silp_nlp at SemEval-2025 Task 5: Subject Recommendation With Sentence Transformer

Pankaj Kumar Goyal and Sumit Singh and Uma Shanker Tiwary
Indian Institute of Information Technology Allahabad, Allahabad
pankajgoyal02003,sumitrsch}@gmail.com
ust@iiita.ac.in

Abstract

Our team, *silp_nlp*, participated in the SemEval-2025 Task 5: LLMs4Subjects, which focuses on subject recommendation based on the given title and abstract. This task provided bilingual data in English and German for training and evaluation purposes. It consists of two data sets: one for all subjects and the other for technical subjects. We utilised statistical models, including TF-IDF and Sentence Transformers, to generate embeddings, and employed cosine similarity for recommendations. Our results show that JinaAi (sentence transformer) performed better than other sentence transformers and TF-IDF.

1 Introduction

The LLMs4Subjects shared task (D'Souza et al., 2025) aims to develop a subject recommendation system. The goal is to predict the most relevant subjects from the entire GND [1] subject collection to tag a given TIB technical record [2]. Each system will receive a technical record's title and abstract as input, and it must generate a customizable top-k list of relevant GND subjects. Since the input records may be in English or German, the systems should support bilingual semantic processing.

This task explores the potential of language model solutions for subject classification and tagging. It is based on the open-access TIB collection, specifically TIBKAT, which includes over 100,000 records such as technical reports, publications, and books, primarily in English and German. These records are classified according to the GND subjects taxonomy.

Leveraging statistical methods and transformer-based architectures for subject classification offers significant advantages. Semantic indexing with other vocabularies has gained attraction (Kazi et al., 2021; Wu et al., 2014). Notably, the prediction of Medical Subject Headings (MeSH) for biomedical literature has experienced significant advancements through the application of deep learning and machine learning techniques (Jin et al., 2018). In recent years, transformer-based models have demonstrated remarkable success across various Natural Language Processing (NLP) tasks. Among them, Sentence Transformers have attracted significant attention, particularly for tasks involving sentence similarity measurement. In this work, we employed Sentence Transformers to compute sentence similarity, leveraging cosine similarity as the distance metric. The datasets provided for the task encompass five distinct types of documents: articles, books, conference papers, reports, and theses.

2 Datasets

As part of the shared task, we were provided with datasets (D'Souza et al., 2024) containing two versions of the GND taxonomy:

GND Subjects - TIB Core: A focused subset containing subjects relevant to the core technical domains of TIB.

Full GND Subjects Collection: The complete set of GND subjects offers a broader classification range.

The total number of training, development, and testing samples for both the language and each document type is summarized in Tables 1 and 2, respectively, for both datasets (All Subjects and Tib-Core Subjects). Additionally, Tables 3 and 4 present the total number of unique subjects for each dataset.

3 Methodology

3.1 Embedding based cosine similarities

In this method, we converted all titles and abstracts into stored embeddings of the titles and abstracts using a sentence transformer (Reimers and Gurevych,

[1] GND
[2] TIB technical record

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 2455–2460

Dataset Split	Article (en)	Article (de)	Book (en)	Book (de)	Conference (en)	Conference (de)	Report (en)	Report (de)	Thesis (en)	Thesis (de)
Train	1042	6	26,966	33,401	3,619	2,210	1,275	1,507	3,452	8,459
Dev	173	1	4,482	5,589	601	371	215	256	574	1,404
Test	423	1	7,598	13,554	808	908	334	524	833	3,003
Total Records					**Train: 81,937**	**Dev: 13,666**	**Test: 27,986**			

Table 1: Statistics of the Dataset for Document Types across Train, Dev, and Test Splits (All Subjects)

