ONE SIZE DOES NOT FIT ALL: EXPLORING VARIABLE THRESHOLDS FOR DISTANCE-BASED MULTI-LABEL TEXT CLASSIFICATION

Jens Van Nooten[*] University of Antwerp (CLiPS)
Andriy Kosar[*] Textgain
Guy De Pauw Textgain
Walter Daelemans University of Antwerp (CLiPS)

ABSTRACT

Distance-based unsupervised text classification is a method within text classification that leverages the semantic similarity between a label and a text to determine label relevance. This method provides numerous benefits, including fast inference and adaptability to expanding label sets, as opposed to zero-shot, few-shot, and fine-tuned neural networks that require re-training in such cases. In multi-label distance-based classification and information retrieval algorithms, thresholds are required to determine whether a text instance is "similar" to a label or query. Similarity between a text and label is determined in a dense embedding space, usually generated by state-of-the-art sentence encoders. Multi-label classification complicates matters, as a text instance can have multiple true labels, unlike in multi-class or binary classification, where each instance is assigned only one label. We expand upon previous literature on this underexplored topic by thoroughly examining and evaluating the ability of sentence encoders to perform distance-based classification. First, we perform an exploratory study to verify whether the semantic relationships between texts and labels vary across models, datasets, and label sets by conducting experiments on a diverse collection of realistic multi-label text classification (MLTC) datasets. We find that similarity distributions show statistically significant differences across models, datasets and even label sets. We propose a novel method for optimizing label-specific thresholds using a validation set. Our label-specific thresholding method achieves an average improvement of 46% over normalized 0.5 thresholding and outperforms uniform thresholding approaches from previous work by an average of 14%. Additionally, the method demonstrates strong performance even with limited labeled examples.

Keywords Multi-label classification · Semantic similarity · Text classification

1 Introduction

Multi-label text classification (MLTC) is a text classification problem where the goal is to predict one or multiple labels from a given finite label set for a single text. This type of classification is more challenging than multi-class classification, where only one label is assigned to each item, because it involves predicting multiple labels simultaneously. This introduces complexities such as an exponentially growing number of label combinations, interdependence between labels, and the fact that the number of labels for each instance can vary [1]. Furthermore, this entails that the semantics of multiple labels are represented in a single text instance.

Even though this classification problem is more complex and difficult to solve than single-label classification, it has found many real-world applications with text data, such as topic classification in the news sector [2] and commerce [3], emotion classification [4], social media monitoring [5, 6, 7], and COVID-19 symptom prediction [8]. Additionally, it is often employed in image [9, 10], video [11], and audio [12] classification. Other examples of applications include the classification of proteins [13, 14].

Distance-based classification (DBC) is a classification method that leverages the semantic similarity between texts and labels in a common embedding space to determine whether a label belongs to a text or not [15]. In traditional

[*]These authors contributed equally to this work.

distance-based classification for binary or multi-class classification, the most common approach is to assume that the closest label embedding to a text embedding is the true label [16, 17]. However, as previously mentioned, the number of labels to be predicted in multi-label classification is unknown. Therefore, an algorithm to determine the number of labels should be introduced. In distance-based classification and information retrieval, this is often achieved by setting an appropriate threshold. The main question in distance-based MLTC therefore is a question of how to optimally determine similarity thresholds to optimize performance. Although some work has explored DBC for MLTC [18, 19, 20, 21], most proposed threshold algorithms only consider a single (uniform) threshold for all labels, ignoring differences in similarity distributions between texts and individual labels. We argue that label-specific thresholds, which account for these differences, yield significantly improved performance on downstream multi-label classification tasks.

This study consists of two parts. First, using multiple MLTC datasets and sentence encoders, it examines similarity scales, i.e. the range between the lower bound and upper bound expressed by the model when measuring the similarity between labels and texts in an embedding space. Second, based on the results of this exploratory analysis, it investigates the effectiveness and feasibility of implementing label-specific thresholds for MLTC.

The contributions of this empirical study are the following:

1. We investigate an under-explored approach to multi-label text classification using a distance-based classification method, offering an efficient alternative to more commonly used methods.
2. We demonstrate that uniform thresholds are suboptimal for multi-label classification tasks, and that label-specific thresholds lead to superior performance.
3. We propose a simple, novel and intuitive method for determining label-specific thresholds with minimal annotated data. Our method significantly outperforms existing uniform thresholding techniques and achieves competitive results compared to state-of-the-art zero-shot classification with LLMs across several publicly available datasets.
4. We conduct a comprehensive evaluation of open-source state-of-the-art sentence encoders for multi-label classification, assessing their effectiveness in addressing complex classification challenges.

The findings of this study are not only applicable to MLTC but also have potential applications in information retrieval, particularly in Retrieval-Augmented Generation (RAG). Moreover, our findings could be extended to other modalities, such as video, audio, and images.

The paper is structured as follows. Section 2 defines the problem and presents the key hypotheses of this empirical study. Section 3 reviews relevant studies that form the foundation of the proposed methodology. Subsequently, Section 4 outlines the methodology: It begins with the exploratory study, followed by an explanation of the classification experiments. Then, Section 5 presents the results and analysis of the experimental study and the classification experiments.

2 Problem Statement and Hypotheses

We formulate the problem as a threshold optimization problem, following prior work [18, 19, 20, 22, 21]. We build on previous literature by finding the threshold for each label individually to maximize the overall performance. Let $L = \{l_1, l_2 ... l_n\}$ be the set of labels and let $\theta = \{0.01, 0.02, 0.03 ... 1.0\}$ be a set of pre-defined thresholds θ. The goal of the proposed algorithm is to find a threshold for each label (θ_L^*) that yields the highest performance (F_1 score):

$$\theta_L^* = arg\ max\ F_{1_l}(\theta) \tag{1}$$

We formulate the following four key hypotheses:

1. **Variability in Embedding Scales Across Models (H1)**: Different embedding models exhibit unique similarity scales, which raises challenges for their interchangeable application.
2. **Domain-Specific Differences (H2)**: Even within a single model, texts from various genres or domains can demonstrate distinct similarity scales, limiting cross-domain applications.
3. **Class-Specific Similarity Variations (H3)**: Within the semantic space of a single model and domain, different classes may exhibit unique similarity scales. This variation depends on how effectively the embedding model captures and reflects the semantic meaning of texts and labels, particularly for labels with weak representations.
4. **Class-Specific Thresholds (H4)**: Optimized class-specific thresholds enhance the performance of multi-label distance-based text classification.

3 Related Research

3.1 Multi-Label Text Classification

As mentioned previously, multi-label classification carries unique difficulties as opposed to single-label classification. Researchers have previously attempted to overcome these challenges by transforming the problem into multiple binary classification problems [23], modeling hierarchies [24] and label correlations [25], performing data augmentation [7, 26], or adapting loss functions [27]. For a more comprehensive overview of multi-label classification approaches and challenges, consult [1] and [28].

Recent advances in large language models (LLMs), such as BERT [29] and the GPT series [30], have transformed MLTC –and text classification in general– into a zero-shot or few-shot task [31, 32, 33]. However, as highlighted in [34] and [33], state-of-the-art generative LLMs still perform poorly on MLTC tasks, mainly due to the complexity of the task and the limited availability of official annotation guidelines that can be included in prompts. Nevertheless, deploying LLMs for MLTC in an industrial setting is attractive because they do not require task-specific training data and enable flexibility, especially with regard to classification on datasets with evolving label sets. Compared to fine-tuned models, they do not need to be retrained whenever a label is added or removed from a dataset. However, as highlighted in [35], many multi-label classification approaches may suffer from high computational complexity, which makes them less applicable in industrial settings. In such cases, models are required to be lightweight and fast.

