

Automated Collection of Evaluation Dataset for Semantic Search in Low-Resource Domain Language

Anastasia Zhukova¹, Christian E. Matt² and Bela Gipp¹

¹University of Göttingen, ²eschbach GmbH

Abstract

Domain-specific languages that use a lot of specific terminology often fall into the category of low-resource languages. Collecting test datasets in a narrow domain is time-consuming and requires skilled human resources with domain knowledge and training for the annotation task. This study addresses the challenge of automated collecting test datasets to evaluate semantic search in low-resource domain-specific German language of the process industry. Our approach proposes an end-to-end annotation pipeline for automated query generation to the score reassessment of query-document pairs. To overcome the lack of text encoders trained in the German chemistry domain, we explore a principle of an ensemble of "weak" text encoders trained on common knowledge datasets. We combine individual relevance scores from diverse models to retrieve document candidates and relevance scores generated by an LLM, aiming to achieve consensus on query-document alignment. Evaluation results demonstrate that the ensemble method significantly improves alignment with human-assigned relevance scores, outperforming individual models in both inter-coder agreement and accuracy metrics. These findings suggest that ensemble learning can effectively adapt semantic search systems for specialized, low-resource languages, offering a practical solution to resource limitations in domain-specific contexts.

1 Introduction

In NLP, a low-resource language lacks sufficient linguistic data, resources, or tools for effective model training and development (Hedderich et al., 2021; Chu and Wang, 2018). Domain-specific German, especially in areas with professional jargon, codes, acronyms, and numeric data, qualifies as a low-resource language because large, publicly accessible datasets for such specialized domains are scarce. As a result, few language models are

Time stamp	Functional locations	Product	Description
2021/08/01 10:04	Alpha-L1-R111- T5002 Tank 5002	ABC	Gesendet an HAH Transfer von B6 nach B1 98779 H2 Wasser nach B6 98781 H2 Organik bleibt bei SFP Wasser D.O. 2-1 .59 2-3 11.06 Kohlenstofftransfer zu K2 B4 32' B9 18' K2 20' Loto't BAC-Zulaufwasser

Figure 1: An example of a mocked text log from a shift book in the German language. The logs contain a log of domain-specific terms, which require domain knowledge in the area and know specifics of the production process

trained specifically for these areas. While general German has extensive NLP resources, specialized sublanguages often demand unique datasets that are difficult to gather and typically limited in volume.

Shift logs in the process industry are detailed records maintained by operators or technicians during their work shifts (see Figure 1¹). They document key operational activities, system statuses, production metrics, equipment performance, process parameters, maintenance activities, safety observations, product quality, and any incidents or anomalies. The process industry produces and transforms raw materials into finished products through chemical, physical, or biological processes. The complexity of parsing and interpreting professional terminology and industry-specific syntax requires models trained on annotated datasets tailored to the domain, which are often non-existent or proprietary. This lack of accessible, high-quality datasets makes it difficult to build, fine-tune, or adapt existing NLP models for these specialized uses. Without significant efforts in curating and labeling domain-specific data, language models will struggle with accurate interpretation and generation in these fields.

Collecting and annotating text collections for se-

¹The text in Figure 1 translates to English as "Sent to HAH Transfer B6 to B1 98779 H2 water to B6 98781 H2 organics still at SFP Water D.O. 2-1 .59 2-3 11.06 Carbon transfer to K2 B4 32' B9 18' K2 20' Loto'd BAC inlet water supply"

semantic search in low-resource languages presents several significant challenges. First, finding qualified annotators for this task who are both fluent in the language and trained in linguistic annotation can be extremely difficult. Moreover, the complexity of semantic search requires annotations beyond basic syntactic labeling, such as entity recognition and coreference resolution, which demand specialized knowledge and increase the task's difficulty. Second, standalone general language models trained on high-resource languages can collect the test data to a certain extent but do not transfer well to these low-resource contexts and lack accurate language representation of the domain language.

This paper explores the principle of ensemble learning to create test collections for semantic search in domain-specific German language. Ensemble learning is a machine learning technique that combines multiple individual models, often called "weak learners," to create a more powerful and accurate predictive model by mitigating each other's weaknesses (Mienye and Sun, 2022). Our experiments demonstrate that combining an ensemble of multiple encoders with a generative LLM (GPT-4o in our case) to reassess relevance scores significantly improves the quality of test collections for semantic search evaluation. Specifically, this approach increases inter-coder agreement (measured by Krippendorff's alpha) by nearly four times and improves the F1-score by 1.5 times.

2 Related work

Ensemble learning improves machine learning performance by combining predictions from multiple models, thus enhancing accuracy, reducing variance, and mitigating bias (Mienye and Sun, 2022). Ensemble learning is popular across domain-specific domains and applications, such as medical diagnosis and fraud detection. It has started evolving from being used with machine learning algorithms to deep learning models.

LLMs have already been widely used for data annotation, specifically for domain-specific tasks requiring specialized domain knowledge, where human annotations are costly but crucial (Tan et al., 2024). Multiple studies have evaluated LLMs in biomedicine (Zhu et al., 2023; Kumar et al., 2024), law and education (Zhu et al., 2023), and financial sector (Aguda et al., 2024). While LLMs are a powerful tool for data annotation, the studies show that standalone LLMs perform worse than human

annotators (Lu et al., 2023; Staff et al., 2023).