Dataset Split	Article (en)	Article (de)	Book (en)	Book (de)	Conference (en)	Conference (de)	Report (en)	Report (de)	Thesis (en)	Thesis (de)
Train	253	5	17,669	12,528	2,840	717	896	761	2,506	3,727
Dev	42	1	2,944	2,088	473	119	149	126	417	621
Test	36	0	2,579	1,867	420	104	126	112	383	547
Total Records					**Train: 41,902**	**Dev: 6,980**	**Test: 6,174**			

Table 2: Statistics of the Dataset for Document Types across Train, Dev, and Test Splits (Tib-core Subjects)

Document Type	Language	Number of Subjects
Article	de	18
Article	en	157
Book	de	16,237
Book	en	16,647
Conference	de	3,215
Conference	en	3,788
Report	de	2,537
Report	en	2,405
Thesis	de	12,700
Thesis	en	7,377

Table 3: Number of Unique Subjects for Each Language and each Document Type (All Subjects)

Document Type	Language	Number of Subjects
Article	de	16
Article	en	69
Book	de	10,231
Book	en	12,434
Conference	de	1,544
Conference	en	2,895
Report	de	1,495
Report	en	1,825
Thesis	de	8,452
Thesis	en	5,554

Table 4: Number of Unique Subjects for Each Language and Each Document Type (tib-core-Subjects)

2019). During testing, the titles and abstracts of the test data were converted to embeddings. We determined the best matches on the basis of the cosine similarities between the embeddings of the test data and those of the training data. We converted the titles and abstracts into embeddings using two Sentence Transformer models. Furthermore, we conducted experiments using TF-IDF embeddings for comparison.

JinaAi/jina-embeddings-v3 (Sturua et al., 2024) JinaAi's jina-embeddings-v3 is a multilingual text embedding model supporting 89 languages, designed to process up to 8,192 input tokens and produce 1,024-dimensional embeddings. Based on a pre-trained XLM-RoBERTa (Conneau et al., 2020) with 559 million parameters, it incorporates five LoRA adapters (Hu et al., 2021) tailored for specific tasks: retrieval (separate adapters for queries and documents), clustering, similarity assessment, and classification. The model employs Matryoshka representation learning (Kusupati et al., 2024), which allows control over embedding dimensions with minimal performance loss.

distiluse-base-multilingual-cased-v2 (Reimers and Gurevych, 2020) The distiluse-base-multilingual-cased-v2 model is a multilingual sentence embedding model developed by the Sentence-Transformers team. It encodes sentences and paragraphs into a 512-dimensional dense vector space, enabling tasks such as clustering and semantic search across more than 50 languages. The model is optimized for efficient processing, with a maximum sequence length of 128 tokens.

In embeddings-based top-50 classification for subject indexing, this model can encode texts into 512-dimensional embeddings that capture semantic

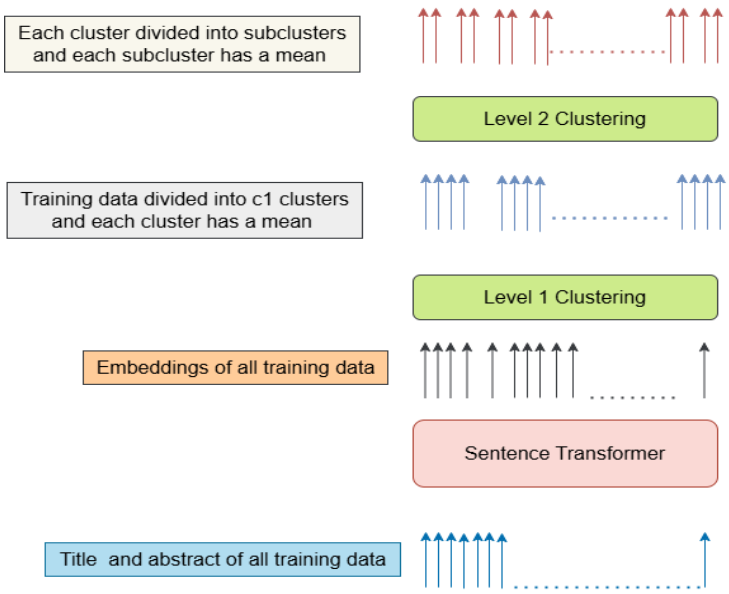

Figure 1: The architecture of hierarchical clustering consists of a two-level clustering framework, where each cluster at every level is represented by a mean vector. During testing, the matching process at each level is performed by comparing input representations with the corresponding mean vectors using cosine similarity.

nuances relevant to classification tasks.

