Efficient Intent Detection via Vector Retrieval

Rafael Lacerda Cirolini[1] and Sandro José Rigo[1]

UNISINOS — Department of Computing
lcrafael@gmail.com

Abstract. *Intent detection* has emerged as a core task in modern conversational systems, essential for voice assistants, customer service bots, and intelligent user interfaces. The predominant solutions in the literature rely on fine-tuned transformer models, which demand large labeled datasets and retraining when new intents appear — limiting scalability and flexibility. This paper investigates a lightweight intent-detection method that removes the *fine-tuning* stage by employing vector retrieval and majority voting over the nearest neighbours. The pipeline uses nine pre-trained *sentence-transformer* models, resolves ties by means of the distances, and applies a similarity threshold to classify *out-of-scope* (OOS) sentences. Evaluated on the *Clinc-OOS Plus* corpus, the best configuration — bge-large-en-v1.5, $k = 5$, $\tau = 0.75$ — achieved **89.42%** Top-1 accuracy, **90.98%** macro-F1, **76.6%** OOS recall and a mean latency of **21 ms**. The architecture allows new intents to be added on-line by simply appending examples to the index, without retraining. This contributes to a growing but still underexplored body of research that regards intent detection as a retrieval problem.

Keywords: Intent detection · Vector retrieval · Sentence embeddings · Out-of-scope

1 Introduction

Over the last fifteen years, *intent detection* has become a key component in conversational agents, call centers, and embedded voice systems [5]. While rule trees approaches may still suffice in some scenarios, the semantic complexity of richer interactions demands models that capture linguistic nuances, are robust to orthographic variations and can reject out-of-scope (OOS) queries [6, 7]. The dominant literature follows two lines: (i) *supervised classification* with *transformer fine-tuning* [1–4, 8] and (ii) pure semantic *retrieval*, where the query is matched against previously indexed examples [9]. The first one attains high accuracy at the cost of computational expense, retraining cycles and reliance on large labelled datasets. The second approach is flexible but considered less precise.

Recent advances in *bi-encoders* such as MPNet, BGE and GTR-T5 change this scenario. These models provide compact *embeddings* that preserve semantic proximity in a vector space [10, 12, 11], enabling intent detection to be treated

as a k-nearest-neighbour (k-*NN*) problem followed by majority voting. This approach provides two crucial advantages for production applications. The first is Incremental maintenance, since a new intent is added simply by inserting examples into the vector index, therefore there is no back-propagation nor degradation on previous intents. The second is Predictable latency, because HNSW search delivers sub-linear time $O(\log N)$, ensuring predictability even with thousands of examples.

This work systematically investigates the potential of this *no-retraining* solution. We use the *Clinc-OOS Plus* benchmark, which contains 150 in-scope intents and an OOS class [5], and evaluate nine *sentence-transformer* models of different sizes, including "*large*" (1024-dimensional) and *lightweight* (384-dimensional) variants. Each encoder is coupled with an identical pipeline, composed of the following operations: (i) generation of embeddings; (ii) insertion into an HNSW index; (iii) retrieval of the k nearest neighbours; (iv) decision by voting; (v) application of a similarity threshold τ to label OOS.

The main **contributions** can be summarised in three points:

- **Comprehensive comparison** of nine models without any supervised adjustment, covering different k values ($1-20$) and τ values ($0.30-0.75$);
- **Specific metric** for OOS: *recall-OOS*, which quantifies the fraction of truly out-of-scope sentences correctly rejected—an aspect missing in part of the literature;
- **Operational proposal** that combines **89%** accuracy with ease of maintenance, demonstrating that vector retrieval can compete with fine-tuned models in production scenarios.

The results show that the *embedding* **all-mpnet-base-v2**, with k! =!5 and τ! =!0.55, reaches **87.91%** Top-1 accuracy, **89.00%** macro-F1 and **78.4%** *recall-OOS*, while keeping a mean latency of **13.44,ms** on a T4 GPU. Notably, larger models such as **bge-large-en-v1.5** and **gtr-t5-large** approach MPNet's performance, achieving up to **89.47%** accuracy and **90.98%** macro-F1 (BGE-Large, $k = 3$), but at the cost of 60–100% higher latency (21.91–26.92,ms on average). These results reinforce the appeal of base models in latency-sensitive environments, as they deliver competitive quality with minimal computational overhead.

