# Does Size Matter? Examining Sentence Similarity Performance in Large Language Models

Alex Maximilian Korga
*Department of Data Science, XU Exponential University of Applied Sciences*
Potsdam, Germany
a.korga@student.xu-university.de

Sebastian Wefers
*Department of Data Science, XU Exponential University of Applied Sciences*
Potsdam, Germany
s.wefers@student.xu-university.de

Keno Hanken
*Department of Data Science, XU Exponential University of Applied Sciences*
Potsdam, Germany
k.hanken@student.xu-university.de

Raad Bin Tareaf
*Head of Data Science, XU Exponential University of Applied Sciences*
Potsdam, Germany
r.bintareaf@xu-university.de

Ben Steemers
*Data Scientist*
*Formerly with Joblift GmbH*
Hamburg, Germany
bensteemers@protonmail.com

Hunaida Avvad
*Management Information Systems Department*
*Izmir Bakircay University*
Izmir, Turkey
hunaida.awwad@bakircay.edu.tr

***Abstract*—Sentence embeddings are a crucial component of natural language processing (NLP), enabling machines to capture subtle meaning and context in text. By converting sentences into compact, fixed-size vectors, sentence transformers unlock a range of applications, including semantic search (identifying relevant documents or passages based on their meaning and context), sentence similarity assessment (measuring the degree of semantic equivalence between two sentences), and paraphrase detection (identifying sentences that convey the same meaning using different words or phrasing). In this study, we fine-tuned multiple state-of-the-art sentence transformer models on a custom dataset specifically designed for sentence similarity tasks and evaluated their performance through rigorous benchmarking. Our findings yield a surprising insight: the choice of model is highly task-dependent, with larger models not always outperforming smaller ones, which underscores the importance of model selection for optimal performance in specific NLP tasks.**

***Index Terms*—Information retrieval, Search methods, Semantic search, Natural language processing, Transformer models, Deep learning, Transfer learning**

## I. Introduction

The rapid advancement of machine learning has enabled machines to perform tasks once exclusive to humans, such as understanding language nuances. A key breakthrough is sentence embeddings, which allow machines to capture the meanings and contexts within text, advancing applications like semantic search, sentence similarity, and paraphrase detection by transforming sentences into vectors for comparison [6].

One major application of sentence embeddings is semantic search, which is crucial for information retrieval. By representing queries and documents as vectors, machines are able to capture semantic relationships, which enables more accurate and personalized search results, improving user experience and retrieval effectiveness [4].

Sentence embeddings have also significantly impacted sentence similarity tasks. By comparing vector representations, machines determine semantic similarity, supporting applications like text classification, clustering, and topic modeling with implications in customer service and content management [3]. Paraphrase detection, another key application, identifies sentences with the same meaning but different wording, aiding in plagiarism detection and improving machine translation by capturing linguistic nuances [12].

Recent advances in sentence transformer models have improved sentence embeddings, but selecting the right model for specific tasks remains challenging due to varying model strengths, task specificity, and resource requirements. Despite the importance of sentence similarity tasks, a knowledge gap persists regarding the efficacy of sentence transformer models. In 2022 study by Casola et al. [1] conducted a study that compared five pre-trained transformers. However, this research aims to close that knowledge gap by fine-tuning state-of-the-art models on a custom sentence similarity dataset. By benchmarking these models, we explore the impact of model size and task specificity, emphasizing the need for dataset-specific fine-tuning to optimize performance. Our findings have implications for improving NLP systems and enhancing sentence similarity tasks in applications like information retrieval, text summarization, and question answering.

## II. Literature Review

The first BERT model, called Bidirectional Encoder Representations from Transformers, was created by Devlin et al. [2], revolutionized natural language processing by pretraining deep bidirectional representations using the Transformer architecture. Unlike previous models that processed text sequentially,

BERT captures context from both directions, making it highly effective for various NLP tasks.

