# NBF at SemEval-2025 Task 5: Light-Burst Attention Enhanced System for Multilingual Subject Recommendation

**Baharul Islam**[*], **Nasim Ahmad**[†], **Ferdous Ahmed Barbhuiya**[*], **Kuntal Dey**[*]
[*]IIIT Guwahati [†]Dibrugarh University
{baharul.islam22b,ferdous,kuntal.dey}@iiitg.ac.in, nasimahmad4117@gmail.com

## Abstract

We present our system submission for SemEval 2025 Task 5, which focuses on cross-lingual subject classification in the English and German academic domains. Our approach leverages bilingual data during training, employing negative sampling and a margin-based retrieval objective. We demonstrate that a dimension-as-token self-attention mechanism designed with significantly reduced internal dimensions can effectively encode sentence embeddings for subject retrieval. In quantitative evaluation, our system achieved an average recall rate of 32.24% in the general quantitative setting (all subjects), 43.16% and 31.53% of the general qualitative evaluation methods with minimal GPU usage, highlighting their competitive performance. Our results demonstrate that our approach is effective in capturing relevant subject information under resource constraints, although there is still room for improvement.

## 1 Introduction

Automated subject classification of scholarly articles is of growing importance for digital repositories, allowing more efficient data retrieval, recommendation, and systematic reviews (Devlin et al., 2019). At **SemEval-2025 Task 5** (D'Souza et al., 2025), participants must predict *subject codes* for articles in two languages (English and German).

Previous SemEval tasks on concept or subject classification have used both classical machine learning and deep approaches (Agirre et al., 2014). Large pre-trained models (e.g., GPT-based (Brown et al., 2020) or other generative approaches) could be powerful but are often computationally heavy. We instead adopt a *dimension-as-token* approach, refining base embeddings from Sentence Transformers (Reimers and Gurevych, 2019) via an ultra-light attention transform.

By participating in **SemEval-2025 Task 5**, we discovered that our parameter-efficient, dimension-as-token self-attention approach can effectively handle both English and German data, achieving moderate recall rates of 32. 24% in the overall quantitative setting (all subjects), 43. 16% in Case 1 and 31. 53% in Case 2 of Overall Qualitative evaluation methods with minimal GPU usage. Despite challenges such as near-synonymous subject labels and sparse domain terms, our system still secured the **9th** position in both quantitative (all subjects) and qualitative evaluations. These findings highlight the potential of cross-lingual pipelines and public code for fragmented training and inference, even under resource-constrained conditions.

## 2 Task Description

The LLMs4Subjects shared task (D'Souza et al., 2025) challenges participants to develop LLM-based systems for automated subject tagging in a national technical library. In this task, systems must assign one or more subject codes to each article, drawing from an extensive GND (German National Library, 2025) subjects taxonomy. Participants are provided with a human-readable version of the taxonomy along with a bilingual data set (English / German) from TIBKAT (TIB - Leibniz Information Centre for Science and Technology, 2025) that includes various record types such as `article`, `book`, `conference`, `report`, and `thesis`.

### 2.1 Dataset Description

The **LLMs4Subjects** corpus combines two complementary resources that are released together for SemEval 2025 Task 5:

a) **GND subject taxonomy:** machine-readable export of more than 200000 controlled subject descriptors[1]. Each entry provides a persistent identifier, the preferred German label, optional alternative labels, and explicit *broader*, *narrower*, and *related* relations.

[1] 'GND Sachbegriff' in the German National Library authority file

*Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 953–958
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

b) **TIBKAT technical records:** 123589 bibliographic records (title + abstract) drawn from the open access catalog of the German National Library of Science and Technology. The corpus is bilingual (52 % German, 48 % English) and covers five record types: *article*, *book*, *conference*, *report*, and *thesis*. We follow the official split with 81937 documents for training, 13666 for development, and 27986 for blind testing; gold GND labels are provided for the first two splits only.

## 3 Proposed System

Our method has two well-defined stages. First, we create fixed embeddings for every data. Second, we train a small neural module that makes the embeddings of the articles match the embeddings of the GND subjects.

All subject labels (originally in German) are translated into English. We then use pre-trained Sentence-Transformer models to encode both subjects and articles. For articles, we join the title and abstract before encoding. Each subject and each article are now represented by a single 768-dimensional vector. These embeddings serve as input to our high-capacity transformation model, which is designed to enhance the semantic alignment between article embeddings and subject embeddings.

