
TURKEMBED: TURKISH EMBEDDING MODEL ON NATURAL LANGUAGE INFERENCE & SENTENCE TEXT SIMILARITY TASKS *

Özay Ezerceli
New Mind AI
Istanbul, Türkiye
oezerceli@newmind.ai

Gizem Gümüşcekiçi, Tuğba Erkoç, Berke Özenç
Faculty of Engineering and Natural Sciences
Isik University
Istanbul, Türkiye
{gizem.gumuscekiçi,tugba.erkoc,berke.ozenc}@isikun.edu.tr

ABSTRACT

This paper introduces TurkEmbed, a novel Turkish language embedding model designed to outperform existing models, particularly in Natural Language Inference (NLI) and Semantic Textual Similarity (STS) tasks. Current Turkish embedding models often rely on machine-translated datasets, potentially limiting their accuracy and semantic understanding. TurkEmbed utilizes a combination of diverse datasets and advanced training techniques, including matryoshka representation learning, to achieve more robust and accurate embeddings. This approach enables the model to adapt to various resource-constrained environments, offering faster encoding capabilities. Our evaluation on the Turkish STS-b-TR dataset, using Pearson and Spearman correlation metrics, demonstrates significant improvements in semantic similarity tasks. Furthermore, TurkEmbed surpasses the current state-of-the-art model, Emrecaan, on All-NLI-TR and STS-b-TR benchmarks, achieving a 1-4% improvement. TurkEmbed promises to enhance the Turkish NLP ecosystem by providing a more nuanced understanding of language and facilitating advancements in downstream applications.

Keywords Semantic text similarity · matryoshka representation · embedding model · natural language inference · downstream task

1 Introduction

Natural Language Processing (NLP) is considered a branch of computational linguistics that focuses on enabling machines to understand, interpret, and generate human language [1]. It has a wide research area and many diverse applications, the most popular ones are sentiment analysis [2], machine translation, and sarcasm detection [3]. Many of these applications are popular topics that are currently being investigated to find better approaches to ongoing challenges. With recent advances in technology, the NLP field has seen remarkable advancements and there is a continuous need for improvement.

NLP applications primarily rely on embeddings to represent words or sentences. An embedding is a numerical representation of words, phrases, or sentences. It is used to convert textual data into numerical vector representations that preserve semantic and syntactic properties. Thus, the performance of embedding systems plays a crucial role in determining the success of NLP systems.

Word embeddings are categorized as non-contextual and contextual, each with distinct impacts on NLP models like TurkEmbed. Non-contextual embeddings, such as Word2Vec [4], GloVe [5], and FastText [6], assign fixed representations to words, ignoring variations in meaning based on context [7]. For instance, the word "bank" is represented identically whether it refers to a "riverbank" or a "financial institution." This limitation makes non-contextual embeddings inadequate for advanced tasks like Natural Language Inference (NLI) and Semantic Textual Similarity (STS), which require contextual understanding [8].

* *Citation:* Özay Ezerceli, Gizem Gümüşcekiçi, Tuğba Erkoç, Berke Özenç. "TurkEmbed: Turkish Embedding Model on Natural Language Inference & Sentence Text Similarity Tasks." 2025 IEEE 11th International Conference on Advances in Software, hardware and Systems Engineering (ASYU), 2025. DOI: 10.1109/ASYU67174.2025.11208511

Contextual embeddings, such as BERT [9], ELMo [10], and T5 [11], address this by generating dynamic representations that adapt to a word’s usage within a sentence, enabling nuanced interpretation. This is crucial for Turkish, a morphologically rich language. TurkEmbed leverages techniques like matryoshka representation learning [12] to optimize contextual embeddings, overcoming challenges in Turkish morphology and syntax. These advancements position TurkEmbed as a state-of-the-art model for Turkish NLP, highlighting the importance of contextual embeddings in capturing semantic nuances.

Embedding models are often language-dependent, with multilingual versions available but potentially less effective for low-resource languages like Turkish due to the need for generalization across multiple languages. While most embedding models are built for English due to the abundance of resources, Turkish NLP research relies on multilingual models or a limited number of Turkish-specific models, which can hinder performance in tasks requiring a deep understanding of semantic relationships; even models like bert-base-turkish-cased-mean-nli-stsb-tr [13] have room for improvement in semantic similarity tasks.