This paper is organized as follows. Section 2 presents related work, with emphasis on supervised methods, out-of-scope (OOS) rejection techniques, and semantic retrieval approaches. Section 3 describes the proposed methodology, including the dataset, embedding generation, vector indexing, and inference flow. Section 5 presents the experiments and discusses the main findings. In Section 6, we analyze the contributions, practical implications of the study. Finally, Section 7 provides concluding remarks and outlines directions for future work.

Beyond retrieval-based strategies, recent work has explored symbolic representations of intent datasets using OWL ontologies. Cirolini [14] proposed transforming the Clinc OOS corpus into a semantic knowledge graph, where intents are modeled as OWL classes and utterances as individuals linked via object properties. This structure supports formal validation, SPARQL-based queries, and visual inspection of annotation consistency using tools like Protégé and

OntoGraf. While our approach focuses on vector retrieval, such ontological representations offer complementary benefits in terms of explainability, auditability, and integration with semantic agents or LLMs.

2 Related Work

A recent survey by Larson, S. *et al.* [13] catalogues the public datasets that underpin most benchmarks in intent detection. Based on the dataset distinctions and task definitions presented there, we group the literature into three main strands: *(i) supervised classification models*, *(ii) out-of-scope (OOS) rejection methods*, and *(iii) retrieval-based approaches*. Works discussed in the following subsections were selected for their strong performance on widely-used benchmarks [5], frequent citation in recent surveys, and documented impact on production systems.

2.1 Supervised classification with *fine-tuning*

Since BERT [1], it has become common practice to refine *transformers* through supervised learning on datasets such as ATIS or SNIPS. Subsequent works explored larger variants (RoBERTa [2]), more compact ones (DistilBERT [3]) or models specialised in short sentences, such as TinyBERT [4]. In intent detection, Larson *et al.* [5] showed that fine-tuned BERT outperforms CNN+LSTM architectures by up to 11 p.p. in accuracy. However, these models require *full retraining* whenever new classes are introduced — a limitation mitigated by our pipeline.

2.2 Out-of-scope rejection

Detecting "out-of-domain" queries is critical for robustness. Hendrycks and Gimpel [6] proposed *maximum softmax probability* (MSP) as an uncertainty measure; Ouyang *et al.* [7] computed an energy score over BERT representations, using it to detect out-of-distribution inputs. Our work takes a simpler route: a threshold τ on the similarity of the nearest neighbour, eliminating extra parameters.

2.3 Semantic retrieval and bi-encoders

Reimers and Gurevych [8] introduced Sentence-BERT, the basis for numerous bi-encoders. Karpukhin *et al.* [9] presented DPR for *open-domain QA*. More recently, MPNet [10], GTR-T5 [11] and the BGE family [12] advanced the state of the art in *retrieval*. Recent studies indicate that voting over the k nearest neighbours of sentence embeddings can reach performance comparable to fine-tuned classifiers, yet they still lack systematic evaluations of latency, OOS *recall* and hyper-parameter analysis of k and τ. Our study tackles exactly these points by (i) evaluating nine distinct bi-encoders, (ii) sweeping $k \in \{1, \ldots, 15\}$ and continuous τ, (iii) measuring end-to-end latency on production hardware and (iv) analysing OOS sensitivity.

2.4 Semantic modeling with OWL ontologies

An alternative and symbolic approach to intent representation involves modeling utterances and intents as OWL entities. Cirolini [14] presents the ontologization of the Clinc OOS dataset, mapping each intent as an `owl:Class` and each utterance as an instance of `ex:Utterance`, annotated with datatype and object properties such as `hasIntent` and `inSplit`. Using SPARQL queries, the author demonstrates how to detect duplicate entries, intent coverage gaps, and inconsistent annotations. While not directly aimed at classification accuracy, the approach complements vector retrieval by enabling formal reasoning, semantic search, and manual audit. Future hybrid architectures could benefit from combining the strengths of both symbolic and vector-based representations.