- **Dimension-as-Token Projection and Reshaping:** We treat every number in the 768-long vector as its own 'token'. A lightweight Burst-Attention layer with 16 hidden units and two heads lets these tokens exchange information.

- **Feed-Forward MLP:** The attention output passes through a three-layer feed-forward network with dropout. The result is the final vector of aligned articles.

Together, these components transform initial embeddings into a space where articles are more closely aligned with their relevant subjects, thus improving the effectiveness of subject retrieval.

**Training procedure.** During training only article vectors go through the alignment module(transformation module). For each article, we also retrieve its gold subject vectors and average them. The loss function moves the aligned article vector closer to this average and further away from a few randomly chosen, incorrect subject vectors. The subject vectors themselves remain fixed.

**Inference procedure.** At test time the alignment module is frozen. A new article is embedded, passed through the module, and compared using cosine similarity to the stored subject vectors. The $k$ subjects with the smallest distance are returned as recommendations.

**Design motivation.** Leaving the large pretrained transformer models untouched avoids costly fine-tuning and keeps subject and article vectors consistent. The tiny alignment module is enough to correct systematic differences between the two sets of vectors while adding very little computation.

Figure 1 illustrates the complete system architecture. It begins by extracting embeddings for both the subject corpus and article texts, followed by a training module that applies the transformation block and computes a margin-based loss. After training, the resulting model is saved for inference, where articles are matched against subject embeddings to retrieve the most relevant subjects.

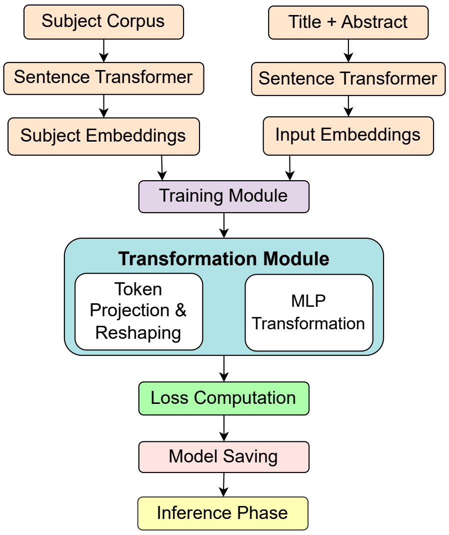

Figure 1: System Architecture. The model processes subject and article embeddings through a transformation module (Token Projection & Reshaping + MLP) and optimizes via margin-based loss

### 3.1 Data Preprocessing

In this stage, we convert all subject names and article texts into fixed-size vectors using off-the-shelf sentence-transformer models. These vectors form the inputs to our alignment module.

**Subject corpus** We begin with the GND subject list. Each entry has a unique *code* and a

German-language *Name* (Subject Name). We translate every name into English using the Helsinki-NLP/opus-mt-de-en model (Tiedemann and Thottingal, 2020). Next, we embed each translated name with a Sentence-Transformer. Concretely, for the $i$th subject name $n_i$ we compute

$$\mathbf{s}_i = ST(n_i) \in \mathbb{R}^d, \quad d = 768. \tag{1}$$

Collecting all $N$ subject vectors yields the subject matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_N \end{bmatrix} \in \mathbb{R}^{N \times d}. \tag{2}$$

**Article corpus** For the training data, each sample comprises a *title*, an *abstract*, and a set of gold-standard (correct) subject labels. We concatenate the title $t_j$ and abstract $a_j$ into a single text $x_j$, and compute its embedding as

$$\mathbf{m}_j = ST(x_j) \in \mathbb{R}^d. \tag{3}$$

For each training sample $j$, every gold subject name $s_{jl}$ (with $l = 1, 2, \ldots, k_j$) is embedded, and the resulting embeddings are averaged to obtain a single representative embedding:

$$\mathbf{g}_j = \frac{1}{k_j} \sum_{l=1}^{k_j} ST(s_{jl}) \in \mathbb{R}^d. \tag{4}$$

Thus, the training document embeddings and the corresponding gold subject embeddings are aggregated into the matrices:

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_M \end{bmatrix} \in \mathbb{R}^{M \times d}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_M \end{bmatrix} \in \mathbb{R}^{M \times d}. \tag{5}$$

**Embedding models** For English text we use **all-mpnet-base-v2** (Reimers and Gurevych, 2019). For German text we use **T-Systems-onsite/german-roberta-sentence-transformer-v2** (T-Systems-onsite, 2022). Both models produce 768-dimensional outputs.