We introduce TurkEmbed, a novel and enhanced Turkish embedding model designed to overcome existing limitations in the Turkish language. The methodology involves combining diverse datasets with advanced training techniques, notably Matryoshka representation learning, and selecting base models from the MTEB leaderboard [14]. TurkEmbed’s performance was evaluated on semantic similarity tasks using the Turkish STSb [15] and STS22 [16] datasets, showing superior results compared to current state-of-the-art models. The main contributions include the TurkEmbed model itself, which excels on NLI and STS tasks, the demonstration of Matryoshka learning’s efficacy for Turkish, and a thorough evaluation on benchmarks like All-NLI-TR, STSb-TR, and STS22-Crosslingual-STS, ultimately aiming to advance Turkish NLP capabilities.

The remainder of this paper is organized as follows: Section 2 reviews related works on Turkish embedding models. Section 3 details the methodology employed in developing TurkEmbed. Section 4 describes the experimental setup, and Section 5 presents the results and discussion. Finally, Section 6 concludes the paper and outlines future research directions.

2 Related Work

The Turkish language, characterized by its agglutinative morphology and rich vocabulary, presents a unique and compelling challenge for natural language processing (NLP). The development of effective word embedding models is paramount to overcoming these linguistic complexities and enabling robust NLP tools for Turkish.

Contextual embedding models like BERT [17] and ELMo [10] significantly changed NLP by overcoming the limitations of non-contextual approaches. ELMo employed deep bidirectional language models for context-aware representations, whereas BERT used the Transformer architecture with bidirectional encoders to achieve state-of-the-art results across many NLP tasks. BERT’s capacity to consider the full context of a word was particularly advantageous for Turkish, facilitating a more nuanced understanding of meaning in complex sentences. Research confirmed that BERT-based models outperformed earlier non-contextual methods in various Turkish NLP tasks, highlighting the crucial role of contextual awareness for the language.

Building upon the success of BERT, Turkish-specific BERT models are developed and trained on large-scale Turkish corpora like the Boun Web Corpus [18] and the Huawei Corpus [19]. These models, pre-trained on extensive Turkish text, offered enhanced performance on downstream tasks due to a better understanding of language-specific nuances. Furthermore, research has explored adapting and fine-tuning these models for specific Turkish NLP tasks. [13]’s work, for example, utilized machine-translated datasets to fine-tune BERT-based models for Turkish NLI and STS, establishing initial benchmarks for these tasks and highlighting the feasibility of cross-lingual transfer learning for Turkish NLP. The YTU Cosmos Lab [20] further introduced Turkish BERT models trained on a sizable corpus of 75GB, compiled from various sources to enhance the diversity and representativeness of the data. These models aimed to improve performance on downstream tasks such as text classification and named entity recognition. Although they provided a solid foundation, they required significant computational resources for training and did not specifically address tasks like NLI and STS.

Beyond core NLP tasks, adaptations of existing architectures have also emerged. Turkish-ColBERT [21] adapted the ColBERT architecture, initially designed for English information retrieval, to the Turkish language, showcasing the adaptability of advanced retrieval models to morphologically complex languages. It was fine-tuned on the machine-translated MS MARCO dataset [22], utilizing over 500,000 translated queries and passages. While it demonstrated improved retrieval performance, the reliance on machine-translated data posed challenges in capturing idiomatic expressions and linguistic nuances unique to Turkish.

In addition, the exploration of multilingual models such as XLM-ROBERTa [23] and multilingual E5 [24] has opened the doors for the learning of cross-lingual transfer in Turkish NLP. These models, trained on data from more than 100 languages, leverage shared representations to improve performance in low-resource languages such as Turkish, offering a cost-effective approach to building robust Turkish NLP systems. Similarly, the GTE-multilingual-base model [25], providing generalized embeddings suitable for various tasks across multiple languages, offers a versatile solution for Turkish NLP, particularly in cross-lingual applications.

In addition, general embedding models show promise for Turkish NLP. One such model is nomic-ai/nomic-embed-text-v2-moe [26], a state-of-the-art multilingual text embedding model using a Mixture of Experts (MoE) architecture. Trained on over 1.6 billion data pairs across approximately 100 languages, including Turkish, it offers competitive performance for multilingual retrieval tasks, making it efficient for Turkish applications requiring cross-language capabilities.