Document Type	Lang-uage	Number of Clusters
Article	de	-
Article	en	-
Book	de	[20, 40,60]
Book	en	[20,40,60]
Conference	de	[5,10,20]
Conference	en	[5,10,20]
Report	de	[5,10,20]
Report	en	[5,10,20]
Thesis	de	[20,40,60]
Thesis	en	[10,20,40]

Table 5: Number of clusters for each language and document type: the first element represents the number of clusters at the last level, while the last element represents the number of clusters at the first level..

Document Type	Lang-uage	Number of Clusters
Article	de	-
Article	en	-
Book	de	[12, 24,48]
Book	en	[20,40,60]
Conference	de	[2,5,10]
Conference	en	[5,10,20]
Report	de	[2,5, 10]
Report	en	[2,5,10]
Thesis	de	[10,20,40]
Thesis	en	[8,16,24]

Table 6: Number of clusters for each language and document type: the first element represents the number of clusters at the last level, while the last element represents the number of clusters at the first level..

3.2 Hierarchical Clustering

In this hierarchical framework (Kavyasrujana and Rao, 2015), the dataset comprised approximately 16,000 subjects across nearly all data types. Comparing the similarity of all 16,000 subjects with an article by extracting embeddings can be inefficient and may result in suboptimal performance. To address this, we divided the data into three levels of clusters, with the number of clusters varying at each level. Our objective was to reduce the number of subjects for comparison, focusing on filtering out the most relevant ones rather than comparing the article with all subjects. Architecture of a two-level cluster given in Fig. 1.

First, we extracted embeddings for all subjects in each dataset using the JinaAI embedding model, a specialized multilingual model for English and Ger-

	Record Type	Language			
			Precision	Recall	F1
K@5	Article	de	0.0000	0.0000	0.0000
	Article	en	0.2203	0.5016	0.3062
	Book	de	0.0000	0.0000	0.0000
	Book	en	0.0380	0.0838	0.0523
	Conference	de	0.1604	0.2605	0.1985
	Conference	en	0.1443	0.2982	0.1945
	Report	de	0.1313	0.2830	0.1794
	Report	en	0.1060	0.2015	0.1389
	Thesis	de	0.1598	0.2186	0.1846
	Thesis	en	0.1311	0.1949	0.1567

Table 7: Results of both languages at all document types of our best model(JinaAi) at top-5 level for all-subjects dataset.. The top-5 level is also given for all three metrics (precision, recall, and f1) for the overall results of each model.

	Record Type	Language			
			Precision	Recall	F1
K@5	Article	en	0.2500	0.3278	0.2837
	Book	de	0.1853	0.3928	0.2518
	Book	en	0.1573	0.3656	0.2199
	Conference	de	0.1827	0.2703	0.2180
	Conference	en	0.1648	0.3500	0.2240
	Report	de	0.1554	0.3646	0.2179
	Report	en	0.1222	0.2084	0.1541
	Thesis	de	0.1481	0.1823	0.1634
	Thesis	en	0.1337	0.1914	0.1574

Table 8: Results of both languages at all document types of our best model(JinaAi) at top-5 level for tib-subjects dataset.. The top-5 level is also given for all three metrics (precision, recall, and f1) for the overall results of each model.

man. After obtaining the embeddings, we clustered the subjects at each level, with each subsequent level containing half the number of subjects as the previous one. Cluster embeddings were calculated by averaging the embeddings of all subjects within each cluster.