3.2 Distance-based Text Classification

Alongside supervised learning, distance-based classification (DBC) has evolved as a computationally efficient method, where the similarity between text and label embeddings is used to determine whether a label belongs to a text or not. Labels and texts are usually encoded using neural word embeddings such as *GloVe* [36], *word2vec* [37] or *fastText* [38], or using contextual embeddings with sentence transformers [39]. This method was introduced by [15], who encoded texts and semantic class concepts from Wikipedia as labels in a semantic space to perform classification based on the cosine similarity between label and text. In single-label text classification, as performed in [17], usually the closest label is considered to be the true label. In this approach, both labels and texts only have to be encoded once, which makes it lightweight and fast compared to inference with LLMs, since inference only requires a similarity metric to be calculated.

However, this becomes more complex when performing distance-based MLTC, since there are multiple potential true label candidates for a single text instance. Encoding the multiple crucial facets of a multi-label text effectively remains a challenge. Several studies have addressed this complexity using distance-based methods for MLTC [18, 19, 20, 22, 21]. In these approaches, both texts and label representations are also embedded in a joint space, after which the similarity between a text embedding and label embedding is calculated. If the similarity (typically cosine similarity) equals or exceeds a threshold, a label is assigned to the text.

Although there are several ways to determine these thresholds, most research opts for using user-defined thresholds, selecting the best-performing one based on validation set performance [18, 22, 21]. However, this method ignores the differences in similarity scales between labels and texts, because the same threshold is applied to every label. Not optimizing the threshold for each label separately when a validation set is available could inhibit performance. Table 1 presents text examples that highlight the need for label-specific thresholds, as illustrated by the variation in cosine similarity values between both models and labels.

Table 1: Comparison of cosine similarity values between a text and the corresponding true labels across models

	LitCovid		**Reuters**			
Text	In this study, we aimed to assess the association between development of cardiac injury and short-term mortality as well as poor in-hospital outcomes in hospitalized patients with COVID-19.		Soybean imports are forecast to rise to 425,000 tonnes in 1987/88 (October/September) from an estimated 300,000 in 1986/87 and 375,000 in 1985/86, the U.S. Embassy said in its annual report on Indonesia's agriculture.			
True Labels	Medical Treatments	Diagnostic Methods	Oilseed	Soybeans	Meal and Feed	Soybean Meal
GIST-Large	0.31	0.27	0.44	0.65	0.32	0.52
Stella	0.28	0.30	0.41	0.58	0.36	0.58
Mxbai	0.43	0.40	0.51	0.66	0.43	0.56

Determining optimal dynamic thresholds has also been researched for RAG. For example, the study by [40] highlights that different models require distinct similarity thresholds to reach optimal performance in RAG due to variations in model architectures, interpretations of similarity scores, and training procedures. However, the authors do not explore

Table 2: Dataset Statistics

Dataset	Text Type	Task	N Train	N Val	N Test	N Lbls	N lbls/ Text	Mn Tkns	Mdn Tkns
SemEval	tweets	emotion	6,837	886	3,259	11	2.38	27	28
BioTech	news	topics	2,344	414	381	31	1.84	655	572
Reuters	news	topics	7,493	1,323	3,023	118	1.01	180	121
AAPD	abstracts	topics	53,840	1,000	1,000	52	2.41	128	99
LitCOVID	abstracts	topics	21,204	3,756	6,239	7	1.37	303	292

variations in similarity by domain or specific query type. In the present study, we expand on this optimization process by accounting for variability in the encoded semantics of each label in a multi-label context.

4 Methodology

This study is divided into two parts, for which the methodology is described in the following sections. In the first part, we explore the aforementioned hypotheses, i.e. the variability in similarity distributions among models (**H1**), datasets (**H2**), and labels (**H3**) by conducting statistical analyses on similarity distributions derived from the labeled training partitions of the datasets. The second part aims to assess the feasibility of label-specific thresholds by applying them to a broad range of MLTC datasets (**H4**).

4.1 Datasets

The experiments are conducted on five publicly available MLTC datasets, namely: the English subset of the SemEval 2018 dataset [41], the BioTech news dataset[2], a modified version ("Apté Mod") of the Reuters-21578 dataset [42][3], the arXiv Academic Papers Dataset (AAPD) [43][4] and the LitCovid dataset [44]. An overview of the datasets is provided in Table 2. These datasets were selected based on the availability of the original text data and the inclusion of label names, which are essential for distance-based text classification. For the AAPD and LitCovid datasets, we created a validation split by performing a stratified split on the training data, following the method described in [45].

4.2 Exploratory Study

For the exploratory study, we first embed the texts from the training partitions and all corresponding labels of the data using neural word embeddings such as *GloVe* [36] and sentence transformers [39]. The latter models are selected based on their ranking on the MTEB leaderboard[5], their size (in number of parameters), and their public availability. To reduce computational cost, we ensured that all experiments could be run on a single *NVIDIA GeForce RTX 2080 Ti* GPU[6]. Table 3 provides an overview of all the embedding models used in the experiments.

Both label names and texts from the training sets are encoded using these models to obtain their embeddings. Given the text embeddings and the embeddings of all labels, we then calculate the cosine similarity between the text embeddings and all label embeddings to obtain distribution θ. We then split θ into two sub-distributions: α and β. Here, α_{ij} represents the similarity score between the text embedding with index i and the corresponding label with index j that is

[2]https://blog.knowledgator.com/finally-a-decent-multi-label-classification-benchmark-is-created-a-prominent-zero-shot-dataset-4d90c9e1c718

[3]https://huggingface.co/datasets/ucirvine/reuters21578

[4]For the AAPD dataset, we mapped the abbreviations to the descriptive names on the arXiv website. When an abbreviation mapped to multiple possible label names, we added the general domain in brackets.

[5]https://huggingface.co/spaces/mteb/leaderboard

[6]Table 11 in Appendix Appendix B shows the computation times for each method.

[7]Model details:
(1) `https://huggingface.co/gist-large-embedding-v0`
(2) `https://huggingface.co/Alibaba-NLP/gte-large-en-v1.5`
(3) `https://huggingface.co/dunzhang/stella_en_400M_v5`
(4) `https://huggingface.co/WhereIsAI/UAE-Large-V1`
(5) `https://huggingface.co/mxbai/Embed-Large-v1`
(6) `https://huggingface.co/BAAI/bge-base-en-v1.5`
(7) `https://huggingface.co/jamesgpt1/sf_model_e5`
(8) `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`
(9) `https://huggingface.co/textgain/TopicAwareSTallmpnetbasev2Wiki`
(10) glove.6B.300d, `https://nlp.stanford.edu/projects/glove/`. Token limit per *sentence-transformers*.

Table 3: Overview of Embedding Models[7]

Model	Size (M)	Embedding Dim.	Max Tokens
(1) Gist-Large [46]	335	1024	512
(2) GTE-Large [47]	434	1024	8192
(3) Stella	435	8192	512
(4) UAE-Large [48]	335	1024	512
(5) Mxbai [49]	335	1024	512
(6) BGE	109	768	512
(7) SF	335	1024	512
(8) all-MPNet	109	768	512
(9) all-MPNet-Wiki [17]	109	768	512
(10) GloVe [36]		300	40,001

part of the true label set $\mathbf{y}_i$. On the other hand, β_{ik} represents the similarity scores between the text embedding with index i and a label with index k that is not part of $\mathbf{y}_i$.