Another significant model is Alibaba-NLP/gte Modernbert-base [25], part of their GTE series. It is a ModernBERT base model language support is English. The GTE series includes models supporting a wide range, potentially including Turkish, with multilingual variants designed for long context lengths and trained on diverse datasets. These models suggest that, with fine-tuning or adaptation, they could further enhance Turkish NLP tasks, especially those involving longer texts or cross-language comparisons, opening exciting avenues for future research.

3 Methodology

3.1 Model Selection

A comprehensive selection process was undertaken to identify appropriate base models for subsequent fine-tuning on Turkish language tasks, considering leading native English, Turkish, and multilingual candidates known for capturing language-specific nuances. The selection criteria prioritized models with demonstrated effectiveness in cross-lingual transfer learning and semantic understanding capabilities, particularly for morphologically rich languages like Turkish.

Among the evaluated models were ModernBERT-base (150M parameters) and its larger variant ModernBERT-large (396M) [27], representing state-of-the-art encoder architectures with improved efficiency; the instruction-tuned KaLM-embedding-multilingual-mini-instruct-v1.5 (494M) [28], specifically designed for multilingual embedding tasks; the compact paraphrase-multilingual-MiniLM-L12-v2 (118M) suitable for resource-constrained scenarios [29]; the generalized multilingual text embedding models GTE-multilingual-base (305M) and gte-modernbert-base (149M) from Alibaba’s Tongyi Lab [25], known for their strong multilingual capabilities; the XLM-RoBERTa-based multilingual-E5-large-instruct (560M) [24], which has demonstrated superior performance on various multilingual benchmarks; and the multilingual Mixture-of-Experts model, nomic-embed-text-v2-moe [26], offering scalable parameter efficiency.

The model selection process considered computational efficiency, multilingual capabilities, and proven performance on semantic similarity tasks. Models with strong cross-lingual transfer capabilities were prioritized, given the limited availability of high-quality Turkish training data compared to resource-rich languages like English.

3.2 Loss Functions and Their Theoretical Foundations

The selection of loss functions for our two-stage training pipeline was guided by both theoretical considerations and empirical evidence from recent advances in sentence embedding research. Multiple Negatives Ranking Loss [30], employed with the All-NLI-TR dataset, was selected for the initial training stage due to its contrastive learning efficiency that leverages in-batch negatives, treating all other sentences in the batch as negative examples for each anchor-positive pair. This approach enables efficient learning from a large number of negative examples without requiring explicit negative sampling strategies [31], and its contrastive nature aligns naturally with NLI data structure where positive pairs (entailment relationships) must be distinguished from negative pairs [29].

CoSENT Loss [32], applied to the STSB-TR dataset in the second training stage, was chosen for its training-inference consistency that directly optimizes cosine similarity between sentence embeddings, creating perfect alignment between the training objective and the inference-time similarity metric. This consistency reduces the gap between training and deployment while providing smoother gradients across the full range of similarity scores, allowing for more stable training and better convergence on continuous similarity prediction tasks [33]. Research has demonstrated that CoSENT Loss produces better-calibrated similarity scores that align more closely with human similarity judgments [32].

Matryoshka Loss [12] integrates with both primary loss functions, enabling the model to learn embeddings across multiple dimensions concurrently. This integration provides adaptive deployment capabilities allowing a single model to generate useful embeddings at various dimensions (64, 128, 256, 512, 768), dimension efficiency where even truncated