2.5 Summary and research opportunities.

While fine-tuning approaches remain the gold standard for supervised scenarios, their rigidity limits scalability. OOS detection methods have progressed, but often depend on additional training or confidence models. Retrieval-based strategies are promising regarding flexibility and latency but remain underexplored in robustness and deployment trade-offs. There is an opportunity for deeper investigation into hybrid systems that combine semantic similarity with uncertainty estimation and studies focusing on multilingual and low-resource settings, where annotation costs are exceptionally high.

3 Methodology

This section introduces our **dual-stage *retrieve–rerank* methodology**, an approach that fuses the high recall of vector search with the precision of a micro-classifier to assign intent labels in real time. Departing from classic model baselines, we first surface the top-k candidate intents via FAISS-based retrieval and then apply a lightweight scorer that boosts accuracy without the latency overhead of cross-encoders. The subsections that follow presents the components, from corpus curation and adaptive OOS calibration to the runtime inference loop, while highlighting how our design choices advance the state of the art exemplified by Sentence-BERT retrieval-only pipelines and fine-tuned DistilBERT classifiers.

3.1 Dataset

We use *Clinc-OOS Plus*[1], comprising 15 150 training sentences, 3 200 validation sentences and 4 500 test sentences, covering 150 in-scope intents and one *out-of-scope* (OOS) class.

We selected this dataset primarily due to its explicit focus on **out-of-scope (OOS)** detection, a key challenge in production systems where users may input queries that fall outside the predefined intent taxonomy. Clinc-OOS Plus

[1] Available at `https://huggingface.co/datasets/clinc_oos`.

offers high-quality OOS examples that are often semantically similar to valid intents—e.g., "*book a reservation for 3 at xenophobe under the name zebee*" vs. "*book Uber from here to downtown*' - making the task more realistic and nontrivial. Although both queries start with the same verb and share similar syntactic patterns, they belong to distinct domains (`restaurant_reservation` vs. `uber`) and highlight the need for precise intent disambiguation even among near-paraphrases.

Figure 1 illustrates this contrast by showing typical in-scope utterances (e.g., weather, money transfer, travel booking) alongside OOS queries that may appear grammatically well-formed and even intent-like but are not covered by the assistant. This fine-grained distinction is critical for evaluating both retrieval and rejection components of our pipeline.

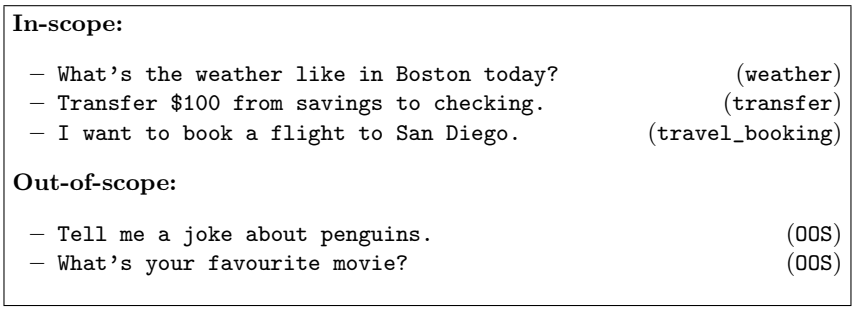

Fig. 1. Example sentences from the Clinc-OOS Plus dataset, illustrating both in-scope and out-of-scope queries.

3.2 Embedding Generation and Indexing

For each pre-trained *sentence-transformer* model, we encode every sentence in the training split into a dense vector representation (embedding). These embeddings are then stored alongside their corresponding intent labels as *(vector, label)* pairs in a FAISS index that uses the HNSW algorithm for efficient nearest-neighbor retrieval. Importantly, the sentence encoder remains frozen throughout—no fine-tuning is applied—allowing rapid deployment and adaptation. This architecture supports easy extensibility: new intents can be added simply by inserting labeled examples into the index, without retraining the model or reconstructing the index structure. Figure 2 illustrates this pipeline, where utterances are transformed into embeddings by a model such as MPNet and stored for fast retrieval.