To compare similarity distributions and highlight variability between models (**H1**) and domain-specific variability (**H2**), we normalize the cosine similarities (cf. Figure 2) with min-max normalization (cf. Appendix Appendix A, Figure 8). Given θ, a vector containing cosine similarity scores, we obtain θ' as follows:

$$\theta' = \frac{\theta - \min(\theta)}{\max(\theta) - \min(\theta)}$$

In accordance with the first three hypotheses addressed previously, we conduct t-tests with Bonferroni correction between:

1. Similarity distributions θ from each model, done in a pairwise fashion for each possible pair of models. This way, we can test the hypothesis that different embedding models exhibit unique similarity scales (**H1**).
2. Similarity distributions θ from different datasets, but from the same model. This supports the hypothesis that, within a single model, texts from various genres or domains can demonstrate distinct similarity scales (**H2**).
3. Similarity distributions in α from the same dataset and model, for each possible label pair in a dataset. Thus, we are able to verify the hypothesis that different classes may exhibit unique similarity scales within the semantic space of a single model and dataset (**H3**).

For **H1**, we report the average mean and median cosine similarity percentage of statistically significant ($p < .05$) differences between the cosine similarity distributions of label pairs in a dataset, per model. For **H2**, we report the proportion of dataset pairs with statistically significant differences in their similarity distributions. Finally, for **H3**, we report the proportion of label pairs that exhibit statistically significant differences in similarity distributions.

4.3 Distance-based MLTC

4.3.1 Text Embeddings

Similarly to the exploratory study, we first embed texts and label names using sentence transformers, following the original implementation of BERT-based sentence encoders in [39]. We use the appropriate pooling operation for each model to obtain embeddings. This yields a text embedding batch T. Let $T = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_m\}$ denote the set of text embeddings, where $\mathbf{t}_i \in \mathbb{R}^d$ represents the embedding of the i-th text. T is a tensor with shape (m, h), where m is the number of texts in the dataset, and h is the embedding dimension of a model E.

4.3.2 Label Embeddings

To obtain label embeddings, we explore various methods of representing labels, including standard (original) label names and enhanced representations that improve semantic clarity:

1. **Label Names**: Each label name is used as is from the original dataset. We embed the label names and obtain the label embedding set L.

Table 4: Label Representation Methods

Representation	Example
Label Name	Mechanism
Adjusted Label Name	Biological Mechanisms
Keywords	"Viral Replication", "Pathogenesis", "Immune Response", "Viral Entry", "Host Interaction", "Molecular Biology", "Virus Lifecycle", "Cellular Mechanism", "Infection Mechanism", "Viral Transmission"

2. **Adjusted Label Names**: In some cases, the label names are not semantically distinct enough[8]. To address this, we manually adjust the names to make them more distinct and semantically meaningful within the dataset. We embed the adjusted label names and obtain the label embeddings L. It should be noted that the label names of the SemEval dataset and AAPD dataset were retained without modification[9].
3. **Averaged Keyword Embeddings**: For each (adjusted) label name, we generate 10 additional keywords that are related to the label name. In contrast to [21], we employ GPT-4o. We obtain embeddings for each keyword using sentence encoder E and average all embeddings, including the label name embedding, resulting in the label embedding set L. With this approach, we aim to expand the search spaces for the models without causing overlap between labels in the embedding space.

Let $L = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_n\}$ be the set of label embeddings, where $\mathbf{l}_j \in \mathbb{R}^d$ represents the embedding of the j-th label. L is a tensor with shape (n, h), where n is the number of labels in the dataset and h is the hidden size of embedding model E. Examples of each label representation method can be found in Table 4.

4.3.3 Similarity Calculation and Thresholding

Given the text embeddings T and label embeddings L, we calculate the pairwise cosine similarity between each text embedding and each label embedding. For a given text embedding T_i, we obtain a list of cosine similarity values C_i with length n that contains the similarity values between T_i and all label embeddings in L. We also conduct experiments using Euclidean distance as an alternative similarity metric.

$$\mathbf{C}_i = [\cos(\mathbf{T}_i, \mathbf{l}_1), \cos(\mathbf{T}_i, \mathbf{l}_2), \ldots, \cos(\mathbf{T}_i, \mathbf{l}_n)]$$

As mentioned previously, a threshold θ must be estimated to determine whether a label should be assigned to a text or not. If the similarity between T_i and the label embedding L_j equals or exceeds θ, the corresponding label is assigned to the text. We experiment with multiple thresholding mechanisms to determine θ:

1. **0.5 and Normalized 0.5**: We experiment with two baseline thresholds. First, we use 0.5 as a threshold for all labels, regardless of the model or dataset. Second, we normalize the similarity distributions with min-max normalization as described in Section 5.2.1 and apply 0.5 as a threshold.
2. **Fine-tuned Uniform Thresholds**: With this method, the optimal threshold is determined by selecting the one that maximizes performance on the validation set. We iterate over thresholds from 0.0 to 1.0 in increments of 0.01, applying each threshold to assign labels. The macro-averaged F_1-score is calculated for each threshold and the threshold with the highest score is chosen for final predictions on the test set.
3. **Fine-tuned Label-Specific Thresholds**: With this method (cf. Figure 1), we approach the multi-label classification problem as a series of independent binary classification problems. We iterate over a range of thresholds between 0.0 and 1.0 with an increment of 0.01 and select the optimal threshold for each label separately based on the highest F_1-score on the positive class for classifying the test set. For labels in the test set where no instance appears in the validation set, we assign the average of all fine-tuned thresholds.

4.4 Upper-bound baseline and Zero-shot with LLMs

To compare our results with other existing methods, we fine-tune a *RoBERTa* model [50][10] on the training splits of the datasets. Since the model is fine-tuned on the training data, we hypothesize that it will outperform the distance-based

[8] For example, "Mechanism" in the LitCovid dataset

[9] We prepended each emotion label in SemEval with *Emotional State:*, but this did not yield any improvements.

[10] `https://huggingface.co/FacebookAI/roberta-base`

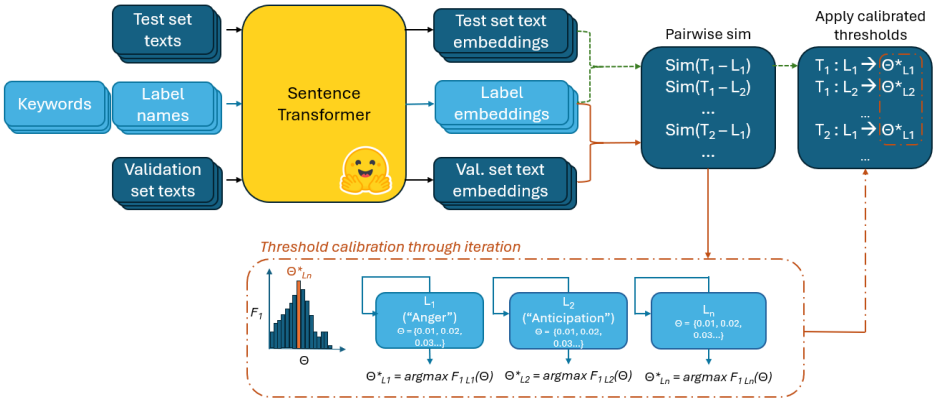

Figure 1: Overview of the thresholding approach and Inference Stage. Texts and label representations are embedded, after which the cosine similarity is used to determine label relevance. Thresholds are optimized based on performance on a validation set.

method. The learning rate is optimized by conducting grid-search experiments. Additionally, we prompt two LLMs, namely *GPT-4o* and *Gemini Pro 1.5* [51][11], to perform zero-shot MLTC using the Binary Relevance method on the test data.