To mitigate the drawbacks of LLM annotations, new methods were proposed to involve reasoning, reevaluating the assigned labels, or involving collective decisions. One of the state-of-the-art techniques is to use a human-in-loop annotation process and help human annotators by augmenting them with the fast LLM-pre-annotated labels (Li et al., 2023). The most recent development employs an ensemble of LLMs for annotation (Farr et al., 2024) or utilizes a synergy of thoughts across multiple smaller-scale LMs (Shang et al., 2024), similar to ensemble learning with "weak" models.

3 Methodology

Ensemble learning is widely used in practice because it can improve model robustness and accuracy and reduce variance, especially when individual models are prone to errors or have high variability. The central idea is that by aggregating the predictions of several models, the ensemble can outperform any single model, reducing the risk of overfitting and improving generalization. Ensemble methods leverage the strengths of different models while compensating for their weaknesses, leading to better performance on complex tasks (Mienye and Sun, 2022). In stacking of ensemble learning, different models (often of different types) are trained, and their predictions are used as input to a "meta-model," which learns how to combine these predictions to make the final decision.

The methodology of the ensemble for annotating a test collection for semantic search comprises two main parts: (1) document indexing and (2) creation of the query-document pairs. The key aspect of document indexing is using a set of encoders with various architectures and training strategies. The goal is to combine different aspects of the document similarity that each encoder has learned. Re-ranking combines the relevance score based on the document similarity with the score generated by a generative LLM. LLM assesses the relevance of the query-document pair independently from the score used for the retrieval, thus allowing the combining of another "point of view" to the query-document relevance. Figure 2 depicts the proposed methodology.

3.1 Database indexing

Multiple encoders are used for the database indexing. Possible ways to encode a text document in-

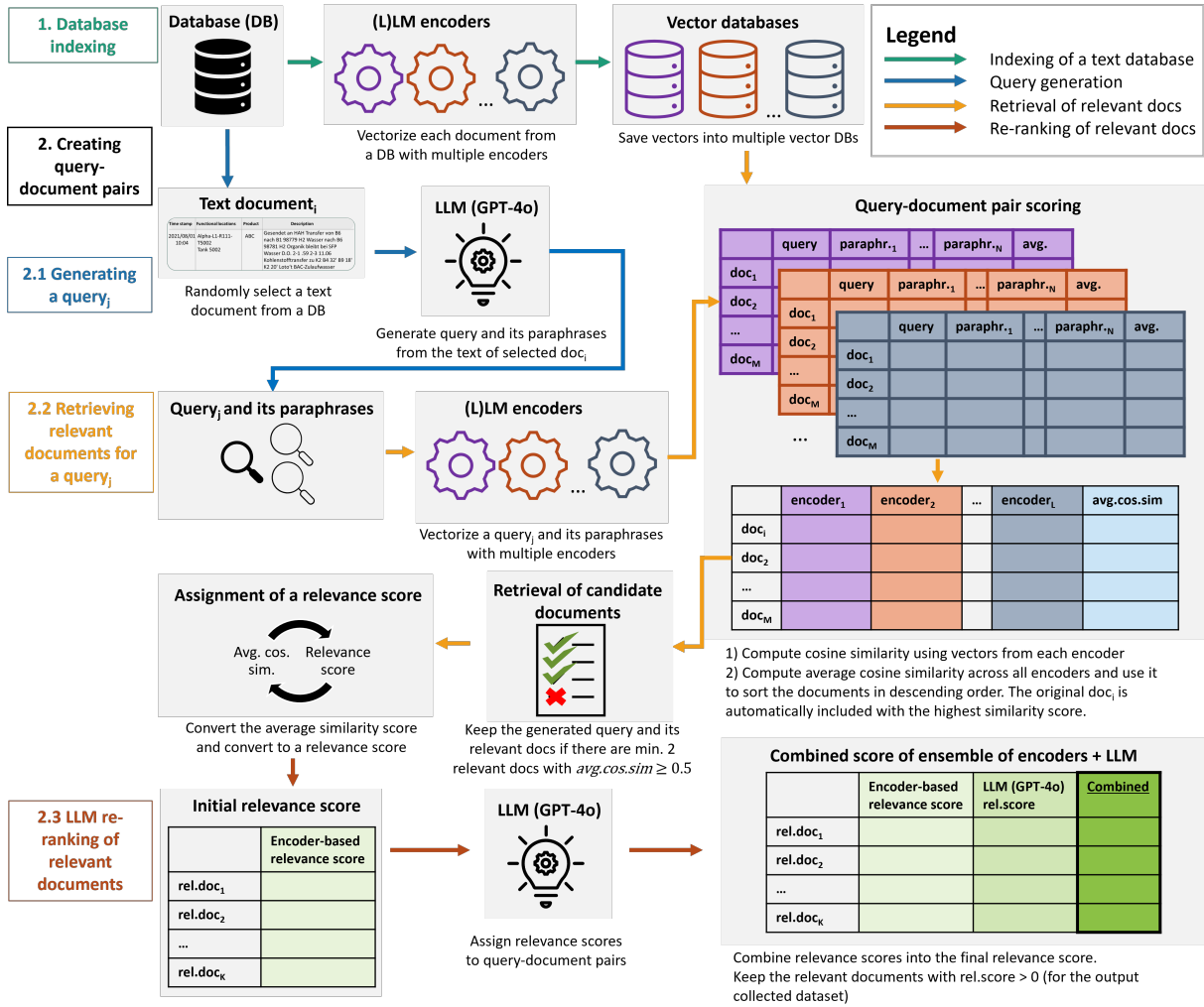


Figure 2: A proposed methodology with ensembles of (L)LM encoders used to retrieve the most relevant documents, i.e., text logs of a shift book, and with an LLM to adjust the relevance score for the document re-ranking.

include document encoding by the model architecture (i.e., bi-encoder) and mean pooling of the word vectors (additionally, see Section 4.2). Each document encoder may have learned different vector representation from the others due to its architecture, training setup, and dataset on which it was trained. We encode with multiple encoders to use this diversity of the vector representation.

