 | 0.7449 |
| Local Storage | Cosine | 0.0057 | 0.6479 |
| Local Storage | Euclidean | 0.0113 | 0 |
| Local Storage | Dot Product | 0.0028 | 0.6479 |

Average time -

1. Pinecone index - 0.0196
2. Local Storage - 0

**4.4 Detailed Results**

The detailed results of the experiment are presented in the tables below. These tables provide a comprehensive overview of the performance of each combination of model, similarity metric, and storage method, including the matched sentences, their similarity scores, and the time taken to evaluate the matching.

**4.5 Ranking of Combinations**

Based on the final comprehensive score for each combination of model, storage method, and similarity metric, the ranking table would be as follows:

| Rank | Model | Storage Method | Similarity Metric | Comprehensive Score |
| --- | --- | --- | --- | --- |
| 1 | RoBERTa | Pinecone Index | Cosine | 0.7449 |
| 2 | RoBERTa | Pinecone Index | Dot Product | 0.7449 |
| 3 | MiniLM | Pinecone Index | Cosine | 0.6972 |
| 4 | MiniLM | Pinecone Index | Dot Product | 0.6972 |
| 5 | MiniLM | Local Storage | Cosine | 0.6972 |
| 6 | MiniLM | Local Storage | Dot Product | 0.6972 |
| 7 | RoBERTa | Local Storage | Cosine | 0.6479 |
| 8 | RoBERTa | Local Storage | Dot Product | 0.6479 |
| 9 | RoBERTa | Pinecone Index | Euclidean | 0.3234 |
| 10 | MiniLM | Pinecone Index | Euclidean | 0.3451 |
| 11 | Word2Vec | Pinecone Index | Cosine | 0.1098 |
| 12 | Word2Vec | Local Storage | Cosine | 0.1098 |
| 13-  18 | Various | Various | Various | 0 |

**4.6 Analysis**

The results of the experiment show that the RoBERTa model combined with the Pinecone storage method and either the Cosine Similarity or Dot Product Similarity metric achieved the highest comprehensive scores, indicating the best performance in terms of semantic similarity search. The MiniLM model also performed well, especially when combined with the Cosine Similarity or Dot Product Similarity metric and either the Pinecone storage method or local storage.

On the other hand, the Word2Vec model achieved the lowest comprehensive scores, indicating the poorest performance. This suggests that the Word2Vec model may not be the best choice for semantic similarity search, especially when compared to more complex models like RoBERTa and MiniLM.

In terms of storage methods, the Pinecone storage method generally achieved higher comprehensive scores than local storage, especially when combined with the RoBERTa or MiniLM model and either the Cosine Similarity or Dot Product Similarity metric. However, the Pinecone storage method also took longer to evaluate the matching, indicating a trade-off between performance and efficiency.

Overall, these results provide valuable insights into the performance of different combinations of language models, similarity metrics, and storage methods in semantic similarity search. They highlight the importance of choosing the right combination to achieve the best performance, and they provide a basis for further research and optimization in this area**.**

**5. Discussion**

**5.1 Emphasis on Evaluation Methodology**

The core contribution of this research lies in the development of a standard approach towards evaluating language models for semantic similarity searches. The comprehensive evaluation methodology, characterized by the novel formula for calculating the comprehensive score, sets a new benchmark in the field. This formula, which integrates semantic similarity scores, precision scores, and the value of k, offers a holistic measure of performance, reflecting both the quality and quantity of matched sentences.

**5.2 Comparison of Models**

The study's results, derived from the innovative evaluation methodology, reveal significant insights into the performance of different language models, similarity metrics, and storage methods. The findings underscore the importance of selecting the right combination to achieve optimal performance in semantic similarity search.

**5.3 Unexpected Findings and Implications**

The unexpected effectiveness of the Pinecone storage method and the trade-off between performance and efficiency highlight the nuanced complexities of semantic similarity search. These insights, derived from the standard evaluation process, contribute valuable guidance for researchers and practitioners.

**5.4 Comparison with Existing Literature**

The research's focus on creating a standardized evaluation approach both affirms and extends existing knowledge. The innovative methodology enriches the current discourse on semantic similarity search, positioning this study as a significant contribution to the field.

**6. Future Work**

The research opens several promising avenues for future exploration:

1. **Refinement of Evaluation Methodology:** The current evaluation method could be further refined and standardized, potentially becoming a widely accepted benchmark in the field.
2. **Inclusion of Time in Comprehensive Formula:** Future work could explore the integration of time taken for search into the comprehensive formula by assigning it an appropriate weight. This addition would provide a more nuanced understanding of efficiency and performance trade-offs.
3. **Expansion of Models and Metrics:** Investigating additional models, metrics, or storage methods could lead to more comprehensive insights.
4. **Development of Adaptive Frameworks:** Creating frameworks that dynamically select the best combination based on specific requirements may offer more tailored solutions.

**7. Conclusion**

This research represents a significant advancement in the field of semantic similarity search, with its primary focus on creating a standard approach towards evaluating language models. The innovative evaluation methodology, characterized by the comprehensive score formula, sets a new standard in the field, offering a holistic and nuanced understanding of performance.

The findings of the study, the unexpected insights, and the future directions all contribute to a richer understanding of semantic similarity search. The research not only provides valuable guidance for current practitioners but also lays a strong foundation for future innovations, promising continued growth and refinement in this vital area of NLP.

**8. References**

1. Chandrasekaran, D., & Mago, V. (2021). Evolution of Semantic Similarity -- A Survey. ACM Computing Surveys, 54(2), 1-37. [Link](https://doi.org/10.1145/3440755)
2. Sunilkumar P., & Athira P. Shaji. (2019). A Survey on Semantic Similarity. 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 20-21 December 2019. [IEEE](https://doi.org/10.1109/ICAC347590.2019.9036843)
3. Li, Y., Zhang, X., Wang, S., Wang, J., Ma, X., Wang, Y., ... & Zhang, H. (2021). A novel semantic similarity computation model based on the human cognitive process. npj Digital Medicine, 4(1), 1-10. [Nature](https://www.nature.com/articles/s41746-021-00437-0)
4. Zhang, P., Huang, X., Wang, Y., Jiang, C., He, S., & Wang, H. (2022). Semantic Similarity Computing Model Based on Multi Model Fine-Grained Nonlinear Fusion. arXiv preprint arXiv:2202.02476. [arXiv](https://doi.org/10.48550/arXiv.2202.02476)
5. Zhang, Y., & Clark, S. (2021). Sentence Similarity Based on Contexts. Transactions of the Association for Computational Linguistics, 9, 111218. [MIT Press Journals](https://doi.org/10.1162/tacl_a_00477)
6. Hernández-Farias, I., Benedí, J. M., & Rosso, P. (2016). Applying Basic Features from Sentiment Analysis for Automatic Emotion Classification. Computación y Sistemas, 20(4), 647-656. [Scielo](https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462016000400647)
7. Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012). Semantic Cosine Similarity. The 7th International Student Conference on Advanced Science and Technology ICAST 2012, Seoul, South Korea. [Link](https://www.researchgate.net/publication/262525676_Semantic_Cosine_Similarity)
8. <https://www.sciencedirect.com/science/article/pii/S1877042811024360>
9. <https://arxiv.org/pdf/2004.13820>
10. <https://paperswithcode.com/task/semantic-similarity>