Building on the success of BERT, Reimers et al. [10] introduced Sentence-BERT (SBERT), a variant designed to efficiently produce semantically meaningful sentence embeddings. SBERT utilizes siamese and triplet network structures, which enable the generation of fixed-size sentence embeddings that can be compared using cosine similarity. This architecture addresses the major limitation of BERT in sentence similarity tasks. While BERT and RoBERTa perform well on these tasks, they require significant computational resources. SBERT, on the other hand, reduce an 65 hour inference on BERT in just 5 seconds while still preserving the accuracy [10].

Since the introduction of Sentence-Bert, different models and variations for semantic sentence similarity tasks have been developed. Reimers et al. [9] focused on modifying the models developed in [10] for paraphrasing tasks to fit on a multitude of tasks, such as information retrieval and classification. Ni et al. on the other hand worked on modifying text-to-text transformers, more specifically the T5 model family developed by Raffel et al. [8], to fit sentence similarity tasks efficiently in the same way SBERT does.

## III. Methodology & Implementation

In this section, we first introduce the data we used for this benchmark (III-A), followed by the chosen models (III-B), and finally the training setup for this benchmark (III-C).

### A. Data

The dataset utilized for training and fine-tuning the sentence similarity models encompasses a total of 21,930 samples, segmented into 16,930 training samples and 5,000 testing samples. This data represents job postings and related attributes and includes entries in both English and German. The primary application of this data is in job matching and recommendation systems, where the goal is to accurately assess and align job descriptions with potential candidates based on various attributes.

The dataset was sourced from Joblift [5], a meta-search platform for job vacancies with millions of open positions across four main countries. The data includes job descriptions and related attributes that allow for semantic analysis and job matching. This dataset was designed to facilitate the development of models that assess the semantic similarity between job descriptions in order to enhance job matching accuracy.

*1) Training Data:* The training dataset is structured with a variety of attributes related to job openings, where some fields are directly provided and others are predicted by external models:

- **Title**: The title of the job opening, providing a succinct summary of the role.
- **Description**: A detailed account of the job responsibilities, requirements, and other relevant details.
- **Flat Skills**: Predicted. This column contains a list of skills necessary for the job, as predicted by an external model. The accuracy of this information may vary, and missing values indicate either no applicable skills or low confidence in prediction.
- **ESCO Occupation Label**: Predicted. Each job is assigned a predicted occupation category based on the ESCO classification system. This helps categorize the job into predefined occupational groups but may have some inaccuracies.
- **Contract Type**: Predicted. A model predicts the type of contract (permanent, temporary, or seasonal), offering insights into the nature of the offered employment.
- **Education Levels**: Predicted. This field shows the anticipated educational qualifications needed for the job, aiding in understanding the educational requirements, but it may contain inaccuracies.
- **Employment Types**: Predicted. The type of employment (e.g., apprenticeship) is predicted, offering insights into the nature of the job but potentially lacking precision.
- **Experience**: Predicted. This attribute indicates the level of experience required for the job, as predicted by an external model. The predictions may not always be accurate or complete.
- **Working Schedule**: Predicted. The working schedule (fixed, flexible, or seasonal) is predicted, which provides information on job flexibility but may not always be precise.
- **Working Times**: Predicted. This feature, indicating whether the job is part-time, full-time, or allows both, is predicted by a model and might have some variability in accuracy.

The predictions of the noted columns are derived from external models and are subject to potential inaccuracies.

In the creation of this benchmark, we utilized columns with consistent values (lack of missing value), which results in the following final list of utilized attributes:

- Description
- Flat Skills
- ESCO Occupation Label
- Contract Type
- Employment Types
- Working Times

*2) Testing Data:* The testing dataset is composed of pairs of job descriptions, each accompanied by a similarity score. This dataset is used to evaluate the performance of the sentence similarity models by assessing their ability to determine the similarity between different job descriptions. The inclusion of both English and German descriptions ensures that the model's performance is tested across multiple languages, providing a comprehensive evaluation of its effectiveness.

### B. Models

For this study, due to their popularity and compact size, the following models have been chosen:

- paraphrase-multilingual-MiniLM-L12-v2
- paraphrase-MiniLM-L12-v2

- paraphrase-MiniLM-L6-v2
- paraphrase-MiniLM-L3-v2
- paraphrase-mpnet-base-v2
- paraphrase-distilroberta-base-v2
- paraphrase-TinyBERT-L6-v2
- paraphrase-albert-small-v2
- distiluse-base-multilingual-cased-v2

These models are part of the Sentence-BERT (SBERT) model family introduced by Reimers et al. in [10].