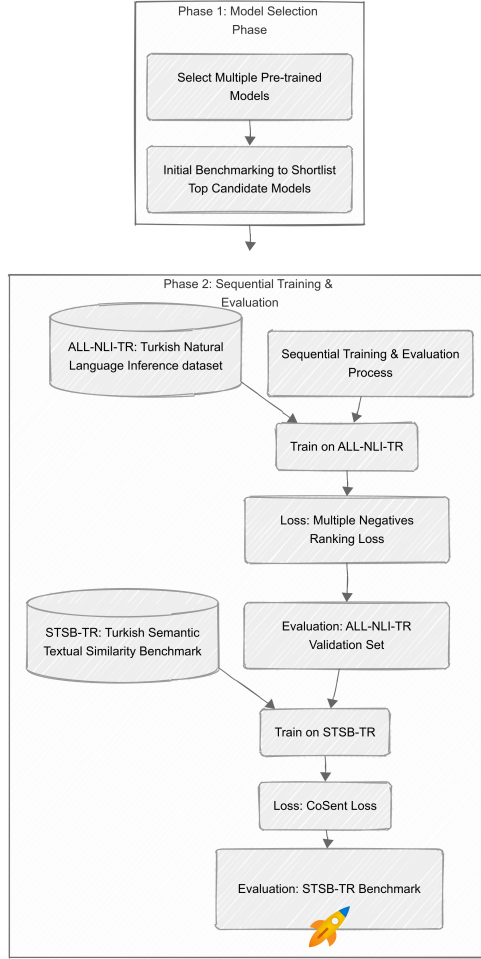


Figure 1: TurkEmbed Sequential Training Pipeline

embeddings maintain competitive performance, and resource optimization that reduces computational and storage requirements for production environments [34].

3.3 Training Procedure and Sequential Learning Rationale

Our training methodology employs a carefully designed two-stage sequential fine-tuning process, illustrated in Figure 1, addressing the specific challenges of developing high-quality Turkish language embeddings. The All-NLI-TR dataset was selected as the initial training corpus because NLI datasets provide explicit semantic relationships (entailment, contradiction, neutral) that force models to learn fine-grained semantic distinctions essential for high-quality embeddings [35]. Multiple studies have demonstrated that NLI training creates robust sentence representations that transfer effectively to various semantic tasks, including STS, developing structural knowledge about semantic relationships that generalizes well across domains [35, 36]. Additionally, NLI training improves cross-lingual transfer capabilities, particularly valuable for Turkish as a morphologically rich language with limited resources, and provides superior zero-shot transfer capabilities compared to models trained directly on specific downstream objectives [37, 38].

The decision to follow NLI training with STSB-TR fine-tuning is supported by substantial empirical evidence showing that while NLI focuses on categorical relationships between sentences, STS provides continuous similarity scores that help models refine their understanding of semantic similarity in a more nuanced way. Sequential fine-tuning on STSB-TR after NLI training helps prevent catastrophic forgetting of semantic distinctions learned in the first stage while enabling task-specific specialization, consistently outperforming simultaneous multi-task training and other training regimens [39, 29].

The training process begins with model initialization and sequence length adjustment, followed by first-stage fine-tuning on All-NLI-TR using Multiple Negatives Ranking Loss within the Matryoshka loss framework, with validation performed on All-NLI-TR and testing on STSB-TR. The second stage continues with STSB-TR fine-tuning using CoSENT Loss combined with Matryoshka loss to enhance sentence similarity capabilities. Final evaluation employs Triplet and Embedding Similarity Evaluators across various embedding dimensions [40]. Training durations varied according to model size and complexity, with All-NLI-TR training requiring approximately 30 minutes to 1 hour and STSB-TR fine-tuning taking 8 to 30 minutes. Smaller models like TurkEmbed (305M parameters) benefit from shorter training durations compared to larger models like multilingual-E5-large-instruct (560M parameters), demonstrating the efficiency and scalability of our approach.

4 Experiments

4.1 Experimental Setup

The experiments were conducted on a high-performance computing setup equipped with an NVIDIA A100 40GB GPU. Python 3.11.11, PyTorch 2.5.1+cu121, and Transformers 4.49.0.dev0, complemented by Sentence Transformers 3.3.1 and Datasets 3.2.0. HuggingFace’s Transformers and Datasets libraries were utilized for efficient model handling and dataset loading, ensuring seamless integration and scalability.

The hyperparameters were carefully selected to optimize model performance while preventing overfitting. Batch sizes of 16, 32, 64, and 128 were employed, depending on the model and GPU memory constraints. The learning rate was tuned within the range of 1×10^{-5} to 8×10^{-5} , and the number of training epochs was set between 1 and 10. To enhance training stability, a warmup ratio of 0.1 or specific warmup steps based on dataset size was applied. The maximum sequence length was adjusted to 75, 128, 256, or 512 tokens, depending on the model’s capacity and task requirements, ensuring efficient processing of input data.