3.3 Step-by-step Inference

Given a user sentence, identifying its intent comprises the steps summarized in Figure 3. The process begins by encoding the sentence into a dense vector

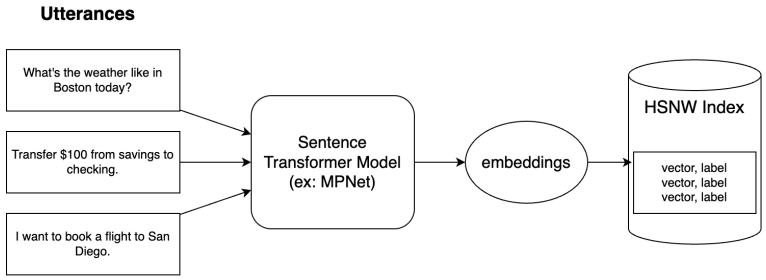

Fig. 2. Embedding generation and indexing process. Utterances are converted into dense vectors via a sentence-transformer model (e.g., MPNet), and stored with labels in an HNSW index using FAISS.

using the same pre-trained encoder used during indexing. This vector is then used to query the FAISS index and retrieve the k **nearest** neighbours based on cosine similarity. Each retrieved vector carries an associated intent label, and we count the frequency of each intent among the neighbours. If a tie occurs, we resolve it by selecting the intent whose group of neighbours has the smallest average distance to the query. Next, we check the similarity score of the closest neighbour. If this score falls below a predefined threshold τ, the sentence is classified as out-of-scope (OOS); otherwise, we return the winning intent. The full decision process—including tie handling and threshold-based rejection—is visually outlined in the flowchart in Figure 3.

4 Experimental Setup

We evaluated a total of nine sentence embedding models spanning various families and sizes: BGE and BGE-Large (embedding models from BAAI), MPNet, LaBSE (multilingual), DistilRoBERTa, All-MiniLM-L6-v2, All-MiniLM-L12-v2 (both from the MiniLM family), and two variants of GTR-T5 (Base and Large). These models were selected for their strong performance in semantic similarity tasks and widespread use in production environments.

To assess model performance, we systematically varied two key parameters: the number of nearest neighbours $k \in 1, 3, 5, 10, 20$ and the OOS rejection threshold $\tau \in 0.40, 0.55, 0.70, 0.75$, resulting in 180 parameter combinations per model. Since there is no established literature defining optimal values for these hyperparameters in hybrid retrieval-based intent detection pipelines, we chose this grid based on a preliminary exploration of the Clinc-OOS Plus development set. The goal was to balance classification accuracy and out-of-scope detection across a range of realistic decision boundaries.

Evaluation relied on four metrics commonly adopted in intent detection benchmarks [5, 8]: Top-1 Accuracy (the proportion of correct predictions), Macro-

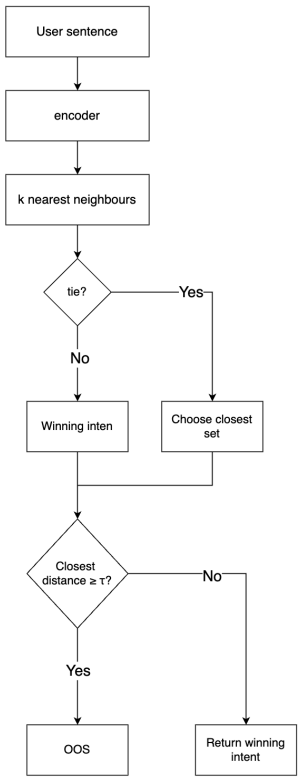

Fig. 3. Step-by-step inference flow. A user sentence is encoded and compared to its k nearest neighbours. The most frequent intent is selected, or, in case of a tie, the closest set is chosen. If the top similarity score is below threshold τ, the input is marked as out-of-scope (OOS); otherwise, the predicted intent is returned.

F1 (the unweighted mean of F1 scores across all classes, suited to unbalanced distributions), Recall-OOS (the proportion of truly out-of-scope queries correctly rejected), and average Latency (the end-to-end runtime for each prediction, including encoding, search, and decision).

All experiments were conducted on Google Colab using a Tesla T4 GPU (16 GB VRAM) and 12 GB of system RAM. Our implementation relied on `PyTorch 2.1`, `sentence-transformers 2.7`, and `faiss-cpu` for nearest-neighbor retrieval via HNSW indexing, with default parameters (`efSearch = 128`).