For our experiments, we used three bi-encoders: two based on the sentence transformer architecture and one text encoder from OpenAI². We selected the models that supported German, had a strong performance on the semantic search on the publicly available benchmarks, yielded the best results on a small manually created dataset (see Appendix A.1 for more details), and could use cosine similarity as a score metric (see Section 3.2). Each document,

²In our implementation we used [azure-text-embedding-3-large](#) with a private endpoint.

i.e., a text log, is of a size between a sentence and paragraph and was encoded based on the input capacity of an encoder, i.e., truncated if needed.

3.2 Creating query-document pairs

Query generation A query was generated from a randomly selected document, i.e., a text log from a database, to ensure that at least one document was relevant to a query. We chose only long enough documents for the query generation, i.e., at least 100 chars. A query was generated with an LLM; in our implementation, it was GPT-4o. The prompt was designed to make generated queries extracted keywords from the text that look like search queries. Following the principle of rewriting a search query in real life to retrieve more fitting documents, the same prompt additionally generated paraphrases to the query:

Extract {query_num} search queries from the following text '{text}'. The queries need to be

meaningful as if you are supposed to use them to google. A query should contain between 2 to 5 words. Minimize using tokens with digits. Avoid using persons' names. Paraphrase each extracted query into 2 to 4 modifications. When creating paraphrases, make them look like you want to reformulate them for better search results. The paraphrases should contain synonyms of the original words in a query or syntactically correct change of the word order. Reply with a list of strings, with each string a query followed by its modifications separated by a semicolon. Keep only the text of queries, no enumeration. Consider the entire context, as it is crucial for understanding the text. The texts are from the context of chemical and pharmaceutical production environments.

If a document was long enough (i.e., more than 300 chars), multiple queries were generated and used in the annotation pipeline. We tracked a list of the documents already used in the query generation and kept selecting only the unused ones.

Retrieval We used the linear search on the L2-normalized vectors with cosine similarity as a similarity score function. We did not use other techniques to ensure each document would acquire a similarity score. We followed a two-step approach to make the final similarity score used for the retrieval more robust.

First, we computed a similarity score independently between a query and all documents and the query's paraphrases and all documents. Using paraphrases enables retrieval of a more complete list of documents than solely using the original query by covering a wider lexical diversity used in the text. The mean score per encoder is used as an intermediate similarity score for a document d and query q vectorized by encoder e_l :

$$\text{cos.sim}_{d,e} = \frac{1}{|QP|} \sum_{q \in QP} \text{cos.sim}_{q,(d,e)} \quad (1)$$

where QP is a set of a query and its paraphrases, and $|QP|$ is the size of this set.

Second, we average the scores across all encoders, thus scoring query-document similarity equally by all used vector models:

$$\text{cos.sim}_d = \frac{1}{|E|} \sum_{e \in E} \text{cos.sim}_{d,e} \quad (2)$$

where E is a set of the used encoders, and $|E|$ is the size of this set.

Despite the calculated score, the similarity score of the original document is assigned to 1.0 to ensure

that it will be among the retrieved documents and has the highest score.

Lastly, we retrieve the best-matching documents and assign relevance scores to the query-document pairs. To decide which documents to retrieve, we check two conditions: (1) the documents must have $\text{cos.sim}_d \geq 0.5$, and (2) per query, should have at least two relevant documents. The following function converted the cosine similarity to the relevance score of the ensemble of encoders on a scale of 1 to 3, where 3 meant high relevance of a document to a query, 2 was partial relevance, and 1 referred to marginal relevance:

$$\text{ensemble}_d = \begin{cases} 1 & \text{if } 0.5 \leq \text{cos.sim}_d < 0.6, \\ 2 & \text{if } 0.6 \leq \text{cos.sim}_d < 0.7, \\ 3 & \text{if } \text{cos.sim}_d \geq 0.7 \end{cases} \quad (3)$$

Re-ranking The goal of re-ranking was to use an LLM to assess the query-document pair independently from the encoders, and (1) use its relevance score combined with the encoders' score, (2) check if an LLM reevaluated the pairs as irrelevant, i.e., assigned 0 scores:

Assign a relevance score between 3 to 0 of how a query $\{query\}$ matches an event $\{text\}$ which occurred at a machinery $\{funcloc\}$. 3 is a strong relevance, i.e., a document directly contains the information requested in a query. The relevance is strong if the query matches a document on a synonym level and some spelling modifications (including a match of a full phrase/word to its abbreviation/shortening). 2 is a middle relevance, i.e., a document contains only some terms or synonyms (more than 1) or the information in a document refers to an adjacent element in a text. For example, a query specifies a specific type of container that is empty, and a document contains a different type of container that is empty. Score 1 means little relevance, i.e., a document partially contains some information requested in a query, e.g., some terms from the query but distributed across the document or only 1-2 terms/synonyms from a query are mentioned in a document, but they don't belong to one neighborhood to reflect the semantics of a query. For example, for a query 'pump is defective' some document contains general information about a pump. A score of 0 means that a document is not relevant to a query. Output only the relevance score in an integer between 0 and 3.

