LLMs4Subjects: LLM-based Automated Subject Tagging for a National Technical Library's Open-Access Catalog

Vincenzo Avantaggiato
Politecnico di Torino
s323112
s323112@studenti.polito.it

Michele Cazzola
Politecnico di Torino
s323270
s323270@studenti.polito.it

Andrea Delli
Politecnico di Torino
s331998
s331998@studenti.polito.it

Abstract

This paper presents a solution for *SemEval'25 Task #5*, which focuses on leveraging large language models (LLMs) for subject tagging of technical records from Leibniz University's Technical Library (TIBKAT) using the GND taxonomy. The task involves bilingual language modeling, as the system must process technical documents in both German and English. We explore three different approaches of increasing complexity: embedding-based retrieval, fine-tuned embedding models, and binary classification models. Our experiments demonstrate that fine-tuning a multilingual sentence transformer model yields the best performance, significantly improving the quality of subject tagging. We also provide a comprehensive evaluation of the models' performance, including precision, recall, and F1 score, and discuss the trade-offs between different approaches. The source code of this project is available at `https://github.com/RonPlusSign/llms4subjects`.

1 Introduction

The objective of this work is to develop a solution for *SemEval'25 Task #5*[1]. This task focuses on leveraging large language models (LLMs) for subject tagging of technical records from Leibniz University's Technical Library (TIBKAT) using the GND taxonomy. A key challenge of this task is bilingual language modeling, as the system must process technical documents in both German and English.

The provided dataset consists of two types of files: GND and TIBKAT.

The **GND** taxonomy (*Gemeinsame Normdatei* in German, or *Integrated Authority File* in English) is an international authority file used primarily by German-speaking libraries to catalog and interlink information on people, organizations, topics, and works. This dataset represents the tags to assign to the different documents.

The **TIBKAT** dataset is composed of technical records annotated with GND subject tags, and it has been used for training and development. These documents are organized in different folders, based on the type of document (Article, Book, Conference, Report, Thesis) and language (`en`, `de`). This dataset is also already divided in three splits: `train`, `dev`, and `test`, where the last one is not provided with true labels (correct GND tags) because this split is intended for the challenge submission.

Both datasets have already been pre-processed and converted to JSON and JSONLD format for convenience, with a uniform format including key informations, such as name, description and abstract. The original repository also provided additional datasets not strictly related to TIBKAT's document tagging, but they have not been used in our work.

To address this task, we explore three different approaches of increasing complexity.

Embedding-based retrieval: Generate embeddings for both documents and tags using an encoder LLM, compute cosine similarity, and assign the top-k highest scoring tags.

Fine-tuned embedding model: Fine-tune a transformer-based encoder and apply the previous embedding-based retrieval approach.

Binary classification model: Train a binary MLP that, given a document and a tag, predicts a similarity score, then selects the top-k highest scoring tags.

2 Method

A fundamental component of our approach is the selection of a suitable encoding model to generate dense vector representations of documents and subject tags. We base our work on transformer-

[1] `https://sites.google.com/view/llms4subjects/home`

based architectures, starting from BERT (Devlin et al., 2018), which introduced bidirectional self-attention mechanisms to learn contextualized word representations. However, since BERT is originally designed for token-level tasks, it does not naturally provide fixed-size sentence embeddings.

To address this limitation, Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) was developed, modifying BERT by introducing a siamese network architecture to derive semantically meaningful sentence embeddings. SBERT significantly improves performance on similarity-based tasks by fine-tuning BERT with a contrastive objective.

Given the bilingual nature of the dataset, we focus on multilingual versions of SBERT (Reimers and Gurevych, 2020), which extend the original SBERT framework to support multiple languages by training on parallel and multilingual corpora. These models are specifically optimized to map semantically equivalent sentences across different languages into a shared embedding space, ensuring robust cross-lingual representations.