4.5 Evaluation Metrics

For the DBC experiments, we evaluate the performance of the different models and thresholding mechanisms using micro- and macro-averaged F_1 scores. In addition, we calculate the precision at k ($P@k$), where $k = 1$. This metric indicates whether the closest label to the text is a true label.

5 Results and Discussion

5.1 Exploratory Study

The results of our experiments on analyzing the differences in similarity scales across embedding models (**H1**), the impact of domain-specific (inter-dataset) characteristics within a single model (**H2**), and the potential for class-specific similarity scales (**H3**) have partially confirmed our initial hypotheses, through the following findings. We address how each hypothesis is supported in the following sections.

5.1.1 Comparing Similarity Ranges Between Models (H1)

Upon comparing the similarity ranges across various embedding models, we observed notable statistical differences in similarity distributions, challenging the suitability of a fixed universal threshold. Specifically, all models exhibit strong statistically significant differences ($p < .001$ for each possible model combination) in distributions between each other. Additionally, median and mean similarity scores (cf. Table 5[12]) vary considerably among models. For example, *all-MPNet* stands out with a median score of 0.12, substantially staying under the commonly used threshold of 0.5, indicating a lower baseline similarity on the SemEval dataset. In contrast, other models like *GTE-Large* and *GIST-Large* exhibit much higher average similarity, with respective median scores of 0.48 and 0.35 on the same dataset. This variability underlines the need for model-specific threshold settings, as the applicability of a single threshold across diverse models can lead to inaccurate interpretations of semantic similarity.

Upon analyzing the similarity ranges, we considered applying min-max normalization to align the distributions across models. This method transforms scores to a common scale (0 to 1), possibly enabling a unified thresholding approach. While the min-max normalization has forced the similarity distribution to become more alike, the 0.5 threshold remains higher compared to the median of each distribution. For example, normalizing the scores of *GTE-Large* and *all-MPNet* brought their median values closer, making them more comparable.

[11]Due to the limitations of the *Gemini Pro 1.5* API in handling the size of nested objects during function calls, we were unable to process datasets with a larger number of labels, such as AAPD and Reuters.

[12]The results for all models can be found in Table 10, Appendix Appendix A.

These results can be attributed to differences in pre-training data and training methods used in the development of the embedding models. For example, *GIST-Large* is trained on top of the *BGE-Large* model using the MEDI dataset, according to the model's dataset card[13]. In contrast, *Stella* uses distilled embeddings from larger models, such as *Qwen*.

Table 5: (Normalized) Mean and Median Cosine Similarity per Model and Dataset

	SemEval				**BioTech**				**Reuters**				**AAPD**				**LitCOVID**			
model	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)
GIST-Large	0.36	0.31	0.35	0.30	0.32	0.38	0.32	0.38	0.35	0.29	0.34	0.37	0.34	0.37	0.33	0.33	0.32	0.33	0.31	0.29
GTE-Large	0.48	0.44	0.48	0.44	0.45	0.42	0.45	0.43	0.36	0.43	0.35	0.46	0.43	0.48	0.43	0.41	0.44	0.41	0.43	0.44
Stella	0.35	0.25	0.34	0.24	0.28	0.24	0.28	0.23	0.26	0.26	0.25	0.25	0.33	0.35	0.32	0.35	0.33	0.34	0.31	0.27
all-MPNet	0.13	0.30	0.12	0.29	0.14	0.36	0.13	0.35	0.29	0.07	0.27	0.22	0.31	0.10	0.30	0.12	0.33	0.11	0.32	0.09
GloVe	0.23	0.39	0.22	0.38	0.45	0.60	0.46	0.60	0.45	0.19	0.44	0.47	0.60	0.47	0.61	0.48	0.65	0.48	0.65	0.21

5.1.2 Domain-Specific Variability (H2)

The experimental findings also support the hypothesis of domain-specific variability affecting semantic similarity scales within a single model. We find that the differences in distribution across all datasets are statistically significant, except for *Stella*. For this model, only the similarity distribution between SemEval and LitCovid did not differ significantly. Concerning the variability, for instance, the *GTE-Large* model exhibited a median similarity score of 0.48 for the SemEval dataset, while for the Reuters dataset, the median score was lower at 0.35. This variation across domains suggests that even within a single model, the perceived similarity between texts can significantly change depending on the domain. Figure 2 illustrates this domain variability by plotting the cosine similarity between each text in a dataset and all labels. After normalizing scores within each domain, we observed that differences tend to diminish (cf. Appendix Appendix A, Figure 8). For example, normalization could raise the lower median similarity score for the Reuters dataset, making it more comparable to scores from other domains.

The variability across domains can be attributed to differences in the underlying text distributions of pre-training or post-training data. Some strands of text data (domains) may be underrepresented in terms of word frequency compared to others. Since word frequency plays a role in the robustness of model representations [52], this can lead to differences in similarity scores.

5.1.3 Class-specific Similarity Scales (H3)

When comparing the similarity scales across different classes within a single model, we observed statistically significant differences in similarity distributions in most cases across all models and almost all datasets, except for the Reuters dataset (cf. Table 6). This could be explained by the relatively large and detailed label set including closely related labels, causing some labels to be clustered together in the embedding space and resulting in overlapping cosine similarity distributions.

Similar to the results for **H2**, these findings may be explained by the effect of word frequency on how certain words are encoded in a model: infrequent words have weaker representations than common ones, thereby influencing the similarity scores observed in our experiments.

As a result, the experimental findings for **H1–H3** question the suitability of a fixed universal threshold.

5.2 Distance-based MLTC

5.2.1 Comparing Thresholding Approaches (H4)

General Results. Overall, we observe that optimizing thresholds for each label separately consistently yields considerable improvements compared to using a uniform threshold –either the commonly used 0.5 and an optimized– across all evaluated sentence encoders (cf. Table 7[14]). This finding aligns with our exploratory study, indicating that the similarity ranges for labels are unique and should be considered when performing distance-based classification. Optimizing each label threshold helps to enhance classification by capturing the specific characteristics of each label, leading to better overall performance with average improvements of 52% and 14% in macro-F_1 over the 0.5 and optimized uniform threshold, respectively. Regarding the baselines, fine-tuned models show superior performance, followed by zero-shot models, with fine-tuning label-specific thresholds in some cases matching or even exceeding the performance of fine-tuned models.

[13] `https://huggingface.co/avsolatorio/GIST-large-Embedding-v0`

[14] Results for all sentence encoders and for Euclidean distance are provided in Tables 15 and 16 in Appendix Appendix D.