Combining the relevance scores from the two sources, i.e., encoders and LLM, is done with the following formula:

$$combined_d = \begin{cases} 0, & \text{if } LLM_d = 0, \\ bins(\frac{2 \cdot LLM_d + ens_d}{3}), & \text{if } LLM_d = 3, \\ bins(\frac{LLM_d + 2 \cdot ens_d}{3}), & \text{if } ens_d = 1, \\ bins(\frac{LLM_d + ens_d}{2}), & \text{else} \end{cases} \quad (4)$$

where

$$bins(x) = \begin{cases} 3 & \text{if } x \geq 2.6, \\ 2 & \text{if } 2.0 \leq x < 2.6, \\ 1 & \text{if } 1.0 \leq x < 2.0 \\ 0 & \text{if } x \leq 1.0 \end{cases} \quad (5)$$

The formula for the *combined* relevance score originates from the moderate agreement between the ensemble and LLM. Figure 3 shows that most scores were either annotated by the ensemble as 1 or by the GPT-4o as 3. Hence, when computing the combined score, we give more weight to the GPT scores when the score is 3 or to the ensemble scores when it is 1; otherwise, we compute their average. Moreover, GPT-4o tends to re-rank the fourth of the ensemble-positive scores as 0. Therefore, we keep the re-ranking score of GPT-4o. The $bins(x)$ function was empirically derived from our experiments.

		score GPT			
		0	1	2	3
score ensemble	1	12%	13%	11%	18%
	2	3%	3%	3%	13%
	3	10%	2%	1%	11%

Figure 3: The distribution of the relevance scores produced by an ensemble of encoders and GPT-4o. While the ensemble assigns 1 relevance score, GPT-4o leans towards the score of 3. The proposed combined approach balances out these model tendencies.

4 Evaluation

We evaluated our approach against the manually assigned relevance scores to the retrieved documents (Pangakis et al., 2023). The goal was to evaluate how the proposed approach agreed with how a human assessed the query-document pairs.

4.1 Experiments

We used the approach to create a test collection from seven plant shift books. We have generated

at least 80 queries for each source for which at least two relevant documents were identified. We selected 28-30 queries with up to 1000 relevant documents each for the manual annotation to make the task feasible. We provided a native German speaker familiar with the domain, and the instructions were identical to those used in the prompt. The documents were already sorted by the automated relevance scores, but the hired annotator was to assign the relevance scores between 3 and 0 without seeing these scores. Since recall-based evaluation is impossible, i.e., evaluating how many documents were retrieved from the overall number of relevant documents, we focus on evaluating final relevance scores.

Metrics We selected a set of diverse metrics to evaluate the automated assignment of the relevance score defined as various tasks: (1) inter-coder agreement between two annotators (i.e., automated and manual) measured by Krippendorff’s alpha, (2) classification metrics for the imbalanced classes, such as macro precision, recall, and F1-score, (3) a ranking metric for information retrieval and recommender systems, such as nDCG.

Krippendorff’s alpha is a robust statistical measure utilized to evaluate the reliability or inter-rater agreement across multiple annotators in categorizing or labeling data (Krippendorff, 2013). Unlike other agreement metrics, Krippendorff’s alpha is versatile, accommodating different levels of measurement, including nominal, ordinal, interval, and ratio scales. The metric yields a value between 0 and 1, where 1 signifies perfect agreement, and 0 indicates no agreement beyond chance. Due to its adaptability and rigorous assessment of inter-rater reliability, Krippendorff’s alpha is extensively employed in fields such as content analysis and qualitative data coding, where ensuring the consistency of human judgment is critical.

In the context of imbalanced datasets, **macro-averaged precision, recall, and F1-score** provide a more balanced evaluation of classification models by giving equal weight to each class, regardless of its frequency. Macro precision, recall, or F1-score first calculates these metrics for each class. Then, it averages the results, ensuring that smaller minority classes are not overshadowed by the majority class and helping to assess the model’s ability to avoid false positives across all classes. This approach is particularly useful for imbalanced datasets, where traditional accuracy measures might be skewed by

		Automated			
		0	1	2	3
Annotated	0	0	6608	5080	7491
	1	0	8446	912	253
	2	0	2312	1002	349
	3	0	1309	1021	995
		Ensemble			
		Automated			
		0	1	2	3
Annotated	0	3519	6804	2514	6342
	1	1762	2960	2370	2519
	2	85	428	849	2301
	3	0	13	31	3281
		GPT-4o (vague examples)			
		Automated			
		0	1	2	3
Annotated	0	6713	2716	1772	6173
	1	1637	3017	2349	2608
	2	77	426	833	2327
	3	0	13	32	3280
		GPT-4o (specific examples)			
		Automated			
		0	1	2	3
Annotated	0	6713	3236	2405	5020
	1	1637	4925	2710	339
	2	77	857	1880	849
	3	0	21	1317	1987
		Combined (ensemble + GPT-4o-SE)			

Figure 4: The confusion matrices of the annotated vs. automated relevance scores for four methods: an ensemble of encoders, GPT-4o with vague examples, GPT-4o with specific examples (SE), and combined ensemble + GPT-4o-SE. The combined approach allocates most of the results on the matrix diagonal, whereas its components separately lean towards one score or another.