- all-MiniLM-L12-v2
- all-MiniLM-L6-v2
- all-mpnet-base-v2

Hugging Face's 'Community Week using JAX/Flax for NLP & CV' event featured the development of these models as part of the 'Train the best sentence embedding model ever with 1B training pairs' project [9].

- sentence-t5-base

This model was introduced by Ni et al. in [7].

To provide a better understanding of the complexity of each model, Figure 1 shows a bar plot of the parameter count for each model. This plot allows for a visual comparison of the model sizes, which can be an important factor in determining their performance and efficiency.

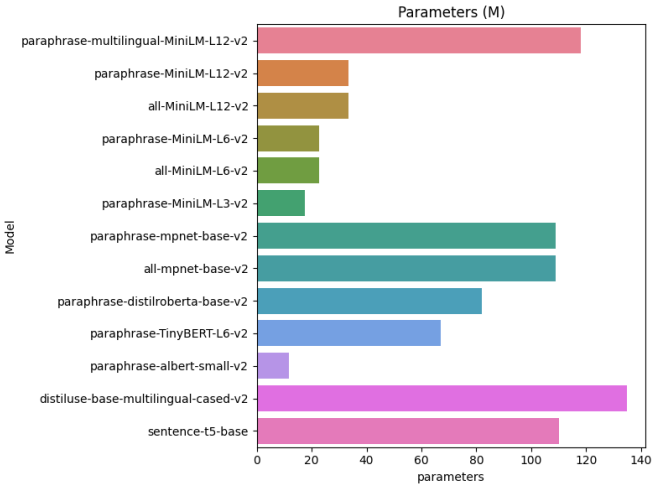


Fig. 1. Sentence Transformers Parameters (M)

### C. Fine-Tuning

In our fine-tuning procedure for sentence similarity, we utilize a triplet loss approach, as in [11], to enhance the model's capability to discern between semantically similar and dissimilar job descriptions. The training process involves the following components:

*1) Triplet Loss:* Each training batch comprises three types of job descriptions:

- **Anchor**: The reference job description used for comparison.
- **Positive**: A job description semantically similar to the anchor, which the model should learn to associate closely with the anchor.
- **Negative**: A job description not semantically similar to the anchor, which the model should learn to distinguish from the positive.

The triplet loss function is defined as:

$$L = max(0, d(a,p) - d(a,n) + \alpha) \tag{1}$$

where d(a,p) is the distance between the anchor and the positive description, d(a,n) is the distance between the anchor and the negative description, and $\alpha$ is the margin that separates the positive and negative pairs. The goal is to minimize this loss, which encourages the model to ensure that the anchor is closer to the positive description than to the negative description by at least a margin $\alpha$ [11].

*2) Training Setup:* The training setup is configured with the following parameters:

- **Batch Size**: A batch size of 4 is used, which is suitable for our training setup and GPU capacity.
- **Epochs**: Training is performed for 1 epoch per model, allowing for initial validation of the training process and initial assessment of the model's ability.
- **Hardware**: Training is conducted on an NVIDIA RTX 4070 Ti GPU, which provides the necessary computational power to handle the model's training efficiently.
- **Library**: We utilize the Sentence Transformers library, which facilitates the implementation and fine-tuning of models for sentence similarity tasks.

Figure 2 shows the training time for each model, illustrating the duration of the training process in seconds.