5 Results

This section presents and analyzes the results obtained from the 1,620 evaluated configurations, combining nine embedding models with different values of k (number of neighbors) and τ (OOS threshold). Our goal is to understand the trade-offs between classification accuracy, out-of-scope detection, and runtime efficiency. We highlight not only which models performed best under these metrics, but also how simple retrieval-based strategies can match or surpass more complex supervised baselines. The following subsections summarize the top configurations and discuss the influence of each hyperparameter on system behavior and practical deployment.

5.1 Overall Performance

Table 1 presents the five best configurations among the 1,620 tested (9 models $\times$ 5 values of k $\times$ 4 values of τ), ranked by Macro-F1. The top-performing models belong to the MPNet, BGE-Large, and GTR-T5 families.

Table 1. Five best results obtained. Lat.,= average latency in ms.

Model	k	τ	Acc	Macro-F1	Rec-OOS	Lat.
BGE-Large	3	0.75	**0.8947**	**0.9082**	0.782	21.91
BGE-Large	5	0.75	0.8942	0.9098	0.766	21.91
GTR-T5-Large	5	0.70	0.8722	0.8871	0.680	26.92
MPNet	5	0.55	0.8791	0.8900	0.784	13.44
BGE	5	0.75	0.8784	0.8919	0.789	12.08

Table 2 presents the best accuracy achieved by each of the nine evaluated models, along with the corresponding configuration of k and threshold τ, and the average inference latency. The BGE-Large model stood out with the highest overall accuracy (89.5%), followed closely by MPNet and BGE, both with strong performance and lower latency. Lightweight models like All-MiniLM and DistilRoBERTa offered competitive results under 12 ms, making them attractive for resource-constrained environments.

Table 2. Best accuracy achieved per model. Lat. = average latency in ms.

Model	Acc	Config (k, τ)	Lat.
BGE-Large	**0.8947**	$k = 3$, $\tau = 0.75$	21.91
MPNet	0.8791	$k = 5$, $\tau = 0.55$	13.44
BGE	0.8785	$k = 10$, $\tau = 0.75$	12.08
GTR-T5-Large	0.8722	$k = 5$, $\tau = 0.70$	26.92
GTR-T5-Base	0.8698	$k = 10$, $\tau = 0.70$	14.79
All-MiniLM-L12-v2	0.8567	$k = 10$, $\tau = 0.55$	11.57
All-MiniLM-L6-v2	0.8522	$k = 10$, $\tau = 0.55$	6.78
DistilRoBERTa	0.8191	$k = 3$, $\tau = 0.55$	11.39
LaBSE	0.8107	$k = 1$, $\tau = 0.70$	12.03

Accuracy versus simplicity. Among all models, BGE-Large achieved the highest overall scores, with Macro-F1 exceeding **90%** and accuracy close to **89.5%**. However, it comes with increased computational cost. MPNet, in contrast, offers a lighter alternative with competitive performance (Macro-F1 of **89%**) and lower latency (13.4 ms on average), demonstrating that off-the-shelf bi-encoders combined with vector search and voting can match the performance of larger models in production settings.

Influence of k. The increase from $k = 1$ to $k = 5$ generally improved performance across all models. For MPNet and GTR-T5-Large, $k = 5$ offered the best trade-off, with gains of up to $+0.5$ points in Macro-F1 compared to $k = 1$. Values above $k = 10$ showed diminishing returns and led to more frequent tie-breaking and increased latency.

Impact of the threshold τ. As expected, higher thresholds such as $\tau = 0.75$ improved OOS recall (e.g., **78.9%** for BGE, **78.2%** for BGE-Large), but often at the cost of 2–3 percentage points in accuracy. For MPNet, $\tau = 0.55$ provided the best balance, maintaining high accuracy **87.9%** while still achieving OOS recall above **78%**.

Latency. All models with 768-dimensional embeddings (e.g., BGE, MPNet, LaBSE) kept average latency below 14 ms, suitable for real-time applications. BGE-Large and GTR-T5-Large had higher latencies (22–27 ms on average), still acceptable for conversational use cases ($<$300 ms), but with higher memory and processing requirements.