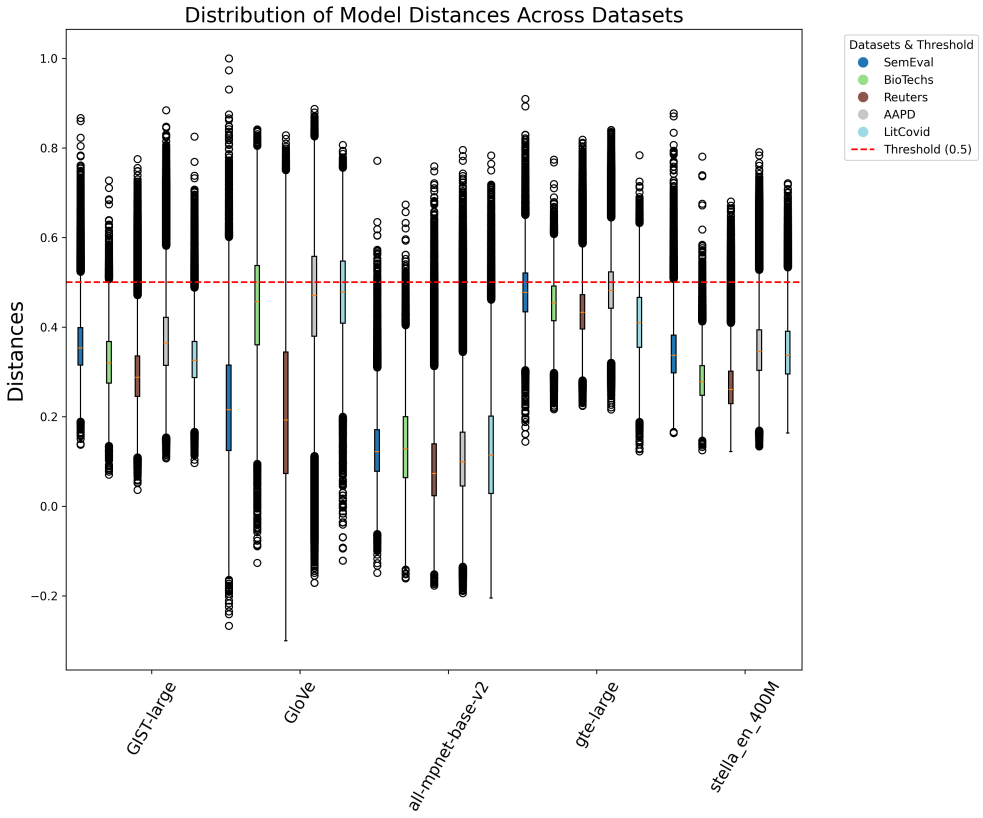

Figure 2: Bar charts showing the variation in (non-normalized) cosine similarity distributions between individual models and domains.

Table 6: Proportion of statistically significant ($p < .05$) differences between cosine similarity distributions of label pairs in a dataset, per model.

Model	SemEval	BioTech	Reuters	AAPD	LitCovid
GIST-Large	0.78	0.40	0.07	0.89	0.81
GTE-Large	0.85	0.35	0.09	0.87	0.95
Stella	0.75	0.50	0.10	0.90	1.00
UAE-Large	0.82	0.35	0.07	0.90	0.95
Mxbai	0.80	0.38	0.07	0.89	0.95
BGE	0.82	0.34	0.09	0.92	1.00
all-MPNet	0.71	0.46	0.10	0.90	1.00
all-MPNet-Wiki	0.67	0.42	0.08	0.88	0.95
GloVe	0.98	0.67	0.28	0.93	0.95

Comparison with Baselines. The results of the classification experiments are summarized in Table 8, which contains the micro-F_1 (miF_1), macro-F_1 (maF_1), and Precision@1P@1 scores for each model and thresholding approach. When comparing distance-based methods (with optimized thresholds for each label) to zero-shot and fine-tuned model baselines, we observe several patterns. When trained on the full training split, *RoBERTa* consistently outperforms all zero-shot and distance-based methods (with optimized thresholds) in terms of maF_1 and miF_1 across the AAPD, and LitCovid datasets. For miF_1, *RoBERTa* also surpasses them on SemEval and Reuters datasets. However, when RoBERTa is trained on a sample, it is generally outperformed by the distance-based method with optimized thresholds in terms of maF_1, except for the LitCovid dataset. For miF_1, this advantage is observed only in the AAPD dataset.

In zero-shot setup, *GPT-4o* achieves the highest maF_1 on the BioTech and Reuters datasets, while *Gemini Pro 1.5* performs best on the SemEval dataset in terms of maF_1. Notably, the distance-based method with optimized thresholds outperforms Gemini Pro 1.5 in miF_1 on the BioTech and LitCovid datasets, and also surpasses *GPT-4o* in miF_1 on the AAPD dataset.

Label-specific vs. Uniform Thresholds. When comparing the uniform fine-tuned threshold with label-specific ones, we observe the largest increases in maF_1 and miF_1 on the BioTech and LitCovid datasets. SemEval, Reuters, and AAPD also demonstrate considerable improvements, though not as substantial as those observed on the previously mentioned datasets. This also highlights that the dynamics between each label and text are unique, which should be taken into account when performing distance-based classification. These interactions are not considered in uniform thresholding methods, as explored in [21]. An exception is observed for *GIST-Large*, which does not benefit from label-specific thresholds on the Reuters dataset. This could be explained by the absence of some labels from the test set in the validation set. Since we estimated the optimal thresholds for these labels by averaging all optimized thresholds, they were therefore not optimally calculated in this case. The $P@1$ scores are the same for both thresholding methods.

In general, our proposed thresholding approach approximates the optimal thresholds, as evidenced in Figure 4. The optimal thresholds are calculated by performing a similar threshold optimization process on the test sets, instead of the validation sets, as described in Section 4.3. For example, the optimized thresholds for "Case Report", "Biological Mechanisms", and "Prevention Strategies" are close to or equal to the optimal thresholds.

Best-performing Embedding Models. Overall, we observe that *Stella* yields the best performance on the Reuters, AAPD, and LitCovid datasets. On the other datasets, *GIST-Large* (SemEval) and *GTE-Large* (BioTech) achieve the best performance. The reason that *GTE-Large* performs best on BioTech can be attributed to its longer context length, as this dataset contains texts that, on average, are longer than 512 tokens (cf. Table 2). The additional information encoded at the ends of the texts therefore still proves useful for classification. Although the performance of non-contextual embedding models such as *GloVe* remains respectable, they are consistently among the worst-performing models (cf. Appendix Appendix D, Table 15).

Table 7: Classification Results of Different Thresholding Methods on All Datasets. An Asterisk (*) Indicates Best Performance with Label Names.