Source	Stats	Model	Kripp.'s alpha	Precision	Recall	F1	nDCG
A	All docs: 17053 # queries: 30 # verified retrieved candidates: 2739	Ensemble	50.30	38.24	39.66	33.61	97.71
		GPT-4o-VE	31.49	52.29	41.55	38.63	95.37
		GPT-4o-SE	44.10	56.51	46.28	44.39	95.60
		Combined	67.03	60.90	53.42	54.89	97.60
B	All docs: 14065 # queries: 30 # verified retrieved candidates: 2022	Ensemble	55.57	45.49	42.02	39.97	98.01
		GPT-4o-VE	40.37	43.28	42.14	38.45	95.32
		GPT-4o-SE	45.61	45.07	44.34	40.79	95.60
		Combined	68.69	51.72	49.50	49.96	98.05
C	All docs: 129345 # queries: 30 # verified retrieved candidates: 2166	Ensemble	31.55	36.33	35.20	32.15	93.49
		GPT-4o-VE	44.41	41.17	53.40	37.47	93.62
		GPT-4o-SE	46.70	41.37	54.18	37.89	93.74
		Combined	61.35	46.92	51.33	48.16	95.35
D	All docs: 70823 # queries: 30 # verified retrieved candidates: 5111	Ensemble	14.39	18.56	25.81	21.25	90.44
		GPT-4o-VE	31.40	45.65	49.42	39.85	93.11
		GPT-4o-SE	38.11	45.64	50.48	41.11	93.72
		Combined	54.71	50.87	48.47	46.38	94.64
E	All docs: 9730 # queries: 28 # verified retrieved candidates: 7562	Ensemble	81.60	8.34	36.16	12.73	59.09
		GPT-4o-VE	-39.27	24.07	39.85	11.32	67.64
		GPT-4o-SE	-23.67	27.56	41.68	20.29	68.52
		Combined	-24.91	30.00	48.40	24.19	66.05
F	All docs: 25752 # queries: 28 # verified retrieved candidates: 2741	Ensemble	8.56	31.16	33.42	28.01	89.88
		GPT-4o-VE	26.33	39.60	44.14	36.66	91.16
		GPT-4o-SE	39.97	45.95	49.05	42.82	92.72
		Combined	41.74	46.48	46.79	44.79	91.63
G	All docs: 63570 # queries: 29 # verified retrieved candidates: 4406	Ensemble	-2.31	23.53	38.09	28.06	87.57
		GPT-4o-VE	-3.68	26.62	42.49	21.71	86.22
		GPT-4o-SE	1.12	32.69	45.55	25.97	86.78
		Combined	14.90	34.19	39.44	30.35	86.65
Average	All docs: 330338 # queries: 205 # verified retrieved candidates: 26747	Ensemble	10.92	28.81	35.77	27.97	88.03
		GPT-4o-VE	18.72	38.96	44.71	32.01	88.92
		GPT-4o-SE	27.42	42.11	47.37	36.18	89.52
		Combined	40.50	45.87	48.19	42.68	90.00

Table 1: The proposed approach of combining relevance scores produced by an ensemble of text encoders and reranking by GPT-4o yields, on average, the best results in three types of metrics, i.e., intercoder agreement, accuracy, and ranking.

the model’s performance on the dominant class. At the same time, macro-averaging ensures a fair evaluation of all classes.

Balanced accuracy is a metric designed to evaluate classification performance on imbalanced datasets, where traditional accuracy may be misleading due to the disproportionate representation

of classes. It is calculated as the average of the true positive rate (recall) for each class, ensuring that all classes, including the minority class, are equally considered. Unlike standard accuracy, which can be inflated by the model’s performance on the dominant class, balanced accuracy provides a more equitable assessment by giving equal weight to both

the positive and negative classes, regardless of their prevalence in the dataset. This makes it a more robust metric for evaluating models in scenarios where class imbalance is a concern, as it reflects the model’s ability to classify both frequent and infrequent classes correctly.

Normalized Discounted Cumulative Gain (**nDCG**) is a widely used evaluation metric for ranking tasks, particularly in information retrieval and recommender systems (Liu and Zsu, 2009). It measures the ranking quality by comparing the predicted order of items to the ideal, or ground truth, ranking. nDCG is based on the Discounted Cumulative Gain (DCG) concept, which assigns higher relevance scores to items ranked at the top of the list by applying a logarithmic discount factor to lower-ranked items. This emphasizes the importance of correctly ranking more relevant items higher. nDCG normalizes this score by dividing the DCG by the ideal DCG (IDCG)—the DCG of the perfect ranking—resulting in a value between 0 and 1. A score of 1 indicates a perfect ranking, while lower scores reflect the degradation in ranking quality. This metric is particularly useful in scenarios where the relevance of items decreases with their position in the ranked list, making it a robust measure for evaluating the effectiveness of ranked outputs.

Baselines To measure the impact of each of these components within the proposed approach, we compare the proposed approach to ranking solely with the ensemble of encoders (Ens.) or GPT-4o (GPT). Moreover, we compare GPT-4o scores produced by two versions of prompts: with vaguely formulated examples of query-document relevance (*GPT-4o-VE*) and specific examples (*GPT-4o-SE*) of the pairs and corresponding scores³. The proposed approach is denoted as *Comb.* and consists of a combined ensemble of encoders and GPT-4o re-ranking prompted with specific examples.

Results Table 1 reports metrics computed per method across 7 created test collections and their average. The table shows that the proposed method of combining an ensemble of encoders and GPT-4o outperformed these methods applied independently. The approach outperformed the baselines in all metrics, but Krippendorff’s alpha measures the most significant impact. Combining the relevance scores

³We report here only a prompt with vague examples of what we used in our experiments. We cannot provide prompts with specific examples because they contain proprietary data.