Three key metrics were used to assess the performance of the model. The Pearson Correlation Coefficient measured the linear correlation between predicted and actual similarity scores, providing insights into the model’s ability to capture semantic relationships. Spearman’s Rank Correlation Coefficient evaluated the monotonic relationship between predicted and actual rankings, ensuring robustness in capturing relative similarities. Additionally, accuracy was used for Natural Language Inference (NLI) tasks to determine the percentage of correct predictions, offering a comprehensive assessment of the model’s overall effectiveness. These metrics collectively ensured a rigorous evaluation of TurkEmbed’s performance in various tasks and datasets.

4.2 Datasets

The All-NLI-TR dataset is a combination of the SNLI [41] and MultiNLI [42] datasets translated into Turkish. It contains 482,091 training samples, 6,802 for development, and 6,827 for testing, covering a diverse range of genres and topics. The dataset includes pairs of sentences labeled with entailment, contradiction, or neutral, providing a robust foundation for training models on NLI tasks.

The STSB-TR dataset [43] is a Turkish version of the Semantic Textual Similarity Benchmark, containing sentence pairs with similarity scores ranging from 0 to 5. It includes 5,749 training samples, 1,500 validation samples, and 1,379 test samples. This dataset enables models to learn fine-grained semantic relationships between sentences.

5 Results and Discussion

5.1 Performance on All Natural Language Inference

To rigorously assess the model’s resilience to catastrophic forgetting following sequential training, TurkEmbed’s performance was evaluated on the All-NLI-TR test set subsequent to its fine-tuning on the STSB-TR dataset. The results, presented in Table 1, indicate that TurkEmbed achieved superior performance compared to all other evaluated models, obtaining a cosine accuracy of 0.935 before stsb fine-tuning and 0.924 after fine-tuning. Notably, this surpasses the performance of strong contemporary models such as bge-m3 and paraphrase-multilingual-MiniLM-L12-v2 under the same evaluation conditions. This outcome suggests that the proposed training methodology incorporating Matryoshka representation learning effectively mitigates catastrophic forgetting, yielding a robust and versatile embedding model capable of retaining task-specific knowledge across different training phases within the Turkish NLP context.

Table 1: Performance on All-NLI-TR Test Set

Model	Max Seq Length	Embedding Dimension	Cosine Accuracy
gte-multilingual-base	8192	768	0.896
bge-m3	8192	1024	0.914
turkish-e5-large	514	1024	0.876
Qwen3-Embedding-8B	32000	32 to 4096	0.876
ModernBERT-base	8192	768	0.605
ModernBERT-large	8192	1024	0.601
KaLM-embedding-multi lingual-mini-instruct-v1.5	131072	896	0.864
paraphrase-multilingual -MiniLM-L12-v2	512	384	0.902
multilingual-E5-large -instruct	514	1024	0.881
nomic-embed- text-v2-moe	2048	256 to 768	0.821
Emreca's Model	512	768	0.885
TurkEmbed-All-NLI-TR	8192	64 to 768	0.935
TurkEmbed4STS	8192	64 to 768	<u>0.924</u>
modernbert-base-tr-uncased -allnli-stsb	8192	256 to 768	<u>0.924</u>

Table 2: Performance on STSB-TR Test Set

Model	Max Seq Length	Embedding Dimension	Pearson Cosine	Spearman Cosine
gte-multilingual-base	8192	768	0.804	0.804
bge-m3	8192	1024	0.795	0.797
turkish-e5-large	514	1024	0.795	0.800
Qwen3-Embedding-8B	32000	32 to 4096	0.798	0.794
ModernBERT-base	8192	768	0.758	0.749
ModernBERT-large	8192	1024	0.772	0.771
KaLM-embedding-multi lingual-mini-instruct-v1.5	131072	896	0.797	0.802
paraphrase-multilingual -MiniLM-L12-v2	512	384	0.812	0.825
multilingual-E5-large -instruct	514	1024	0.846	0.854
nomic-embed-text-v2-moe	2048	768	0.828	0.834
Emreca's Model	512	768	0.834	0.830
TurkEmbed-All-NLI-TR	8192	64 to 768	0.813	0.820
TurkEmbed4STS	8192	64 to 768	0.845	0.853
modernbert-base-tr-uncased -allnli-stsb	8192	256 to 768	0.825	0.832

5.2 Performance on Semantic Textual Similarity Benchmark

In the final fine-tuning on STSB-TR, TurkEmbed achieved state-of-the-art results for Turkish semantic tasks, with Pearson and Spearman correlations of 0.845 and 0.853, given in Table 2. It closely rivals multilingual-E5-large-instruct while using nearly half the parameters (305M vs. 560M), making it more efficient. It also outperforms models like nomic-embed-text-v2-moe and Emreca's Model. Models with poor Turkish performance, such as gte-modernBERT-base and IBM Granite, were excluded. TurkEmbed's strong accuracy and efficiency make it a top choice for Turkish NLP.