### 3.2 Dimension-as-Token Projection and Reshaping

To enhance expressiveness, we reinterpret each dimension of an article embedding $\mathbf{m}$ as an individual "token". Starting with an embedding $\mathbf{m}$ of shape (batch_size $B$, $d$), we first reshape it to a tensor of shape ($d$, $B$, 1). This operation enables us to treat each scalar dimension as a separate token, which is then projected to a hidden representation of size 16 (denoted as *model_dim*). Next, we apply a Burst Attention (Sun et al., 2024) encoder across these $d$ tokens to capture their interdependencies. Finally, the tokens are projected back to a single scalar and reshaped to recover an embedding of the original dimension $d$. The forward pass is expressed as:

$$\begin{aligned} \tilde{\mathbf{x}} &= \text{Reshape}(\mathbf{m}) \in \mathbb{R}^{d \times B \times 1}, \\ \mathbf{X}_0 &= \text{Linear}_{\text{in}}(\tilde{\mathbf{x}}) \in \mathbb{R}^{d \times B \times \text{model\_dim}}, \\ \mathbf{X}_L &= \text{BurstAttnEnc}(\mathbf{X}_0), \\ \mathbf{X}_{\text{out}} &= \text{Linear}_{\text{out}}(\mathbf{X}_L) \in \mathbb{R}^{d \times 1}, \\ \mathbf{x}_{\text{attn}} &= \text{Reshape}^{-1}(\mathbf{X}_{\text{out}}) \in \mathbb{R}^d. \end{aligned} \tag{6}$$

Within the *BurstAttnEnc*, a Burst Attention(Sun et al., 2024) layer is used. Initially, layer normalization is applied to the input, after which multi-head self-attention is computed using 2 attention heads, each with a per-head dimension of 8. A dropout of 0.03 is applied to the attention output before it is passed to a feedforward network. This network expands the hidden representation to 64 dimensions, applies a ReLU activation, and includes another dropout of 0.03, with residual connections added throughout to ensure stable learning. The Burst Attention encoder is defined as:

$$\begin{aligned} \mathbf{X}^{(0)} &= \mathbf{X}_0, \\ \mathbf{Z}^{(l-1)} &= \text{LayerNorm}(\mathbf{X}^{(l-1)}), \\ \tilde{\mathbf{X}}_{\text{attn}}^{(l)} &= \text{MultiHead}(\mathbf{Z}^{(l-1)}), \\ \mathbf{Y}^{(l)} &= \mathbf{X}^{(l-1)} + \text{Dropout}(\tilde{\mathbf{X}}_{\text{attn}}^{(l)}), \\ \mathbf{U}^{(l)} &= \text{LayerNorm}(\mathbf{Y}^{(l)}), \\ \tilde{\mathbf{X}}_{\text{ff}}^{(l)} &= W_2 \cdot \text{ReLU}(W_1 \cdot \mathbf{U}^{(l)} + b_1) + b_2, \\ \mathbf{X}^{(l)} &= \mathbf{Y}^{(l)} + \text{Dropout}(\tilde{\mathbf{X}}_{\text{ff}}^{(l)}). \end{aligned} \tag{7}$$

Here, the weight matrices $W_1 \in \mathbb{R}^{16 \times 64}$ and $W_2 \in \mathbb{R}^{64 \times 16}$ are learnable parameters within the feed-forward sub-module.