Rel.score	Ens.	GPT-4o-VE	GPT-4o-SE	Comb.
0	–	18.3	38.6	38.6
1	87.9	30.8	31.4	51.2
2	27.4	23.2	22.7	51.3
3	29.9	98.7	98.6	59.8
average	36.3	42.8	47.9	50.2

Table 2: Recall the score classification compared to the manually assigned relevance scores. Providing specific examples on prompting (GPT-4o-SE) outperformed prompting with vague examples (GPT-4o-VE), with the most noticeable improvement in recognizing irrelevant query-document pairs, which scored as 0. Combining an ensemble of encoders (Ens.) with GPT-4o-SE yielded worse recall for relevance scores 1 and 3 but significantly improved the recall on the more ambiguous score 2.

produced by an ensemble of encoders with GPT-4o, on average, improved the inter-coder agreement by a factor of 4. The results also show that providing explicit examples of query-document pairs with their corresponding scores systematically improves all metrics compared to a prompt with vague examples.

Further, we built confusion matrices to see how the score assignment was distributed between manually annotated and automated relevance scores. Figure 4 shows that the annotator often assessed the query-document pairs as irrelevant despite the score. Moreover, we see that the ensemble of encoders assigned a lot of pairs to score 1, whereas GPT-4o tends to assess the pairs more positively, with a score of 3 in many cases. Providing examples of query-document pairs with positive relevance scores has improved the correct assignment of the 0 score. Table 2 shows recall computed based on these matrices. The ensemble of encoders has the highest recall score of 1, whereas all versions of GPT-4o have the highest recall score of 3. Combining both yields the highest result on score 2 (which seems to be the hardest category to decide) and the highest average recall.

4.2 Discussion and future work

The evaluation results show that combining multiple relevance scores from diverse scoring methods increases the approach’s agreement and performance. We tested the approach on the low-resource language of the domain-specific German used on the production sights. Although the approach reaches moderate agreement with the human labels, it can produce a large-scale, diverse evaluation collection with minimum human anno-

tation effort. If the final relevance scores are not ideal and still require manual verification of the query-document pairs, the time required for it is considerably lower than performing the full annotation pipeline from scratch. Below, we discuss the findings, possible adjustments to the other languages, and further improvements.

Zero- vs few-shot learning for the domain-specific tasks Our experiments have shown that providing specific examples of the query-document pairs and describing how to assign each score enables LLMs to provide more accurate scores. These examples in the few-shot learning setup help shift an LLM towards a domain of interest, which is crucial in prompting an LLM mainly trained on the data with common knowledge towards a specific knowledge area.

Other languages Nowadays, there is a vast majority of publicly available and commercial document encoders⁴. For example, some sentence transformer models support 50 languages⁵. A model store of HuggingFace comes in handy for selecting suitable document encoders for an ensemble of encoders. One of the most recent public multilingual encoders is E5 Text Embeddings⁶ (Wang et al., 2024) trained for 94 languages. Another hub of a vast selection of encoders is available via LangChain integration⁷. Moreover, for sentences or short paragraphs, mean pooling of the word vectors can serve as an extra document encoding method. For example, fastText supports 157 languages⁸ and has already been applied as a document encoder for a domain-specific language (Zhukova et al., 2021, 2024).

The recent releases of multiple public multilingual LLMs make the methodology more feasible to expand to more languages. For example, LLaMa 3⁹, EuroLLM-9B¹⁰, Salamandra-7B¹¹, and OpenGPT-X Teuken-7B¹² can be used instead of GPT-4o for

⁴Some examples of commercial encoders are [OpenAI embeddings](#) and [Cohere](#)

⁵<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>

⁶<https://huggingface.co/intfloat/multilingual-e5-base>

⁷LangChain supports official integration of embeddings or APIs and offers community API for more models

⁸<https://fasttext.cc/docs/en/crawl-vectors.html>

⁹<https://ai.meta.com/blog/meta-llama-3/>

¹⁰<https://huggingface.co/utter-project/EuroLLM-9B>

¹¹<https://huggingface.co/BSC-LT/salamandra-7b>

¹²<https://huggingface.co/openGPT-X/>

query generation and re-ranking query-document pairs as a free alternative. Still, the performance comparison of these models compared to GPT-4o remains for further investigation.

Further improvements Despite the approach performing better than the baselines, the final metrics can be interpreted as weak agreement or moderate effectiveness. The proposed approach of combining the similarity scores and, later, the relevance scores is rather naive and can be improved. First, the encoders may have a more sophisticated way of score combination, e.g., from reliability weight per encoder score to the loss function that will minimize disagreement between the encoders. Second, multi-agent LLMs can be used to solve a complicated task of the query-document relevance assessment (Suzgun and Kalai, 2024; Becker, 2024; Yang et al., 2024), or alternatively, various LLMs can be asked to perform the same task (Yin et al., 2023; Tan et al., 2024).

5 Conclusion

This paper investigates a principle of ensemble learning with "weak" text encoders to create a test collection for the semantic search evaluation. We combined multiple text encoder models for document retrieval. We experimented with creating a test collection for semantic search evaluation in the domain of the German process industry. The experiments showed that computing the final relevance score by combining the average score of the ensemble of text encoders and an independent relevance score created by an LLM for each query-document pair increases the inter-coder agreement and accuracy metrics for several datasets. We invite the research community to apply further and investigate the proposed methodology across additional languages and domains.

6 Limitations

The methodology for automated data collection for semantic search in low-resource languages faces several limitations.

Limited Access to Commercial LLMs The lack of accessibility to commercial APIs of LLMs can lead to different results when relying on publicly available LLMs than those reported. These public models may not have the same performance or language support as commercial offerings, making

Teuken-7B-instruct-research-v0.4

it difficult to ensure reliable and high-quality data collection across different low-resource languages.