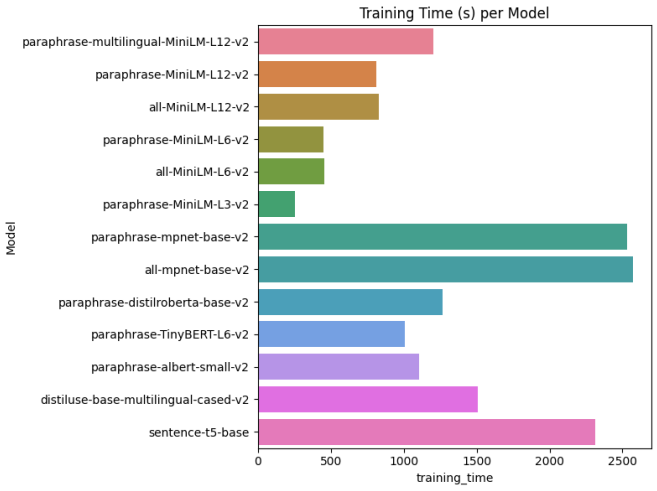


Fig. 2. Sentence Transformers Training Time

## IV. Results & Evaluation

The fine-tuned sentence transformer models were evaluated using two key metrics: the Pearson correlation between cosine distances of the embedding vectors and inference times on the

test set. The results are summarized in Table I, while a detailed comparison is provided below.

### A. Cosine Pearson Correlation

There is a strong link between the cosine distance of encoded test samples and true similarity scores, as shown in 3. The paraphrase-multilingual-MiniLM-L12-v2 model does the best, with a correlation of 0.3426. It is followed by paraphrase-albert-small-v2 (0.3407) and paraphrase-distilroberta-base-v2 (0.3386). The lowest performers are all-MiniLM-L6-v2 (0.1808), all-MiniLM-L12-v2 (0.1848), and all-mpnet-base-v2 (0.2160). This comparison shows that models trained on paraphrase detection consistently outperform models trained on a wide range of tasks. This shows the benefits of choosing task specific models.

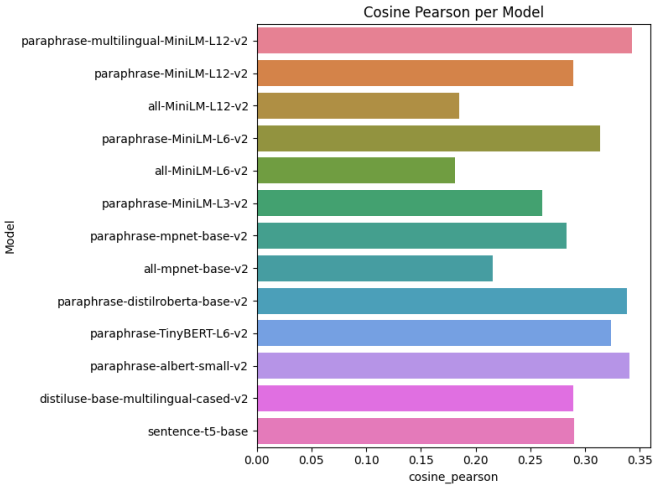


Fig. 3. Sentence Transformers Cosine Pearson Correlation

### B. Comparison with Base Model Performance

One important thing to consider is the difference in the models' performance before and after fine-tuning. As can be seen in Figure 4, all models had different changes in their performance.

The comparison shows that in general all models that were pre-trained solely on paraphrase detection had a significant increase in performance compared to the base models, while models that were pre-trained on multiple tasks had a decrease of up to -32.76% in performance after fine-tuning on the dataset. Upon examining the comprehensive data in the benchmark table (Table I), it becomes evident that the sole factor influencing performance increases or decreases compared to the base model, is dependent on whether the base model was pre-trained on paraphrase detection or on multiple tasks.

The sentence-t5-base model, which builds on top of T5 and can also perform classification, clustering, and other NLP tasks [7], appears to be the exception to this rule. Its performance has increased by 34.78%.