### 3.3 Feed-Forward MLP

Subsequent to obtaining the intermediate representation $\mathbf{x}_{\text{attn}}$ from the burst-attention block, we further refine the embedding using a Feed-Forward Multi-Layer Perceptron (MLP). This MLP is composed of three linear layers with ReLU activations and incorporates a dropout of 0.03 at each stage to reduce the risk of overfitting. The purpose of the MLP is to transform the intermediate representation back into the original embedding space, thus producing the final transformed embedding $\mathbf{z}$. This process is mathematically described as:

$$\begin{aligned} \mathbf{h}_1 &= \mathrm{ReLU}(\mathrm{Dropout}(W_1^{\mathrm{MLP}}\mathbf{x}_{\mathrm{attn}} + b_1^{\mathrm{MLP}})), \\ \mathbf{h}_2 &= \mathrm{ReLU}(\mathrm{Dropout}(W_2^{\mathrm{MLP}}\mathbf{h}_1 + b_2^{\mathrm{MLP}})), \\ \mathbf{z} &= W_3^{\mathrm{MLP}}\mathbf{h}_2 + b_3^{\mathrm{MLP}}. \end{aligned} \tag{8}$$

In this formulation, $\mathbf{h}_1$ and $\mathbf{h}_2$ represent the outputs of the first and second hidden layers, respectively, and $\mathbf{z}$ is the output of the final layer. The weight matrices $W_1^{\mathrm{MLP}} \in \mathbb{R}^{d\times 256}$, $W_2^{\mathrm{MLP}} \in \mathbb{R}^{256\times 256}$, and $W_3^{\mathrm{MLP}} \in \mathbb{R}^{256\times d}$, along with their corresponding biases $b_1^{\mathrm{MLP}}$, $b_2^{\mathrm{MLP}}$, and $b_3^{\mathrm{MLP}}$, are learned during training.

Finally, the overall transformation function, denoted as $\mathrm{transform}()$, is defined as the composition of the above operations, and for an input $\mathbf{m}$ it is computed as:

$$\begin{aligned} \mathrm{transform}(\mathbf{m}) = \mathrm{MLP}\Big(&\mathrm{BurstAttnEnc}\big( \\ &\mathrm{Linear}_{\mathrm{in}}(\mathrm{Reshape}(\mathbf{m}))\big)\Big), \end{aligned} \tag{9}$$

which outputs $\mathbf{z} \in \mathbb{R}^d$. This architecture is designed to capture the complex interdimensional relationships present in the initial embeddings, ultimately yielding a more robust and semantically aligned representation.

## 4 Training and Inference

During training, we learn only the small transformation block, keeping the Sentence-Transformer weights and the subject matrix $\mathbf{S}$ fixed. For each article $j$, with its raw embedding $\mathbf{m}_j$ we apply the learned transform to obtain the anchor embedding $\mathbf{a}_j = \mathrm{transform}(\mathbf{m}_j) \in \mathbb{R}^d$. Let the averaged gold-subject embedding be $\mathbf{p}_j$, and let $\mathbf{n}_{j1}, \ldots, \mathbf{n}_{jk}$ denote $k$ negative subject embeddings sampled from $\mathbf{S}$. Using cosine distance $d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|}$ , we define the margin-based loss for article $j$ as

$$\mathcal{L}_j = \sum_{i=1}^{k} \max\Big\{0,\ \alpha + d(\mathbf{a}_j, \mathbf{p}_j) - d(\mathbf{a}_j, \mathbf{n}_{ji})\Big\}, \tag{10}$$

$\alpha = 0.2$, Minimizing $\mathcal{L}_j$ brings the transformed article closer to its correct subjects while pushing it away from the incorrect ones.

We train for 20 epochs with batch size 4, using the AdamW optimizer and a cosine annealing scheduler. In each batch, only the embeddings of the article, their positives and the sampled negatives are passed through $\mathrm{transform}(\cdot)$; the embeddings of the subject remain frozen.

**Inference** At inference time, we first transform and cache all GND subject embeddings as $\mathbf{S}' = \mathrm{transform}(\mathbf{S})$, so that each row $\mathbf{s}'$ is the aligned vector for one subject. For a new article (title and abstract), we compute its embedding $\mathbf{m} = ST(\text{title}\|\text{abstract})$ using a Sentence-Transformer, then apply the trained mapping to obtain $\mathbf{a} = \mathrm{transform}(\mathbf{m})$. Next, we measure the cosine-distance between $\mathbf{a}$ and each precomputed subject vector $\mathbf{s}'$: $d(\mathbf{a}, \mathbf{s}') = 1 - \frac{\mathbf{a}\cdot\mathbf{s}'}{\|\mathbf{a}\|\,\|\mathbf{s}'\|} \quad \forall \mathbf{s}' \in \mathbf{S}'$. Finally, we sort all subjects by increasing distance and return the $k$ codes with the smallest values.