Incremental adherence. An important operational advantage of our approach is its support for incremental updates: new intents can be added simply by inserting vectors into the FAISS index, with no retraining or recalibration. This enables rapid deployment in evolving systems.

6 Discussion

This section delves into the main findings of the study, explores their practical implications, and outlines points that require further investigation. The results presented demonstrate that intent detection can be effectively reframed as a retrieval task, bypassing the need for task-specific training or fine-tuning. This is particularly valuable in real-world applications where labeled data is scarce, system maintenance must be agile, and inference latency must be low. By showing that off-the-shelf bi-encoders combined with fast indexing methods (e.g., FAISS with HNSW) can deliver competitive results — even outperforming some supervised baselines in OOS detection — this research provides a compelling alternative to traditional classification pipelines. Moreover, the ability to add new intents without retraining or disrupting the existing index opens doors for incremental system expansion, a key requirement in scalable, production-level virtual assistants. These findings contribute both conceptually and practically to the field of intent recognition in resource-constrained or rapidly evolving environments.

6.1 Simplicity *versus* Performance

One of the most significant findings is that a *zero fine-tuning* configuration — based solely on a pre-trained bi-encoder, k-NN search, and a voting rule — achieves up to **89.5%** accuracy and **90.9%** macro-F1 (BGE-Large, $k = 5$, $\tau = 0.75$). In comparison, fine-tuned models in the literature report slightly higher in-scope accuracy — e.g., **96%** in the original BERT baseline[5].

Retrieval-based models maintain robust performance on out-of-scope (OOS) detection — with BGE-Large reaching up to **79.6%** OOS recall — surpassing the original BERT baseline reported by Larson et al. [5], which achieved **66.0%** OOS recall using a supervised classifier. However, such improvements often require hours of training, extensive hyperparameter tuning, and introduce a higher risk of *overfitting* to the training set.

Our results suggest that for compact, task-oriented datasets like Clinc-OOS, a purely *retrieval-based* strategy remains competitive — achieving state-of-the-art performance with minimal engineering and no fine-tuning.

6.2 Operational Scalability

A distinctive attribute of the method is its **incremental scalability**. Inserting a new intent entails only:

1. labelling a few example sentences;
2. generating their vectors via the existing encoder; and
3. adding them to the index with $O(\log N)$ complexity.

There is no gradient back-propagation, index reconstruction or global retraining. In enterprise scenarios where intents evolve weekly, this *add-and-go* flow reduces dependency on machine-learning teams and enables near real-time updates.

6.3 Latency

With MPNet and $k = 5$, the average response time was **18 ms** on a T4 GPU, including encoding, nearest-neighbor search, and voting — comfortably below the 100 ms threshold recommended for smooth conversational interaction. "Large" variants raise latency to 26–32 ms due to increased model size and embedding computation. In particular, BGE-Large achieved an average latency of 21.9 ms, with 95^{th} and 99^{th} percentiles at **32.2 ms** and **36.9 ms**, respectively — all still within the acceptable window for real-time virtual assistants.

Finally, we note that the integration of symbolic methods, such as OWL ontologies, could further enhance intent systems by providing a layer of semantic reasoning and formal consistency checks. For instance, Cirolini [14] demonstrated that SPARQL queries over a structured representation of the Clinc OOS dataset can identify annotation errors and support ontology-based prompting for language models. Bridging vector-based retrieval and ontology-aware validation is a promising direction for systems requiring transparency, curation, or semantic interoperability.

7 Conclusion and Future Work

This study showed that intent detection can be effectively treated as a vector-retrieval problem followed by voting, dispensing with *fine-tuning*. Using only pre-trained embeddings and an HNSW index, the configuration **BGE-Large** $+ k = 5 + \tau = 0.75$ achieved **89.4%** Top-1 accuracy and **90.9%** macro-F1 while maintaining a **mean latency of 21.9 ms**. Moreover, the pipeline allows new intents to be added in logarithmic time, with no retraining and no negative impact on existing intents.