5.3 Evaluation on STS22-cross-lingual Semantic Textual Similarity (TR Subset)

To evaluate generalization capabilities, TurkEmbed4STS was assessed on the Turkish STS subset derived from the STS22-Crosslingual-STs dataset. The model achieved a Pearson cosine correlation of 0.646 and a Spearman cosine correlation of 0.668, as presented in the Table 3. These results position TurkEmbed4STS competitively, surpassing several models, including Emreca's Model and demonstrating performance comparable to the top-performing nomic-embed-text-v2-moe. Qwen3-Embedding-8B achieved the highest Pearson and Spearman cosine correlations at 0.701 and 0.721, respectively.

5.4 Inference Speed Comparison

Inference speed is a critical factor for real-world applications, especially for tasks requiring real-time processing. TurkEmbed's inference speed was compared with Emreca's model on the Google Colab T4 GPU using a batch size

Table 3: Performance on STS22-Crosslingual-STS

Model	Pearson Cosine	Spearman Cosine
gte-multilingual-base	0.647	0.669
bge-m3	0.663	0.698
turkish-e5-large	0.668	0.692
multilingual-E5-large-instruct	0.676	0.695
Qwen3-Embedding-8B	0.701	0.721
ModernBERT-base	0.436	0.471
ModernBERT-large	0.375	0.380
KaLM-embedding-multilingual-mini-instruct-v1.5	0.342	0.365
Emreca’s Model	0.540	0.563
NeoBERT	0.622	0.663
nomic-embed-text-v2-moe	0.653	0.706
Emreca’s Model	0.540	0.563
TurkEmbed4STS	0.646	0.668
modernbert-base-tr-uncased-allnli-stsb	0.520	0.559

of 32 and 10,000 samples from the All-NLI-TR dataset. For tensor type FP32, TurkEmbed’s encoding speed was approximately 310 sentences per second, which is 2.17 times slower than Emreca’s model. For tensor type FP16, TurkEmbed achieved an encoding speed of 1,561 sentences per second, 1.23 times slower than Emreca’s model, as given in Table 4. This difference in speed is largely attributed to TurkEmbed’s larger size, as it has nearly three times the parameters of Emreca’s model. Despite the slower speed, TurkEmbed’s advanced architecture and higher accuracy make it a valuable choice for applications prioritizing performance over speed.

Table 4: Inference Speed Comparison

Model	Inference Speed (sentences/sec)	Tensor Type
Emreca’s Model	~675	FP32
Emreca’s Model	~1933	FP16
TurkEmbed	~310	FP32
TurkEmbed	~1561	FP16

6 Conclusion

This paper introduced TurkEmbed, a novel Turkish embedding model addressing the limitations of existing approaches, particularly the reliance on machine-translated datasets and difficulties capturing Turkish morphology. By fine-tuning strong multilingual base models (gte-multilingual-base, multilingual-e5 large-instruct) with advanced techniques, including Matryoshka representation learning, TurkEmbed achieves state-of-the-art performance on Turkish NLI (ALL-NLI-TR) and STS (STSB-TR) benchmarks, evaluated using Pearson and Spearman correlations. These results demonstrate the effectiveness of adapting multilingual models to enhance language specificity for resource-limited, morphologically rich languages. Future research will explore larger model architectures, integration of native Turkish datasets, transfer learning opportunities, and the evaluation of TurkEmbed across diverse downstream applications and real-world deployment scenarios.