		SemEval			BioTech			Reuters			AAPD			LitCovid			Avg
model	**thr**	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**avg ma**F_1
GIST-Large	0.5	42.2	48.58	67.51	18.6	19.3	28.1	32.8	37.92	50.55	28.46	30.02	64.7	35.35	42.95	61.07	31.48
	n0.5	42.94	49.53	67.51	15.43	19.0	28.1	12.3	15.81	50.55	30.89	32.64	64.7	40.76	48.47	61.07	28.46
	unif	44.99	51.39	67.51	18.54	21.5	28.1	**32.8**	37.92	50.55	38.21	43.11	64.7	40.32	47.41	61.07	34.97
	lbl	**47.59**	**55.0**	**67.51**	**23.29**	**33.0**	**28.1**	31.6*	**38.02***	**50.55***	**41.06***	**49.97***	**64.3***	**51.34***	**56.31***	**44.96***	**38.98**
GTE-Large	0.5	37.25	39.42	61.95	13.95	16.7	26.8	8.3	11.44	48.76	17.51*	18.91*	61.0*	38.82	44.07	60.65	23.17
	n0.5	41.23	45.84	61.95	13.32	15.9	26.8	8.86	12.2	48.76	23.98	27.02	62.0	39.76	47.55	60.65	25.43
	unif	41.35	46.79	61.95	19.56	21.0	26.8	33.2*	38.13*	48.76*	33.88	39.49	62.0	43.64*	48.53*	51.16*	34.33
	lbl	**44.01**	**50.87**	**61.95**	**26.18***	**36.6***	**26.8***	**34.5***	**41.31***	**48.76***	**37.41**	**40.13**	**62.0**	**48.79***	**55.71***	**51.16***	**38.18**
Stella	0.5	28.91	32.37	61.83	6.29	2.34	28.9	29.2	29.41	69.04	35.19	43.68	64.1	29.2	28.36	55.47	25.76
	n0.5	25.62	28.45	61.83	18.03	19.9	28.9	20.9	29.83	69.04	31.81*	35.07*	56.8*	43.68	46.43	55.47	28.01
	unif	41.54	45.9	61.83	18.25	19.4	28.9	36.5*	54.26*	69.04*	35.19	43.68	64.1	44.5	47.4	55.47	35.20
	lbl	**42.54**	**48.66**	**61.83**	**26.02**	**35.3**	**28.9**	**37.1***	**55.54***	**69.04***	**42.41**	**48.16**	**64.1**	**57.71**	**63.88**	**55.47**	**41.16**
UAE-Large	0.5	42.08*	48.1*	64.34*	16.85	20.7	25.2	22.7	27.39	46.44	12.85*	14.09*	62.5*	31.44	33.96	56.61	25.18
	n0.5	40.98	45.65	63.82	15.15	18.4	25.2	12.1	16.0	46.44	21.78	22.68	64.0	35.58	41.7	56.61	25.12
	unif	42.1	48.69	63.82	17.95	20.8	25.2	31.4*	32.4*	46.44*	33.28	38.15	64.0	35.36	40.97	56.61	32.02
	lbl	**44.05***	**53.07***	**64.34***	**22.55**	**32.49**	**25.2**	**34.0***	**35.37***	**46.44***	**38.6**	**43.86**	**64.0**	**45.77***	**53.89***	**39.77***	**36.99**
BGE	0.5	39.51*	42.56*	57.32*	15.57*	17.18*	17.85*	11.4*	13.34	51.04	11.03*	11.69*	49.0*	34.94*	38.4*	53.55*	22.49
	n0.5	40.35*	43.83*	57.32*	15.13*	17.3*	17.85*	8.36	9.95	51.04	23.39	24.09	49.6	35.4*	39.3*	53.55*	24.53
	unif	39.31	44.93	61.64	16.77	19.5	21.5	32.1*	35.82*	51.04*	31.62	36.64	61.4	40.53*	45.24*	53.55*	32.07
	lbl	**42.68**	**48.97**	**61.64**	**21.63**	**30.4**	**21.5**	**33.0***	**43.43***	**51.04***	**37.05**	**43.97**	**61.4**	**48.08***	**56.53***	**53.55***	**36.49**

Effect of Sample Size. Additionally, we conducted learning curve experiments by taking five random stratified splits [45] of the validation data for each sample size to evaluate how much data is needed to obtain accurate thresholds. We calculate label-specific thresholds as before, but on smaller samples of the validation data. The results –as summarised in Figure 3– suggest that more data leads to more accurate thresholds and, therefore, better performance. Notably, for all datasets except Reuters, label-specific thresholds derived from just 50-100 validation samples achieve equal or superior maF_1 scores compared to an optimized uniform threshold derived from the entire validation dataset. Using only 10 examples per label achieves 74% of the full method's performance, while 100 examples reach 91% of full performance. These results suggest that our proposed method is more attractive for scenarios where limited annotated data is available.

Table 8: Classification Results from Fine-tuned models and Generative LLMs on all Datasets, Compared with Best-Performing Distance-Based Classification (DBC) Model.

	SemEval		BioTech		Reuters		AAPD		LitCovid		Avg	
approach/model	**ma**F_1	**mi**F_1	**ma**F_1	**mi**F_1	**ma**F_1	**mi**F_1	**ma**F_1	**mi**F_1	**ma**F_1	**mi**F_1	**avg ma**F_1	**avg mi**F_1
DBC	47.59	55.0	26.18	36.6	37.1	55.54	41.06	49.97	57.71	63.88	41.93	52.20
Roberta (sample)	44.22	63.83	16.64	**47.92**	19.2	83.23	11.77	47.99	72.45	86.10	32.86	65.81
Roberta	51.83	**69.40**	13.23	47.46	38.7	**89.12**	**52.99**	**72.02**	**81.93**	**88.28**	47.74	**73.26**
GPT-4o	52.72	60.99	**36.84**	43.4	**59.5**	76.01	42.72	45.62	63.6	66.35	**51.08**	58.47
Gemini Pro 1.5	**54.73**	59.24	33.9	34.5	N/A	N/A	N/A	N/A	49.72	56.46	N/A	N/A

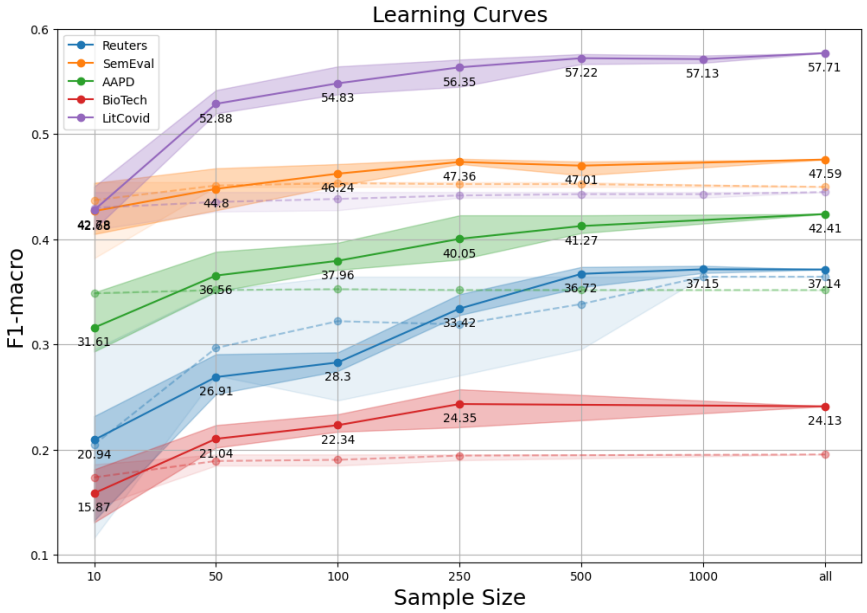

Figure 3: Results from learning curve experiments with the best performing model on each dataset (*GIST-Large* for SemEval, *GTE-Large* for BioTech and *Stella* for all other datasets). The dashed lines show the results conducted with uniform fine-tuned thresholds. Five random samples for each size are taken.

5.2.2 Effect of Label Representation Methods

In general, we observe that enhancing label representations either by adjusting the label names or generating additional keywords improves classification results. Adjusting label names aids the separation between labels in the embedding space, therefore leading to better performance on all datasets. Tables 12 and 13 in Appendix Appendix C show the differences in performance.