All the proposed approaches rely on an encoding model to generate dense vector representations of documents and subject tags. We experimented with four different architectures, each varying in model size, embedding dimensionality, and multilingual capabilities.

The **all-MiniLM-L6-v2**[2] model is a lightweight transformer with 22.7M parameters, producing 384-dimensional embeddings while maintaining computational efficiency.

The **distiluse-base-multilingual-cased-v1**[3] model, with 135M parameters, generates 512-dimensional embeddings and is optimized for multilingual sentence representation.

The **cross-en-de-roberta-sentence-transformer**[4] is a 278M parameter model designed specifically for cross-lingual applications, aligning semantically similar sentences across different languages within a shared vector space.

Finally, the **multilingual-e5-large**[5] model (Wang et al., 2024) is the most complex architecture, featuring 560M parameters and a 1024-dimensional embedding space, offering robust performance for multilingual text processing.

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[3] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1

[4] https://huggingface.co/T-Systems-onsite/cross-en-de-roberta-sentence-transformer

[5] https://huggingface.co/intfloat/multilingual-e5-large

2.1 Embedding-based retrieval

Sentence embeddings encode the semantic meaning and relationships between sentences, ensuring that semantically similar sentences have closely related representations. By leveraging this property of sentence encoders, we can identify tags that are most relevant to the document we want to annotate. To measure the similarity between embeddings, we use cosine similarity, defined as:

$$cossim(\mathbf{A}, \mathbf{B}) = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \quad (1)$$

where $\mathbf{A} \cdot \mathbf{B}$ is the *dot product* of the two vectors, $\|\mathbf{A}\|$ is the *Euclidean norm* (magnitude) of vector $\mathbf{A}$, $\|\mathbf{B}\|$ is the *Euclidean norm* (magnitude) of vector $\mathbf{B}$, θ is the *angle* between the two vectors. Finally, we assign the document the top-k tags with the highest similarity scores.

2.2 Fine-tuned embedding model

To perform fine-tuning, we created a dataset structured around three key elements. The first is the **anchor**, which contains the text of the document. The second is the **positive**, which includes the correct tag associated with the document. Finally, the third element is the **negative**, which contains an incorrect tag. This structure allows us to refine the model, helping it better distinguish between appropriate and inappropriate tags.

We used different losses and compared them in order to see the best performing ones.

TripletLoss (Schroff et al., 2015): Minimizes the distance between the anchor and positive while maximizing the distance between the anchor and negative.

CosineSimilarityLoss: Computes the cosine similarity between the anchor and either the positive or negative sample. It requires a float label, where 0.0 represents a negative tag and 1.0 represents a positive tag.

CoSENTLoss (Huang et al., 2024): A ranking-based loss function that optimizes sentence embeddings by ensuring consistency between training and prediction, aligning cosine similarity rankings with annotated similarity labels to improve semantic representation and model efficiency.

AnglELoss (Li and Li, 2024): A modification of CoSENTLoss designed to address the issue where the cosine function's gradient approaches zero near the peak or trough of its wave.