After clustering, we compared the article's embeddings (derived from both the text and abstract) with the embeddings of one relevant cluster at each level, except at the final level, where we adopted a different approach. At the preceding level, we identified the most relevant cluster. If that cluster contained more than 50 subjects, we compared the article's embeddings with the embeddings of each subject within that cluster and selected the top 50 subjects based on similarity. If the cluster had fewer than 50 subjects, we included an additional cluster—the second most similar to the article—ensuring that the total number of subjects compared exceeded 50.

We repeated this process using the Distiluse-base-multilingual model as well.

Table 8 displays the clusters for the TIB dataset, while Table 7 shows the clusters for the complete subjects dataset. We have also included a table outlining the number of clusters utilized at each level for each dataset.

4 Results & Analysis

Table 9 presents the overall average results for each model across different levels. Additionally, the average results for each level are summarized at the end of the table. The JinaAi sentence transformer model outperforms the other models based on cosine similarities. JinaAi sentence transformer model was pretrained for the English and German languages; therefore, it showed better results.

Table 7 displays the top-5 results for each language and document type across all subjects in the dataset, highlighting the superior performance of the JinaAi sentence transformer model compared to the other models.

Similarly, Table 8 shows the top-5 results for each language and document type for the TIB subjects dataset, also demonstrating that the JinaAi sentence transformer model outperforms all other remaining models.

5 Conclusion

This work explored subject recommendation using sentence transformers within the SemEval-2025 Task 5 (LLMs4Subjects) challenge. Our approach leveraged embedding-based cosine similarity and hierarchical clustering to predict relevant GND subjects for TIB technical records in English and German. By experimenting with different models, including JinaAi, Distiluse-base-multilingual, and TF-IDF, we found that the JinaAi sentence transformer consistently outperformed other methods in terms of precision, recall, and F1-score.

Our results highlight the effectiveness of transformer-based embeddings in semantic similarity tasks for subject classification. Additionally, hierarchical clustering helped reduce computational complexity by narrowing down candidate subjects efficiently. Despite the improvements, future work can focus on fine-tuning domain-specific embeddings, exploring knowledge graph integration, and enhancing multilingual capabilities for better generalization.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–