The averaged keyword embedding as a label representation often yields improvements, suggesting that the centroid of the label and its corresponding keywords provides a more robust representation of the label's meaning. The most notable improvements are observed in terms of $P@1$, regardless of whether keywords yield better F_1-scores for a specific model. For a complete overview of results regarding label representations, consult Table 14 in Appendix Appendix C. We expected that the relative score increase between label names and keyword embeddings would be the highest for word embedding models [16], but we did not observe any notable difference compared to other, more advanced sentence transformers. However, *GloVe* benefits more consistently from keyword embeddings than other sentence transformers.

An exception to this is the Reuters dataset, where no model benefits from generated keywords, likely due to the large label space of the dataset. Since all labels belong to the same domain, the keywords generated by an LLM are likely to introduce semantic overlap between labels on the one hand, or be too generic for the label name, thereby introducing noise[15].

While these findings generally indicate that keyword-based representations can enhance performance, the effect is not uniform across datasets or models. For example, *GIST-Large* benefits from averaged keyword embeddings on SemEval and BioTech, but not on AAPD or LitCovid. Similarly, *GTE-Large* yields better results with averaged keyword representations on all datasets except BioTech and LitCovid. This could be explained by the fact that the quality of the label names is already satisfactory for those models to perform classification, and the keywords may introduce noise.

5.2.3 Error Analysis

Similarity Distributions. To further analyze the results of the experiments and the effectiveness of our proposed thresholding approach, we plot the cosine similarities between all text embeddings from the test data and the label

[15]We experimented with filtering the keywords further using LLMs, though this did not yield any improvements.

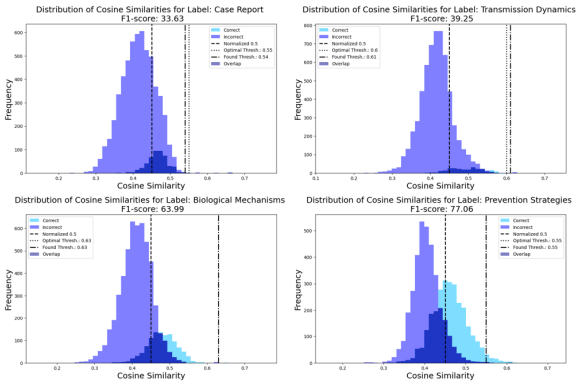

Figure 4: Distributions of cosine similarity scores per label of the LitCovid dataset, obtained using *Stella*.

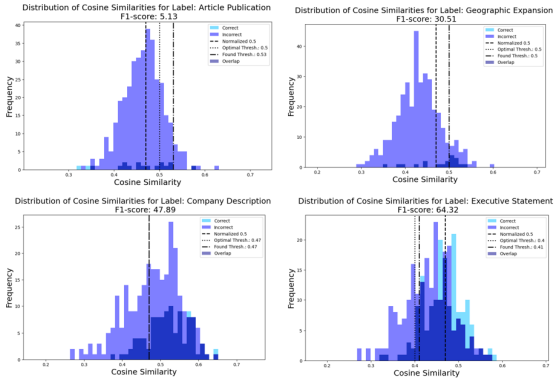

Figure 5: Distributions of cosine similarity Scores per label of the BioTech dataset, obtained using *GIST-Large*.

embeddings. Given label i, the similarities are divided into two groups: similarities between label i's embedding and texts where i is a true label (α), and similarities with texts where i is not a true label (β). By plotting the similarities this way (cf. Figures 4 and 5), we can visualize the overlap between similarity distributions and, consequently, assess the effectiveness and feasibility of applying label-specific thresholds. For LitCovid (Figure 4), it can be observed that a clearer separation between α and β leads to a higher F_1-score for a given label. The Pearson correlation coefficient between the overlap per class –as shown in Figure 4– and the F_1-scores per class indicates that there is a very strong negative correlation (-0.97). We also find similar strong correlations for SemEval (-0.89) and AAPD (-0.71), in addition to weaker negative correlations for Reuters (-0.44) and BioTech (-0.55).

While distance-based classification works well on the LitCovid dataset, this classification method is more challenging on BioTech, as evidenced by the low performance scores from each model (cf. Table 7). It should also be noted that this dataset is challenging for LLMs and fine-tuned transformers as well. The challenge for distance-based methods stems from the completely overlapping distributions (Figure 5), rendering it virtually impossible to select a threshold that can reliably separate texts where the label is true from those where it is false. However, the number of instances per class also affects the performance on the BioTech dataset: the more frequently a label occurs, the higher the F_1-score. This is further supported by a high Pearson correlation coefficient of 0.77.

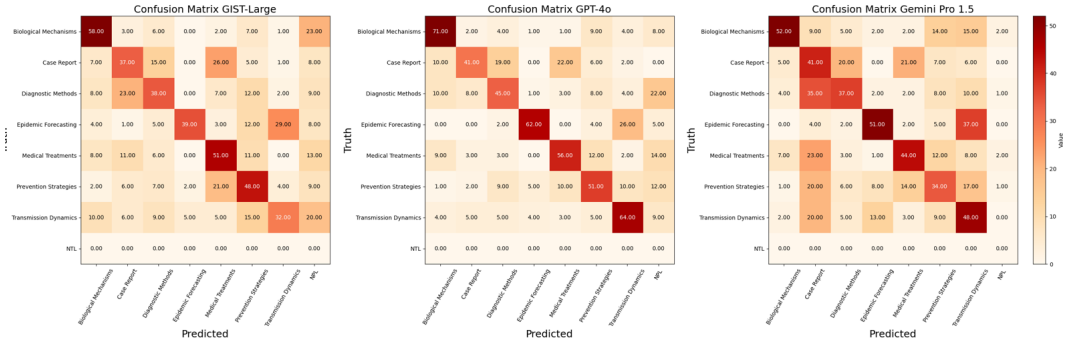

Figure 6: Confusion Matrices of *Stella*, *GPT-4o* and *Gemini Pro 1.5* on the LitCovid Dataset.

Confusion Matrices. We visualize the confusion matrices obtained from the best-performing sentence encoder, *GPT-4o*, and *Gemini Pro 1.5* using the method proposed by [53]. In contrast to confusion matrices from binary or multi-class classification problems, an extra row and column are added, respectively indicating cases where no true labels are present or no labels are predicted. For LitCovid, we observe that the distance-based method makes similar mistakes to *GPT-4o*. For example, both models frequently confuse "Case Report" with "Medical Treatments", "Epidemic Forecasting" with "Transmission Dynamics", and "Preventive Strategies" with "Medical Treatments". As evidenced by the lower performance of *Gemini Pro 1.5* shown in Table 7, the model confuses more labels with each other, which is particularly apparent for "Case Report". This label appears to serve as a fallback during the model's prediction process.