Other notable models are paraphrase-MiniLM-L12-v2, which achieved the highest increase in performance, with

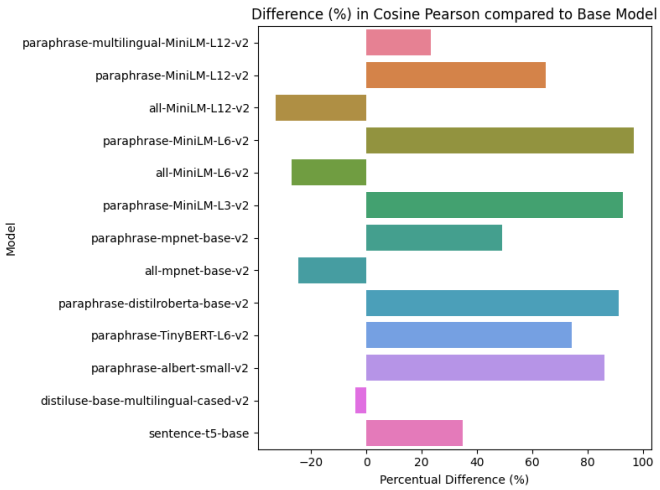


Fig. 4. Sentence Transformers Percentual Difference of Cosine Pearson

96.69% and paraphrase-albert-small-v2 being the smallest model (Figure 1) with the second heighest cosine Pearson correlation (Figure 3) and an increase of 86.15% .

Important to consider is that this benchmark focuses on the performance of models after a single epoch of training, as mentioned in section III-C. Other results may occur when fine-tuning each single model to their optimal state.

### C. Inference Times

In addition to evaluating the models' performance in cosine Pearson correlation, we also assessed their efficiency by measuring their inference time. Specifically, we recorded the total inference time and the inference time per sample, over the 5000 samples in the test set, and present the results in Figure 5 and Table I. This analysis allows us to gain insight into the models' processing speed and identify potential bottlenecks in their architecture.

The result shows that paraphrase-MiniLM-L3-v2 model had the fastest inference (6.6 seconds total, 0.0013 seconds per sample), followed by the MiniLM-L6-v2 models (10.9 seconds total). In contrast, the mpnet-base-v2 models required 73.1 seconds, a 11.1-fold increase over the MiniLM-L3-v2 model.

These findings underscore the substantial disparities in inference time among the evaluated models, emphasizing the importance of selecting the most suitable model for a specific application, which often necessitates a trade-off between accuracy and computational efficiency.

### D. Cosine Pearson Correlation and Inference Times

Considering the cosine Pearson correlation and inference time together allows for a more comprehensive evaluation of the models' performance. When looking at the benchmark table, Table I, the side-by-side comparison highlights the strength of the top three models in a broader perspective.

The paraphrase-multilingual-MiniLM-L12-v2 model stands out with the highest cosine Pearson correlation coefficient of 0.3426, as previously shown in section IV-A, and a low

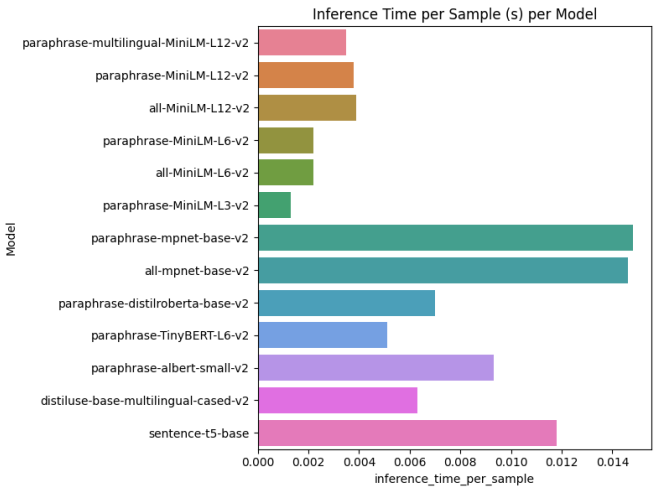


Fig. 5. Sentence Transformers Inference Time per Sample

average inference time of 0.0035 seconds per sample. This makes it both the most accurate and a computationally efficient solution, ideal for applications requiring high accuracy and fast processing. Its performance across both metrics highlight the models well-rounded capabilities.

Similarly, the paraphrase-albert-small-v2 model performs well, achieving a cosine Pearson correlation of 0.3407, just behind the paraphrase-multilingual-MiniLM-L12-v2 model. Although its inference time of 0.0093 seconds per sample is higher, its overall performance remains competitive, making it a viable alternative where a marginally slower inference time is acceptable.