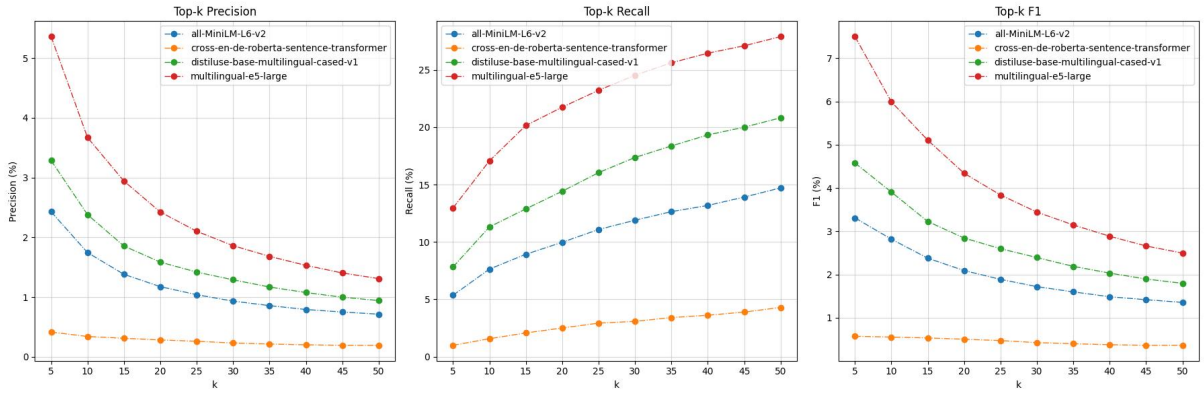

Figure 1: Comparison of different models in tagging using cosine similarity

MultipleNegativesRankingLoss (Henderson et al., 2017): A loss function designed for training sentence embeddings using only positive pairs, where each anchor-positive pair in a batch treats all other positives as negatives, optimizing embeddings for tasks like semantic search and paraphrase identification.

2.3 Binary classification model

We trained a Multi Layer Perceptron, starting from the document and tag embeddings. The goal is to generate a model capable of recognizing when a document is related to a given tag.

Thus, the training is performed on a binary task. We stacked the document embedding and the tag embedding in a single vector and passed it to the network. We created several pairs (*document embedding*, *tag embedding*), corresponding to both positive and negative cases, associated with positive or negative ground truth, respectively. The embeddings are created using a pretrained model as described in 2.1.

We designed the network by first doubling the dimensionality of the input features. These features are then projected onto 1024 neurons, followed by a single output neuron, as shown in Figure 2. To enhance the model's generalization, we incorporate dropout layers with a probability of 0.5 after each activation, except for the output layer. We initialized the network weights using Kaiming normal initialization to facilitate efficient training.

Loss. We used Binary Cross Entropy loss, to account for errors in binary classification:

$$f(z) = \log(\sigma(z)) \tag{2}$$

$$\mathcal{L} = -\sum_{i=1}^{N} [y_i f(\hat{y_i}) + (1 - y_i) f(1 - \hat{y_i})] \tag{3}$$

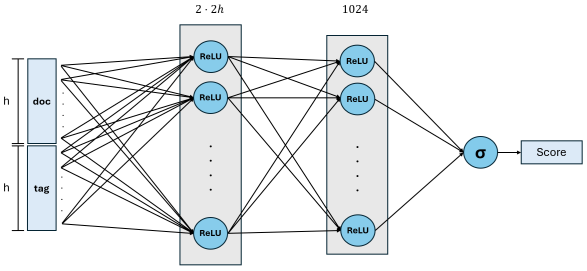

Figure 2: MLP architecture; h represent the hidden dimensionality of the embeddings.

where σ is the sigmoid function, $\hat{y_i}$ is the output of the model and y_i is the ground-truth.

We also added weight decay regularization, to mitigate overfitting.

3 Experimental results

In this section, we discuss the experiments conducted. Following the SemEval task guidelines, we evaluated our solution using top-k metrics, where $k \in \{5, 10, \ldots, 45, 50\}$, considering precision, recall, and F1 score. Due to the nature of these metrics, precision tends to decrease as k increases, while recall exhibits the opposite trend. For clarity, Tables [2, 3] present only the top-5 metrics, whereas the complete results are illustrated in Figures [1,3].

3.1 Performance

We computed the latency and resource consumption of the models, using the GPU NVIDIA T4 of Google Colab. As expected, *all-MiniLM-L6-v2* has the lowest latency, due to its smaller size. For each model, latency is measured by encoding one sentence at a time and iterating for 10,000 steps. Results can be found in Table 1.