## 5 Experimental Setup

Our experiments were carried out on a bilingual data set partitioned into training, development and test sets for both English and German articles. The training set was used for margin-based optimization, the development set for intermediate performance monitoring, and the test set was reserved for official SemEval evaluation.

The embedding dimension is set to 768, and our model is configured with a hidden dimension (`model_dim`) of 16, 2 attention heads, a single Burst Attention layer, an MLP hidden dimension of 256, and a dropout rate of 0.03. The training employs a batch size of 4 and utilizes 15 negative samples per anchor-positive pair over 20 epochs.

We apply a margin-based loss with a margin of 0.2, optimizing the model using the AdamW optimizer with a learning rate of $1 \times 10^{-4}$ (without weight decay) and a cosine annealing learning rate scheduler ($T_{max} = 20$, $\eta_{\min} = 1 \times 10^{-6}$). All experiments were executed on Google Colab L4 GPUs under Python 3.9, using PyTorch 1.13.1, SentenceTransformers 2.2.2, pandas 1.5.3, numpy 1.23.5, and tqdm 4.64.1. Evaluation measures included Precision@k, Recall@k, and F1@k across multiple cutoffs, along with average recall, as reported in Section 6.

## 6 Results

We report the performance of our system on the official test segment as well as on two qualitative evaluation subsets. For evaluation, we use Precision @ k (P @ k), Recall @ k (R @ k), and F1 @ k. Precision @ k is defined as the number of relevant items in the top $k$ predictions divided by Total number of items in $k$, Recall @ k is defined as the number of relevant items in the top $k$ predictions divided by the total number of relevant items. The F$\beta$-score is computed as

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision @ k} \times \text{Recall @ k}}{\beta^2 \times \text{Precision @ k} + \text{Recall @ k}} \tag{11}$$

and in our experiments we set $\beta = 1$ to obtain the F1@k score.

**Quantitative Results** Evaluation 1 of the shared task compared systems in a fully automatic setting in the blind TIBKAT test corpus. Participants submitted the *top–50* GND subject codes for every English or German record, and the organisers calculated P @ k, R @ k, and F1 @ k for each $k \in \{5, 10, \ldots, 50\}$. Scores were released at three granularities: (i) language level (en vs. de), (ii) record–type level (five technical record types), and (iii) the combined language–record level. Systems were ranked by the *average Recall @ k* over all cut-offs, a criterion in which our **NBF** run placed **9th**.

**Qualitative Results** Evaluation 2 complemented the automatic leaderboard with a manual assessment by subject specialists from TIB. A stratified sample of 140 records—ten from each of fourteen discipline codes (arc, che, elt, fer, his, inf, lin, lit, mat, oek, phy, sow, tec, ver)—was drawn. For every record, librarians inspected the *top–20* subjects proposed by each system and labelled them `Y` (correct), `I` (irrelevant but technically admissible), or `N`/blank (incorrect). Two leaderboards were produced: **Case 1**, counting both `Y` and `I` as correct, and **Case 2**, counting only `Y`. These settings correspond exactly to Tables 2 and 3.

Table 1 summarizes the overall quantitative results on the all-subjects evaluation (official test set) at cutoffs $k \in \{5, 10, 15, \ldots, 50\}$. The model achieves an average recall of 32.24% across these cutoffs, although the precision remains relatively low. Based on these official metrics, our system ranked **9th** in the quantitative track.

To better understand our design choices, we conducted experiments on two different subsets of the test data. Table 2 shows the overall qualitative results for Case 1, a subset characterized by common and less ambiguous subject labels. On this subset, the model achieves higher Precision, Recall, and F1 (with averages of 39.54%, 43.16%, and 39.03%, respectively). In contrast, Table 3 presents overall Qualitative results for Case 2, which consists of articles with more specialized or challenging subject mappings, yielding lower average metrics (Precision = 19.00%, Recall = 31.53%, F1 = 22.40%). These analyses demonstrate that while our parameter-efficient approach is competitive overall, its performance is sensitive to the complexity of subject labels.