Ethical and Legal Constraints using LLMs Depending on a domain, using public APIs or publicly hosted LLMs, e.g., on a university cluster, may not be possible. For instance, the legal constraints around data privacy in the healthcare domain (e.g., GDPR compliance) may be stricter than in other industries, necessitating different data handling practices. This could limit the generalizability of the methodology when crossing into different regulatory environments.

Different prompting requirements Low-resource languages may require tailored prompting strategies to extract meaningful and accurate data from LLMs. A prompting approach that works for one language or model might not generalize well to others, necessitating the design of custom prompts for each language or LLM, adding complexity to the automated data collection process.

Lack of multiple strong text encoders Not all low-resource languages have sufficient encoder-based language models for effective use in automated data collection. Some languages may have only one or even no pre-trained encoders, limiting the ability to implement encoder-decoder architectures commonly used in semantic search, which could reduce performance and accuracy for these languages.

Complex adjustments for other downstream tasks Automated collection of datasets for downstream tasks, such as named entity recognition, sentiment analysis, or machine translation, may require significant adjustments for low-resource languages. This could involve re-tuning models, modifying preprocessing pipelines, or adapting annotations, which can be time-consuming and resource-intensive, hindering the scalability of the methodology across different languages.

7 Ethic considerations

Data Privacy and Consent The sensitive private data used in these studies is protected under GDPR regulations, ensuring full compliance with privacy laws. As a result, explicit consent from data subjects was obtained where required. Due to GDPR restrictions, specific examples or direct details regarding the data cannot be provided. Additionally, anonymization techniques were applied

to safeguard personal information.

Transparency and Accountability The code, datasets, and implementation details that can be shared publicly have been fully discussed, with links provided throughout the main paper and appendix. These resources ensure the research is transparent and can be replicated and scrutinized. However, parts of the work that fall under commercial secrets cannot be revealed due to proprietary restrictions. This limitation impacts transparency, but necessary steps have been taken to share as much as possible without violating commercial confidentiality.

Acknowledgments

This Project is supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag.

References

- Toyin D. Aguda, Suchetha Siddagangappa, Elena Kochkina, Simerjot Kaur, Dongsheng Wang, and Charese Smiley. 2024. [Large language models as financial data annotators: A study on effectiveness and efficiency](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10124–10145, Torino, Italia. ELRA and ICCL.
- Jonas Becker. 2024. [Multi-agent large language models for conversational task-solving](#). *Preprint*, arXiv:2410.22932.
- Chenhui Chu and Rui Wang. 2018. [A survey of domain adaptation for neural machine translation](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- David Farr, Nico Manzonelli, Iain Cruickshank, Kate Starbird, and Jevin West. 2024. [Llm chain ensembles for scalable and accurate data annotation](#). *Preprint*, arXiv:2410.13006.
- Michael A. Hedderich, Lukas Lange, Heike Adel, Jan-nik Strötgen, and Dietrich Klakow. 2021. [A survey on recent approaches for natural language processing in low-resource scenarios](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.
- K. Krippendorff. 2013. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications.

- Bhawesh Kumar, Jonathan Amar, Eric Yang, Nan Li, and Yugang Jia. 2024. [Selective fine-tuning on llm-labeled data may reduce reliance on human annotation: A case study using schedule-of-event table detection](#). *Preprint*, arXiv:2405.06093.
- Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023. [CoAnnotating: Uncertainty-guided work allocation between human and large language models for data annotation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1487–1505, Singapore. Association for Computational Linguistics.
- Ling Liu and M. Tamer Zsu. 2009. *Encyclopedia of Database Systems*, 1st edition. Springer Publishing Company, Incorporated.
- Yuxuan Lu, Bingsheng Yao, Shao Zhang, Yun Wang, Peng Zhang, Tun Lu, Toby Jia-Jun Li, and Dakuo Wang. 2023. [Human still wins over llm: An empirical study of active learning on domain-specific annotation tasks](#). *Preprint*, arXiv:2311.09825.
- Ibomoiye Domor Mienye and Yanxia Sun. 2022. [A survey of ensemble learning: Concepts, algorithms, applications, and prospects](#). *IEEE Access*, 10:99129–99149.
- Nicholas Pangakis, Samuel Wolken, and Neil Fasching. 2023. [Automated annotation with generative ai requires validation](#). *Preprint*, arXiv:2306.00176.
- Yu Shang, Yu Li, Fengli Xu, and Yong Li. 2024. [Synergy-of-thoughts: Eliciting efficient reasoning in hybrid language models](#). *Preprint*, arXiv:2402.02563.
- AAAI Press Staff, Pater Patel Schneider, Sunil Issar, J Scott Penberthy, George Ferguson, Hans Guesgen, Francisco Cruz, and Marc Pujol-Gonzalez. 2023. [From humans to machines: Can chatgpt-like llms effectively replace human annotators in nlp tasks?](#) In *Workshop Proceedings of the 17th International AAAI Conference on Web and Social Media*, Limassol, Cyprus.
- Mirac Suzgun and Adam Tauman Kalai. 2024. [Meta-prompting: Enhancing language models with task-agnostic scaffolding](#). *Preprint*, arXiv:2401.12954.
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. [Large language models for data annotation: A survey](#). *Preprint*, arXiv:2402.13446.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. [Multilingual e5 text embeddings: A technical report](#). *arXiv preprint arXiv:2402.05672*.
- Joshua C. Yang, Damian Dailisan, Marcin Korecki, Carina I. Hausladen, and Dirk Helbing. 2024. [Llm voting: Human choices and ai collective decision-making](#). *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 7(1):1696–1708.
- Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. 2023. [Exchange-of-thought: Enhancing large language model capabilities through cross-model communication](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15135–15153, Singapore. Association for Computational Linguistics.
- Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. [Can chatgpt reproduce human-generated labels? a study of social computing tasks](#). *Preprint*, arXiv:2304.10145.
- Anastasia Zhukova, Felix Hamborg, and Bela Gipp. 2021. ANEA: Automated (named) entity annotation for German domain-specific texts. In *Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021) co-located with JCDL 2021*, volume 3004, Virtual Event. CEUR-WS.org.
- Anastasia Zhukova, Lukas von Sperl, Christian E. Matt, and Bela Gipp. 2024. [Generative user-experience research for developing domain-specific natural language processing applications](#). *Knowledge and Information Systems*, 66:7859–7889.