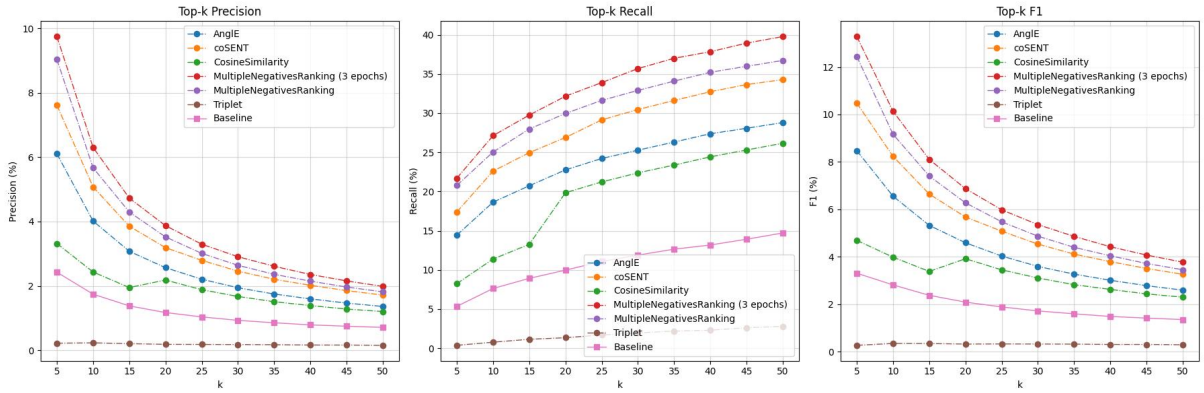

Figure 3: Finetuning of all-MiniLM-L6-v2 with different losses

Model	Latency (ms)	Parameters (M)
(1)	8	22.7
(2)	8	135
(3)	18	278
(4)	55	560

Table 1: Perfomances of the different models [(1): *all-MiniLM-L6-v2*, (2): *distiluse-base-multilingual-cased-v1*, (3): *cross-en-de-roberta-sentence-transformers*, (4): *multilingual-e5-large]*.

Model	Precision ↑	Recall ↑	F1 ↑
(1)	2.43%	5.35%	3.31%
(2)	3.29%	7.81%	4.58%
(3)	0.41%	0.98%	0.58%
(4)	**5.36%**	**12.96%**	**7.50%**

Table 2: top-5 metrics using Cosine Similarity with different models [(1): *all-MiniLM-L6-v2*, (2): *distiluse-base-multilingual-cased-v1*, (3): *cross-en-de-roberta-sentence-transformers*, (4): *multilingual-e5-large*].

3.2 Embedding-based retrieval

For this approach, we tested the four models in order to see the best performing ones. As expected, the models get better as they grow in size. The only exception is *cross-en-de-roberta-sentence-transformer* that actually performs worse than the smaller models. This could be due to the fact that *cross-en-de-roberta-sentence-transformer* is specifically tailored for cross-lingual sentence embeddings between English and German, but it might not generalize well to the specific domain of technical documents used in this task.

Results are shown in Figure 1 and Table 2.

Loss	Precision ↑	Recall ↑	F1 ↑
Triplet	0.22%	0.38%	0.27%
CosineSim	3.32%	8.24%	4.69%
AnglE	6.10%	14.44%	8.47%
CoSENT	7.61%	17.41%	10.48%
MultiNeg	**9.03%**	**20.78%**	**12.44%**
MultiNeg*	9.74%	21.70%	13.28%

Table 3: top-5 metrics for fine-tuning the model *all-MiniLM-L6-v2* for one epoch with different losses. *MultiNeg** is fine-tuned for 3 epochs.

3.3 Fine-tuned Embedding Models

Since fine-tuning is a time- and resource-intensive task, we selected only the smallest model, *all-MiniLM-L6-v2*, for experimentation.

As shown in Figure 3, *TripletLoss* leads to performance degradation due to the absence of *hard negatives* in our dataset. All other loss functions are effective, with *MultipleNegativesRankingLoss* performing the best due to its ability to leverage multiple negative samples, thereby improving the model's discriminative power.