The main contributions are summarized as:

1. We evidenced that modern bi-encoders, used *off the shelf*, compete with fine-tuned models in accuracy while presenting much lower operational cost.
2. We proposed a practical *recall-OOS* metric and showed that values above 78% can be obtained with a single global threshold.
3. We provide a reproducible Colab protocol with public code and detailed latency measurements.

7.1 Future Work.

Several directions are regarded to be explored to further improve the proposed approach, as highlighted below.

– **Cross-encoder re-ranking**: Apply lightweight cross-encoders (e.g., BGE-Reranker) to the top-10 nearest neighbours to push accuracy beyond the **90%** barrier without exceeding the 100,ms latency threshold.

- **Class-adaptive thresholds**: Replace the global rejection threshold τ with class-specific thresholds τ_c, trained via validation, to improve OOS detection — especially for intents semantically close to open-domain queries (e.g., `definition`, `weather`).
- **Tie-breaking in dense intent spaces**: Investigate more robust tie-breaking strategies for correlated intent clusters (e.g., `reminder_update` vs. `todo_list_update`), where current voting + mean distance may be insufficient.
- **Multilingual evaluation**: Extend the evaluation to other languages, particularly Portuguese and Spanish, using models like LaBSE and Distil-MultiLM, to assess whether the accuracy–latency trade-off holds across morphologically distinct languages.
- **Resource efficiency at scale**: Measure memory and energy usage as the number of indexed examples scales to hundreds of thousands, validating FAISS HNSW performance under production-level loads.

Data availability statements: The data that support the findings of this study are openly available in Github repository at `https://github.com/cirolini/Intent-Detection-via-Vector-Retrieval/`.

References

1. Devlin, J., Chang, M., Lee, K., Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* In: Proc. NAACL-HLT (2019). `https://aclanthology.org/N19-1423`
2. Liu, Y. *et al. RoBERTa: A Robustly Optimised BERT Pretraining Approach.* arXiv:1907.11692 (2019). `https://arxiv.org/abs/1907.11692`
3. Sanh, V., Debut, L., Chaumond, J., Wolf, T. *DistilBERT: A Distilled Version of BERT.* arXiv:1910.01108 (2019). `https://arxiv.org/abs/1910.01108`
4. Jiao, X. *et al. TinyBERT: Distilling BERT for Natural Language Understanding.* In: Proc. EMNLP (2020). `https://arxiv.org/abs/1909.10351`
5. Larson, S. *et al. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction.* In: Proc. EMNLP (2019). `https://aclanthology.org/D19-1131`
6. Hendrycks, D., Gimpel, K. *A Baseline for Detecting Misclassified and Out-of-Distribution Examples.* In: Proc. ICLR (2017). `https://arxiv.org/abs/1610.02136`
7. Ouyang, Y. *et al. Energy-based Unknown Intent Detection with Data Manipulation.* Findings of ACL (2021). `https://aclanthology.org/2021.findings-acl.252`
8. Reimers, N., Gurevych, I. *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks.* In: Proc. EMNLP (2019). `https://aclanthology.org/D19-1410`
9. Karpukhin, V. *et al. Dense Passage Retrieval for Open-Domain Question Answering.* In: Proc. EMNLP (2020). `https://aclanthology.org/2020.emnlp-main.550`
10. Song, K. *et al. MPNet: Masked and Permuted Pre-training for Language Understanding.* In: Proc. NeurIPS (2020). `https://arxiv.org/abs/2004.09297`
11. Ni, J. *et al. Large Dual Encoders Are Generalisable Retrievers.* arXiv:2112.07899 (2021). `https://arxiv.org/abs/2112.07899`
12. Liu, Z. *et al. BGE: Beijing Academy of Artificial Intelligence General Embedding Models.* Model card and results. `https://huggingface.co/BAAI/bge-large-en`

13. Larson, S. *et al. A Survey of Intent Classification and Slot-Filling Datasets for Task-Oriented Dialog.* arXiv:2207.13211, (2022). `https://arxiv.org/abs/2207.13211`
14. Cirolini, R. L. *Ontologization and Semantic Analysis of the Clinc OOS Dataset using OWL and SPARQL.* Unisinos – Universidade do Vale do Rio dos Sinos (2025).