References

- [1] A. B. A. Girgin and G. Gümüşçekiçi. From past to present: Spam detection and identifying opinion leaders in social networks. *Sigma Journal of Engineering and Natural Sciences*, 40(2):441–463, 2022.
- [2] Ö. Ezerçeli and R. Dehkharghani. Mental disorder and suicidal ideation detection from social media using deep neural networks. *Journal of Computational Social Science*, 7(3):2277–2307, 2024.
- [3] A. B. A. Girgin, G. Gümüşçekiçi, and N. C. Birdemir. Turkish sentiment analysis: A comprehensive review. *Sigma Journal of Engineering and Natural Sciences*, 42(4):1292–1314, 2024.
- [4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.
- [5] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical*

- Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017.
 - [7] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey, 2023.
 - [8] Alessio Miaschi and Felice Dell’Orletta. Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In Spandana Gella, Johannes Welbl, Marek Rei, Fabio Petroni, Patrick Lewis, Emma Strubell, Minjoon Seo, and Hannaneh Hajishirzi, editors, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 110–119, Online, July 2020. Association for Computational Linguistics.
 - [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
 - [10] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.
 - [11] 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, 2023.
 - [12] Aditya Kusupati, Girish Bhatt, Aaditya Rege, Matthew Wallingford, Aniruddha Sinha, Vivek Ramanujan, and Ali Farhadi. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.
 - [13] EmreCan. Bert-base-turkish-cased-mean-nli-stsb-tr. Hugging Face, 2024.
 - [14] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark, 2023.
 - [15] Figen Beken Fikri, Kemal Oflazer, and Berrin Yanikoglu. Semantic similarity based evaluation for abstractive news summarization. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 24–33, Online, August 2021. Association for Computational Linguistics.
 - [16] Tomáš Hercig and Pavel Král. Evaluation datasets for cross-lingual semantic textual similarity. pages 524–529, 01 2021.
 - [17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
 - [18] Haşim Sak, Tunga Güngör, and Murat Saraçlar. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *GoTAL 2008*, volume 5221 of *LNCS*, pages 417–427. Springer, 2008.
 - [19] Mehmet Şen and Hakan Erdogan. Learning word representations for turkish. pages 1742–1745, 04 2014.
 - [20] H. Tugrul Kesgin, Murat Kemal Yuce, and Mehmet Fatih Amasyali. Developing and evaluating tiny to medium-sized turkish bert models. *arXiv preprint*, arXiv:2307.14134, 2023.
 - [21] Keshav Santhanam, Omar Khattab, Jonathan Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint*, arXiv:2112.01488, 2021.
 - [22] Tri Nguyen, Miranda Rosenberg, Xiaodong Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human-generated machine reading comprehension dataset. 2016.
 - [23] Alexis Conneau. Unsupervised cross-lingual representation learning at scale. *arXiv preprint*, arXiv:1911.02116, 2019.
 - [24] Linjie Wang, Nan Yang, Xiaodong Huang, Liangyou Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint*, arXiv:2402.05672, 2024.
 - [25] Xingyu Zhang, Yeqi Zhang, Dong Long, Weizhi Xie, Zhen Dai, Jinfeng Tang, and Ming Zhang. Mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint*, arXiv:2407.19669, 2024.
 - [26] Zach Nussbaum and Brandon Duderstadt. Training sparse mixture of experts text embedding models, 2025.
 - [27] Ben Warner, Austin Chaffin, Baptiste Clavié, Olivia Weller, Oscar Hallström, Samir Taghadouini, and Ilaria Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint*, arXiv:2412.13663, 2024.
 - [28] Xiaoyu Hu, Zihan Shan, Xing Zhao, Zheng Sun, Zhilin Liu, Ding Li, and Ming Zhang. Kalm-embedding: Superior training data brings a stronger embedding model. *arXiv preprint*, arXiv:2501.01028, 2025.

- [29] Nils Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint*, arXiv:1908.10084, 2019.
- [30] Leo Gao, Yifan Zhang, Jheng-Hong Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. *arXiv preprint*, arXiv:2101.06983, 2021.
- [31] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020.
- [32] Jing SU. Cosent (1): A more efficient sentence vector scheme than sentence-bert, January 2022.
- [33] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.
- [34] Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2d matryoshka sentence embeddings. *arXiv preprint arXiv:2402.14776*, 2024.
- [35] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.
- [36] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.
- [37] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019.
- [38] Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Learning semantic textual similarity from conversations. *arXiv preprint arXiv:1804.07754*, 2018.
- [39] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.
- [40] 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.
- [41] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. *arXiv preprint*, arXiv:1508.05326, 2015.
- [42] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint*, arXiv:1704.05426, 2017.
- [43] Tomáš Hercig and Pavel Král. Evaluation datasets for cross-lingual semantic textual similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 524–529, 2021.