The results in Table 3 indicate a significant improvement, even though the model was trained for just a single epoch. This suggests that more extensive fine-tuning could yield even better performance. So, we finetuned up to 3 epochs using the best loss (*MultipleNegativesRankingLoss*) observing a discrete improvement. However, as the primary objective of this study is to compare different approaches rather than to optimize a single model exhaustively, we opted not to invest additional resources into longer training.

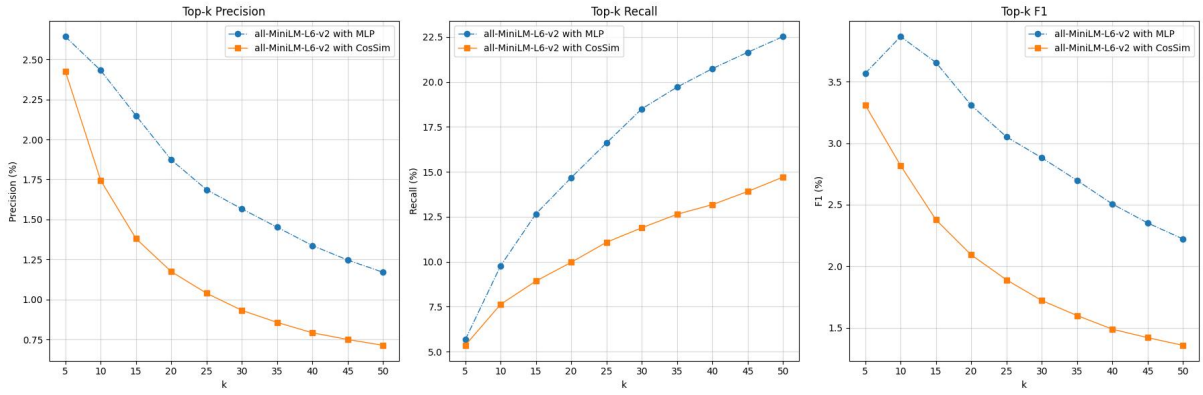

Figure 4: Comparison of quality metrics using MLP or Cosine Similarity

3.4 Binary classification model

As in fine-tuning, we decided to use *all-MiniLM-L6-v2* to generate embeddings for the binary classification task. After extensive hyperparameter tuning and architecture design, we achieved better results than those obtained by using cosine similarity. As shown in Figure 4, the MLP does not provide a significant improvement in the top-5 metrics. However, for $k \geq 10$, its performance follows a trend similar to that of the fine-tuned models, though with lower overall results. The training was carried out over 30 epochs.

4 Conclusions

All three approaches we tested proved to be successful. The best results were achieved with fine-tuning using *MultipleNegativesRankingLoss*, while the other two approaches yielded similar performance.

The MLP we designed does not achieve comparable results, but it improves performance with respect to using cosine similarity directly on embeddings.

Given the promising results from fine-tuning even the smallest model, we believe that applying the same fine-tuning techniques to larger models like *multilingual-e5-large* could further enhance performance. Future work could focus on extensive fine-tuning and hyperparameter optimization of these larger models, or on combining fine-tuned models with a binary classifier, to fully leverage their potential for the task of automated subject tagging in bilingual technical documents.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yunhsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply.

Xiang Huang, Hao Peng, Dongcheng Zou, Zhiwei Liu, Jianxin Li, Kay Liu, Jia Wu, Jianlin Su, and Philip S. Yu. 2024. Cosent: Consistent sentence embedding via similarity ranking. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 32:2800–2813.

Xianming Li and Jing Li. 2024. Angle-optimized text embeddings.

Nils Reimers and Iryna Gurevych. 2019. 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.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report.

A Loss plots

This appendix contains the training and validation loss plots obtained during the experiments.