### *E. Combined Result & Evaluation*

To provide an even more comprehensive evaluation, we now consider the models' performance alongside their training duration and model size. Although training duration may not be critical in this case due to the small data size, it becomes a bottleneck with larger or continuously growing datasets, such as those with continuous customer feedback. In such scenarios, faster training and adaptation are crucial for maintaining a competitive edge.

Model size is another key factor, impacting not only training speed but also batch inference scalability. Smaller models allow for larger batch sizes in both training and inference, increasing throughput. Additionally, smaller models reduce memory requirements, making them more suitable for devices with limited resources, such as edge or mobile devices.

The radar chart (Figure 6) offers a comprehensive view of all four factors, providing a clear comparison of each model's strengths and weaknesses. This visual representation aids decision-making by highlighting the trade-offs.

From the chart, it is clear that the paraphrase-multilingual-MiniLM-L12-v2, with the highest cosine Pearson coefficient, performs well in all other categories except model size, as it is the second largest model. Its good performance makes it a strong choice for production environments, capable of handling diverse tasks and scenarios.

In contrast, the paraphrase-albert-small-v2 model, which was previously identified as the second-strongest model in Section IV-D, reveals a remarkable characteristic: an extremely small model size. Specifically, it is ten times smaller than the paraphrase-multilingual-MiniLM-L12-v2 model. This significant size reduction enables faster training and increased inference capacity when utilizing higher batch sizes. In production environments with limited budget or compute resources, this model can serve several times more customers than the paraphrase-multilingual-MiniLM-L12-v2 model making it a compelling choice in this regard.

Another interesting model is distiluse-base-multilingual-cased-v2, where the base model has a cosine Pearson correlation of 0.3017, which comes close to the top-performing fine-tuned models, making it an excellent choice for application prototyping, though it can be challenging to deploy on a larger scale with a limited budget due to its comparatively large size.

## V. Conclusion & Future Work

In conclusion, our evaluation of various sentence transformers has revealed that the optimal model choice is not as straightforward as it may seem. While model performance on the specific problem is a crucial factor, it is not the only one to consider. Our analysis has shown that models with great performance may not necessarily be the best choice in all scenarios, as they may come with significant computational costs, larger model sizes, or longer training times.

On the other hand, models that may not be the most accurate can still offer advantages in terms of computational efficiency, smaller model sizes, or faster training times. Our findings suggest that the correct model choice depends on a comprehensive evaluation of multiple factors, including accuracy, inference speed, training duration, and model size.

The paraphrase-multilingual-MiniLM-L12-v2 model, for example, offers the highest accuracy but may not be the best choice for production environments with limited computational resources. In contrast, the paraphrase-albert-small-v2 model, while slightly less accurate, offers a significant advantage in terms of model size and computational efficiency, making it a more suitable choice for certain applications.

These results emphasize the need to consider multiple evaluation metrics when selecting a model. Assumptions like "bigger is better" or choosing the top-performing model are not always valid. A comprehensive approach helps identify the optimal model by balancing accuracy, efficiency, and computational resources.

Future work can build upon the insights gained from this study by exploring the application of sentence similarity models in more diverse and complex scenarios or by creating a benchmark that trains all models as close as possible to their optimal state instead of a single epoch.

The development of more sophisticated evaluation metrics that do consider the specific requirements of real-world applications, such as latency, memory constraints, and inter-