Record Type	all subject					tib-core-subject				
Models	distiluse-base-multilingua	Hierarchical Clustering (JinaAi)	Hierarchical Clustering (distiluse)	JinaAi	TF-IDF	distiluse-base-multilingua	Hierarchical Clustering (JinaAi)	Hierarchical Clustering (distiluse)	JinaAi	TF-IDF
precision_5	0.056	0.022	0.014	0.109	0.031	0.098	0.044	0.039	0.167	0.055
recall_5	0.108	0.045	0.025	0.204	0.055	0.172	0.08	0.071	0.295	0.104
f1_5	0.074	0.03	0.018	0.141	0.039	0.123	0.056	0.049	0.21	0.071
precision_10	0.039	0.015	0.008	0.07	0.019	0.067	0.032	0.023	0.111	0.036
recall_10	0.145	0.06	0.028	0.25	0.066	0.229	0.111	0.08	0.37	0.131
f1_10	0.061	0.024	0.013	0.108	0.029	0.103	0.049	0.035	0.169	0.055
precision_15	0.031	0.013	0.006	0.052	0.014	0.052	0.025	0.016	0.084	0.028
recall_15	0.168	0.073	0.03	0.275	0.072	0.256	0.132	0.084	0.409	0.146
f1_15	0.052	0.021	0.01	0.088	0.024	0.085	0.042	0.027	0.138	0.047
precision_20	0.025	0.011	0.005	0.042	0.011	0.043	0.021	0.013	0.069	0.024
recall_20	0.181	0.081	0.03	0.293	0.075	0.283	0.148	0.086	0.439	0.16
f1_20	0.044	0.019	0.008	0.074	0.019	0.075	0.037	0.022	0.118	0.041
precision_25	0.022	0.01	0.004	0.036	0.01	0.038	0.019	0.01	0.058	0.02
recall_25	0.193	0.09	0.031	0.306	0.08	0.307	0.163	0.088	0.461	0.168
f1_25	0.039	0.017	0.007	0.064	0.017	0.068	0.034	0.019	0.103	0.036
precision_30	0.019	0.009	0.003	0.031	0.009	0.034	0.017	0.009	0.051	0.017
recall_30	0.203	0.098	0.031	0.318	0.087	0.326	0.173	0.09	0.478	0.171
f1_30	0.035	0.016	0.006	0.057	0.016	0.062	0.031	0.016	0.092	0.031
precision_35	0.017	0.008	0.003	0.028	0.008	0.031	0.016	0.008	0.045	0.017
recall_35	0.212	0.104	0.032	0.327	0.095	0.341	0.184	0.091	0.492	0.191
f1_35	0.032	0.015	0.005	0.051	0.015	0.057	0.029	0.014	0.083	0.032
precision_40	0.016	0.008	0.002	0.025	0.007	0.029	0.015	0.007	0.041	0.016
recall_40	0.223	0.11	0.033	0.333	0.099	0.357	0.194	0.092	0.503	0.195
f1_40	0.03	0.014	0.005	0.046	0.013	0.053	0.028	0.013	0.075	0.029
precision_45	0.015	0.007	0.002	0.022	0.007	0.026	0.014	0.006	0.037	0.015
recall_45	0.23	0.115	0.033	0.338	0.107	0.37	0.202	0.093	0.511	0.208
f1_45	0.027	0.013	0.004	0.042	0.013	0.049	0.026	0.012	0.069	0.028
precision_50	0.013	0.007	0.002	0.021	0.006	0.024	0.013	0.006	0.034	0.015
recall_50	0.237	0.12	0.033	0.344	0.112	0.38	0.209	0.094	0.519	0.219
f1_50	0.025	0.013	0.004	0.039	0.012	0.046	0.024	0.011	0.063	0.027
Ave_precision	0.025	0.011	0.005	0.044	0.012	0.044	0.022	0.014	0.07	0.024
Ave_recall	0.19	0.09	0.031	0.299	0.085	0.302	0.16	0.087	0.448	0.169
Ave_f1	0.042	0.018	0.008	0.071	0.02	0.072	0.036	0.022	0.112	0.04

Table 9: Average Results of both languages at all document types of our all models at each level. The average of each level is also given for all three metrics (precision, recall, and f1) for the overall results of each model.

8451, Online. Association for Computational Linguistics.

Jennifer D'Souza, Sameer Sadruddin, Holger Israel, Mathias Begoin, and Diana Slawig. 2024. The semeval 2025 llms4subjects shared task dataset.

Jennifer D'Souza, Sameer Sadruddin, Holger Israel, Mathias Begoin, and Diana Slawig. 2025. Semeval-2025 task 5: Llms4subjects - llm-based automated subject tagging for a national technical library's open-access catalog. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1082–1095, Vienna, Austria. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. 2018. AttentionMeSH: Simple, effective and interpretable automatic MeSH indexer. In *Proceedings of the 6th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering*, pages 47–56, Brussels, Belgium. Association for Computational Linguistics.

D. Kavyasrujana and B. Chakradhara Rao. 2015. Hierarchical clustering for sentence extraction using cosine similarity measure. In *Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1*, pages 185–191, Cham. Springer International Publishing.

Nazmul Kazi, Nathaniel Lane, and Indika Kahanda. 2021. Automatically cataloging scholarly articles using library of congress subject headings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 43–49, Online. Association for Computational Linguistics.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2024. Matryoshka representation learning. *Preprint*, arXiv:2205.13147.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *Preprint*, arXiv:2409.10173.

Hao Wu, Martin Renqiang Min, and Bing Bai. 2014. Deep semantic embedding. In *SMIR@SIGIR*.