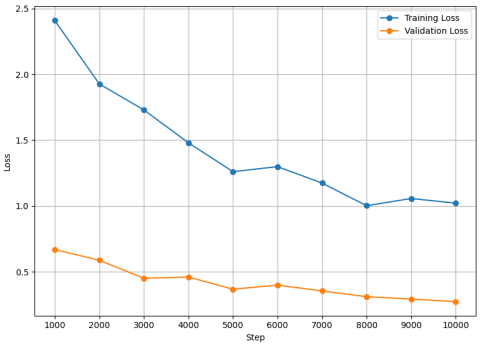

Figure 5: Training and Validation loss of fine-tuning using CoSENT loss

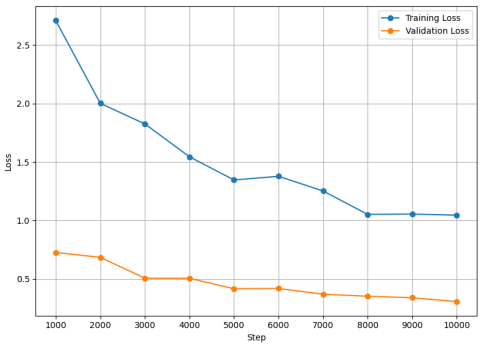

Figure 6: Training and Validation loss of fine-tuning using AnglE loss

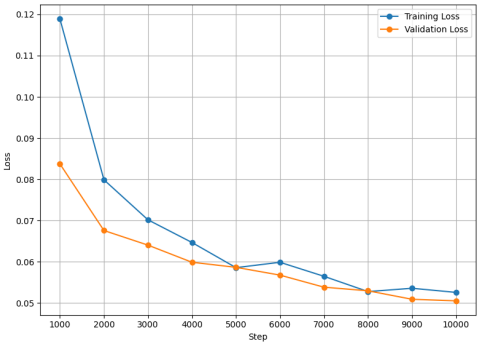

Figure 7: Training and Validation loss of fine-tuning using Cosine Similarity loss

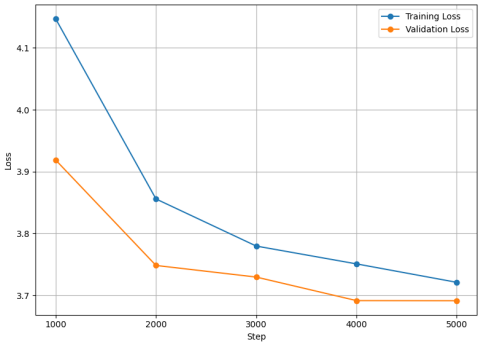

Figure 8: Training and Validation loss of fine-tuning using Triplet loss

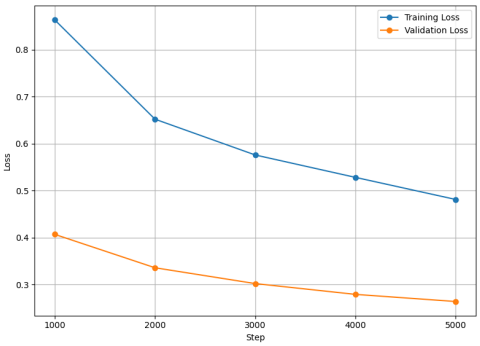

Figure 9: Training and Validation loss of fine-tuning using Multiple Negative Ranking loss

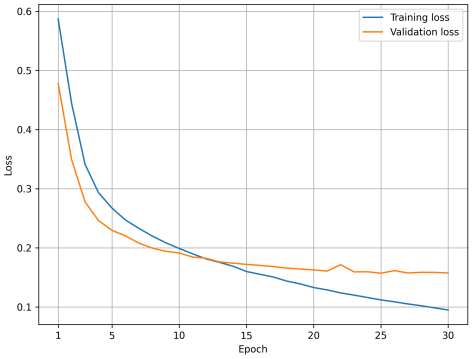

Figure 10: Training and Validation loss of Multi Layer Perceptron