A Appendix

A.1 Selection of the bi-encoder models

The following section describes the methodology of the semi-automated collection of the test dataset for semantic search. The produced dataset is an intermediate version that helped navigate the decision on the model selection for the ensemble of encoders.

We selected five publicly available text encoders and one commercial, all supporting the German language. All models use cosine similarity as a similarity metric.

A.1.1 Dataset

Table 4 reports the properties of a small manually created test dataset used to select encoders for the ensemble.

A.1.2 Evaluation and metrics

The models listed in the Table 3 were evaluated with multiple information retrieval metrics described below.

Liu and Zsu (2009) defines the metrics from our evaluation as follows.

Models	P@10	R@10	F1@10	MAP@10	MRR	nDCG@10	AVG
T-Systems-onsite/ german-roberta-sentence-transformer-v2	16.84	9.08	9.85	38.02	45.49	20.25	23.26
thuan9889/ llama_embedding_model_v1	27.37	12.91	14.74	37.49	45.83	26.86	27.53
PM-AI/ bi-encoder_msmarco_bert-base_german	34.21	20.30	20.79	46.67	53.92	31.42	34.55
sentence-transformers/ msmarco-distilbert-multilingual-en-de-v2- tmp-lng-aligned	28.95	24.43	17.96	49.12	54.91	32.73	34.68
sentence-transformers/ multi-qa-mpnet-base-cos-v1	30.00	20.80	18.99	51.17	58.77	31.99	35.29
azure-text-embedding-3-large	38.42	22.25	23.39	66.68	69.30	39.13	43.20

Table 3: An evaluation of the text encoder models to be used with the ensemble of encoders. We used a small, manually-created test collection to assess the capabilities of the available encoders. We selected the top 3 best encoders based on the average across six information retrieval metrics.

Parameter	Value
# documents	79.6K
# queries	20
# relevant documents	406

Table 4: Small manually created test dataset used to select encoders for the ensemble.

Precision@N In an information retrieval system that retrieves a ranked list, the top-n documents are the first n in the ranking. Precision at n is the proportion of the relevant top-n documents.

Recall@10 Recall at n is the proportion of the relevant top-n documents given the overall number of relevant documents.

F1@10 is a harmonic mean of precision and recall, providing a single metric that balances the two.

MAP@10 The Mean Average Precision (MAP) is the arithmetic mean of the average precision values for an information retrieval system over a set of n query topics. It can be expressed as follows:

$$MAP@10 = \frac{1}{n} \sum_n AP@10_n \quad (6)$$

where $AP@N$ represents the Average Precision value for a given topic from the evaluation set of n topics. Average precision is a measure that combines recall and precision for ranked retrieval results. For one information need, the average precision is the mean of the precision scores after each relevant document is retrieved.

$$AP@10 = \frac{\sum_r P@10}{R} \quad (7)$$

where r is the rank of each relevant document, R is the total number of relevant documents, and $P@10$ is the precision of the top-10 retrieved documents.

MRR The Reciprocal Rank (RR) information retrieval measure calculates the reciprocal of the rank at which the first relevant document was retrieved. RR is 1 if a relevant document was retrieved at rank 1; if not, it is 0.5 if retrieved at rank 2, and so on. The measure is called the Mean Reciprocal Rank (MRR) when averaged across queries.

nDCG Discounted Cumulated Gain (DCG) is an evaluation metric for information retrieval (IR). It is based on non-binary relevance assessments of documents ranked in a retrieval result. It assumes that, for a searcher, highly relevant documents are more valuable than marginally relevant documents. It further assumes that the greater the ranked position of a relevant document (of any relevance grade), the less valuable it is for the searcher because the less likely it is that the searcher will ever examine the document – and at least has to pay more effort to find it. $nDCG$ is a normalized metric calculated on the maximum possible DCG through position p, e.g., 10.

A.2 Results

Table 3 reports the evaluation of the selected text encoders. We selected the top 3 best encoders based on the average across six information retrieval metrics, i.e., two public and one commercial model. The commercial model in a multilingual encoder LLM shows a steep metric improvement compared to the public LMs. We assume that having an initial strong encoder in the ensemble can impact the overall result later.

Citation for this Paper

```
@inproceedings{zhukova2024,  
  author={Anastasia Zhukova, Christian E. Matt, Bela Gipp},  
  title={Automated Collection of Evaluation Dataset for Semantic Search in Low-Resource  
Domain Language},  
  booktitle={Proceedings of the First Workshop on Language Models for Low-Resource Languages  
(LoResLM) co-located with the 31st International Conference on Computational Linguistics  
(COLING 2025)},  
  pages={11},  
  publisher={Association for Computational Linguistics},  
  address={Abu Dhabi, the United Arab Emirates},  
  year={2025},  
  month={01}  
}
```