Table 1: All-subjects (Overall Quantitative) results on the official test set for Team NBF (best run).

| k | P@k | R@k | F1@k |
|---|---|---|---|
| 5 | 0.0835 | 0.1699 | 0.1120 |
| 10 | 0.0594 | 0.2329 | 0.0946 |
| 15 | 0.0475 | 0.2742 | 0.0809 |
| 20 | 0.0401 | 0.3048 | 0.0708 |
| 25 | 0.0351 | 0.3305 | 0.0635 |
| 30 | 0.0314 | 0.3515 | 0.0576 |
| 35 | 0.0285 | 0.3694 | 0.0529 |
| 40 | 0.0261 | 0.3839 | 0.0489 |
| 45 | 0.0241 | 0.3974 | 0.0455 |
| 50 | 0.0225 | 0.4095 | 0.0426 |
| **Average Recall = 0.3224** | | | |

Table 2: Case 1 Results (Overall Qualitative Evaluation: Common Subjects).

| k | P@k | R@k | F1@k |
|---|---|---|---|
| 5 | 0.4405 | 0.2068 | 0.2815 |
| 10 | 0.4099 | 0.3738 | 0.3910 |
| 15 | 0.3753 | 0.5097 | 0.4323 |
| 20 | 0.3559 | 0.6362 | 0.4565 |
| **Average** | **0.3954** | **0.4316** | **0.3903** |

Table 3: Case 2 Results (Overall Qualitative Evaluation: Specialized Subjects).

| k | P@k | R@k | F1@k |
|---|---|---|---|
| 5 | 0.2344 | 0.1718 | 0.1983 |
| 10 | 0.1956 | 0.2830 | 0.2313 |
| 15 | 0.1734 | 0.3666 | 0.2354 |
| 20 | 0.1565 | 0.4398 | 0.2309 |
| **Average** | **0.1900** | **0.3153** | **0.2240** |

## 6.1 Error Analysis

Although our system shows potential in assigning subject labels, we found two main sources of error. First, the system struggles with synonym overlaps. For instance, near-synonymous labels such as *"Natural Language Processing"* and *"Compu-*

*tational Linguistics"* often appear together in the training data without sufficient distinction, causing the model to confuse these overlapping concepts and misclassify the subjects.

Second, errors frequently occur with sparse domain terms. Articles addressing highly specialized or infrequent topics tend to be mislabeled because the negative sampling does not adequately cover these less common domains. These challenges suggest that future work should focus on enhancing the model's ability to differentiate between similar subject labels and on improving the representation of niche topics in the training process.

## 7 Conclusion and Future Work

We presented our system, which employs a novel dimension-as-token self-attention mechanism (*Burst Attention*) on top of Sentence Transformers. Our experiments demonstrate that, even with an ultra-light hidden dimension (16) and a single attention layer, the approach is competitive in terms of recall, achieving average recall rates of 32.24%, 43.16% and 31.53% in the Overall Quantitative, Qualitative Case 1, and Qualitative Case 2 evaluations, respectively. These results indicate that our model is effective at capturing relevant subject information under resource constraints, although there is still room for improvement.

In future work, we plan to explore deeper stacking of Burst Attention layers, more advanced negative sampling strategies, and the incorporation of hierarchical subject ontologies for improved synonym disambiguation. Additionally, we intend to investigate the integration of multimodal features to further enhance system performance. These enhancements aim to further improve both the accuracy and robustness of our subject classification system.

## Acknowledgments

We thank the organizers of the SemEval-2025 Task 5 (D'Souza et al., 2025) for the data and evaluation framework, as well as the PyTorch, Hugging Face and SentenceTransformers communities for their invaluable open-source tools.

## References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

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)*, Vienna, Austria. Association for Computational Linguistics.

German National Library. 2025. Integrated authority file (gnd). Accessed: 5 December, 2024.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ao Sun, Weilin Zhao, Xu Han, Cheng Yang, Zhiyuan Liu, Chuan Shi, and Maosong Sun. 2024. Burstattention: An efficient distributed attention framework for extremely long sequences. *arXiv preprint arXiv:2403.09347*.

T-Systems-onsite. 2022. German roberta sentence transformer v2. `https://huggingface.co/T-Systems-onsite/german-roberta-sentence-transformer-v2`. Accessed: December 5, 2024.

TIB - Leibniz Information Centre for Science and Technology. 2025. Tibkat - catalogue of the tib. Accessed: 5 December, 2024.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt–building open translation services for the world. In *Annual Conference of the European Association for Machine Translation*, pages 479–480. European Association for Machine Translation.