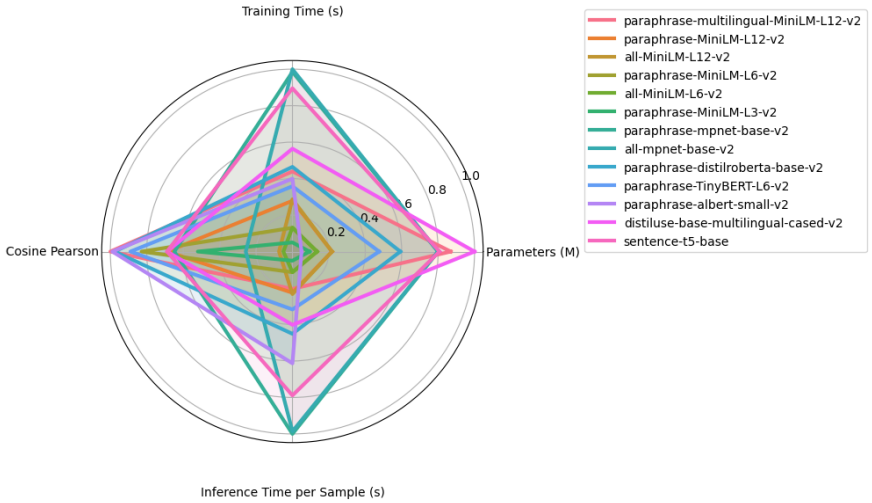


Fig. 6. Sentence Transformers Relative Training Time, Parameters, Inference Time and Cosine Pearson Correlation

| Model | Parameters (M) | Training Time (s) | Cosine Pearson | Base Model Cosine Pearson | Total Inference Time (s) | Inference Time per Sample (s) |
|---|---|---|---|---|---|---|
| paraphrase-multilingual-MiniLM-L12-v2 | 118 | 1204 | **0.3426** | 0.2774 | 17.431 | 0.0035 |
| paraphrase-MiniLM-L12-v2 | 33.4 | 812 | 0.2895 | 0.1756 | 19.2243 | 0.0038 |
| all-MiniLM-L12-v2 | 33.4 | 832 | 0.1848 | 0.2749 | 19.3172 | 0.0039 |
| paraphrase-MiniLM-L6-v2 | 22.7 | 451 | 0.3134 | 0.1594 | 10.8922 | 0.0022 |
| all-MiniLM-L6-v2 | 22.7 | 454 | 0.1808 | 0.2474 | 10.9884 | 0.0022 |
| paraphrase-MiniLM-L3-v2 | 17.4 | **255** | 0.2608 | 0.1352 | **6.6432** | **0.0013** |
| paraphrase-mpnet-base-v2 | 109 | 2534 | 0.2827 | 0.1896 | 73.9009 | 0.0148 |
| all-mpnet-base-v2 | 109 | 2570 | 0.2160 | 0.2861 | 73.0814 | 0.0146 |
| paraphrase-distilroberta-base-v2 | 82.1 | 1265 | 0.3386 | 0.1770 | 35.1753 | 0.0070 |
| paraphrase-TinyBERT-L6-v2 | 67 | 1006 | 0.3238 | 0.1860 | 25.4528 | 0.0051 |
| paraphrase-albert-small-v2 | **11.7** | 1107 | 0.3407 | 0.1830 | 46.4308 | 0.0093 |
| distiluse-base-multilingual-cased-v2 | 135 | 1508 | 0.2895 | **0.3017** | 31.3898 | 0.0063 |
| sentence-t5-base | 110 | 2315 | 0.2896 | 0.2149 | 59.1648 | 0.0118 |

TABLE I
SENTENCE TRANSFORMER MODELS PARAMETER, TRAINING TIME, COSINE PEARSON CORRELATION AND INFERENCE TIMES

pretability, can provide a more comprehensive understanding of model performance. Furthermore, the exploration of novel model architectures and training techniques that can efficiently balance accuracy, efficiency, and computational resources can lead to more robust and practical solutions.

## VI. ACKNOWLEDGEMENT

We sincerely thank Joblift GmbH for sharing the dataset with us and permitting its publication.

## REFERENCES

[1] Silvia Casola, Ivano Lauriola, and Alberto Lavelli. Pre-trained transformers: an empirical comparison. *Machine Learning with Applications*, 9:100334, 2022.
[2] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
[3] Mamdouh Farouk. Measuring sentences similarity: a survey. *arXiv preprint arXiv:1910.03940*, 2019.
[4] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42, 2022.
[5] Joblift. Joblift: all jobs on one platform.
[6] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.
[7] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models, 2021.
[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
[9] Nils Reimers. Train the best sentence embedding model ever with 1b training pairs, Jun 2021.
[10] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
[11] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.
[12] Tedo Vrbanec and Ana Meštrović. Corpus-based paraphrase detection experiments and review. *Information*, 11(5):241, 2020.