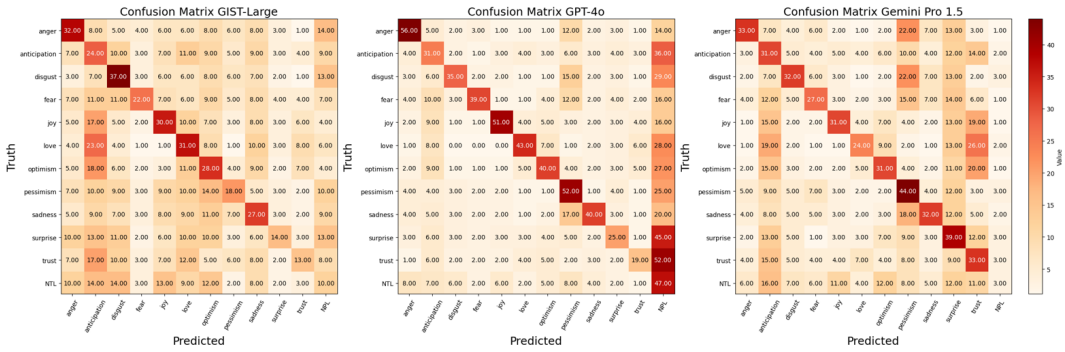

Figure 7: Confusion Matrices of *Stella*, *GPT-4o* and *Gemini Pro 1.5* on the SemEval Dataset.

Table 9: Average cosine similarities (Stella) between labels and all other labels.

Label Name	Cos. Sim.
anger	0.66
anticipation	**0.72**
disgust	0.67
fear	0.69
joy	0.70
love	0.69
optimism	0.69
pessimism	0.69
sadness	0.68
surprise	0.67
trust	0.67

On the SemEval dataset, we observe that the distance-based method underperforms compared to LLMs. One reason, according to the confusion matrix in Figure 7 is that the model often confuses the label "anticipation" with other labels. It could be hypothesized that this label is close to other labels in the embedding space due to the abstract nature of the emotion, especially compared to other emotions like "anger". To verify this, we embedded the emotion labels (and corresponding keywords) using *GIST-Large* and calculated the average cosine similarity to all other labels in the dataset (cf. Table 9). We found that, indeed, "anticipation" is more similar to other labels, albeit by a slight margin. This could cause confusion during classification, thereby introducing a high number of false positives. Interestingly, *Gemini Pro 1.5* also experiences difficulties with the same label, in addition to "trust", "surprise", and "pessimism". *GPT-4o*, on the other hand, excels at predicting correct subsets of true labels, as evidenced by the highlighted "NPL" column. For other datasets, we also observe that certain labels are too general and accommodate other labels during classification. We further observe that *GPT-4o* excels at predicting correct subsets of true labels.

In general, we conclude that the introduced false positives can be attributed to the representations of texts and labels from sentence transformers, which seem to struggle with capturing the general semantics of a text. This causes the overlap in distributions, as discussed in Section 5.2.3. This issue is exemplified in a misclassified instance from the Reuters dataset:

Text: Shr loss five cts vs profit 10 cts // Net loss 381,391 vs profit 736,974 // Revs 6,161,391 vs 9,241,882 // NOTE: Canadian dollars. // Proved oil reserves at year-end 3.3 mln barrels, up 39 pct from a year earlier, and natural gas reserves 4.7 billion cubic feet, off nine pct.
True: Earnings and Forecasts. **Pred.**: Canadian Dollar, Crude Oil, Earnings and Forecasts, Natural Gas

The reason the model predicts these labels is likely due to the presence of (partial) label names in the text. However, since these labels are not annotated, they are regarded as false positives, thereby mirroring the findings from [17]. Sentence transformers only manage to encode shallow semantics of the texts and fail to capture their general meaning.

Another false positive is observed in the following example, where a label name partially occurs in the text itself ("South-African"). However, the actual true label is only implied, which is challenging for distance-based models:

Text: Six black miners have been killed and two injured in a rock fall three km underground at a South African gold mine, the owners said on Sunday. Rand Mines Properties Ltd, one of South Africa's big six mining companies [...]
True: Gold. **Preds.**: South African Rand

A similar mistake is observed in the SemEval dataset, where the model predicts "anger" for the following tweet due to the presence of the word "furious", which is actually part of a movie title:

Text: Fast and furious marathon soon!
True: anticipation, joy, optimism. **Pred.**: anger, anticipation, joy

These mistakes highlight that the models are sensitive to context and (partial) matches of label names in texts, which may explain the higher performance of both fine-tuned models and LLMs on some datasets.

6 Conclusion

In this paper we investigated label-specific thresholds for distance-based MLTC, an under-explored yet computationally efficient classification method that is particularly well-suited for scenarios where there is insufficient data to train custom embeddings and classifiers. We first researched the possibility of label-specific thresholds by experimenting with several state-of-the-art sentence encoders on multiple publicly available datasets, and analyzed cosine similarity distributions. We found statistically significant differences between models (inter-model), datasets (intra-model), and labels (intra-model). These findings challenge the applicability of uniform thresholds for classification, and to a certain extent, for information retrieval tasks. We further applied these findings to classification experiments where we optimized thresholds based on annotated validation sets. The results showed that label-specific thresholds clearly outperform uniform thresholds and (normalized) 0.5, even when optimized on smaller data samples. Moreover, our method matches or even surpasses the performance of zero-shot LLMs on certain datasets.

7 Limitations and Future Work

This study is subject to a few limitations. First and foremost, while our proposed method yields promising results, it still falls behind LLMs and fine-tuned models on some datasets, with the largest performance gap observed on the Reuters dataset. Second, the proposed label representation method of averaging keyword embeddings yields inconsistent results, which limits its applicability to other datasets.

In addition, because of the black-box nature of the employed sentence encoders and LLMs, it cannot be determined whether the datasets have already been seen by the models during their pre-training phase. The presented results might therefore represent an overestimation of their performance. Nonetheless, our extensive evaluation of a multitude of models and sentence encoders trained from scratch has shown the effectiveness of our thresholding approach.

The performance of our thresholding approach is mainly limited by the separation between distributions in Figure 4 and 5. Better label representations or training regimes might encourage the models to more clearly separate similarity distributions between labels, therefore leading to better classification performance. Future work could explore more label representations or the use of contrastive learning to improve the representations of texts and labels. Additionally, we found that DBC is sensitive to (partial) matches of label names in texts, which can negatively affect the performance. Future work could focus on improving the encoding to capture deeper semantics, potentially overcoming this limitation.

8 Acknowledgments

This research was funded by Flanders Innovation & Entrepreneurship (VLAIO), grant HBC.2021.0222 and by the Flemish government under FWO IRI project CLARIAH-VL.

Appendix A Similarity Scale Variation Across Models (Exploratory Study)

Table 10 contains the (normalized) mean and median cosine similarity scores for all models across all datasets.

Table 10: (Normalized) Mean and Median Cosine Similarity per Model and Dataset

	SemEval				BioTech				Reuters				AAPD				LitCOVID			
model	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)	mn	mn (norm)	mdn	mdn (norm)
GIST-Large	0.36	0.31	0.35	0.30	0.32	0.38	0.32	0.38	0.35	0.29	0.34	0.37	0.34	0.37	0.33	0.33	0.32	0.33	0.31	0.29
Gte-Large	0.48	0.44	0.48	0.44	0.45	0.42	0.45	0.43	0.36	0.43	0.35	0.46	0.43	0.48	0.43	0.41	0.44	0.41	0.43	0.44
Stella	0.35	0.25	0.34	0.24	0.28	0.24	0.28	0.23	0.26	0.26	0.25	0.25	0.33	0.35	0.32	0.35	0.33	0.34	0.31	0.27
UAE-Large	0.46	0.34	0.46	0.34	0.41	0.42	0.41	0.42	0.38	0.38	0.37	0.44	0.41	0.51	0.40	0.46	0.36	0.46	0.36	0.39
Mxbai	0.48	0.38	0.48	0.37	0.41	0.42	0.41	0.42	0.40	0.38	0.40	0.46	0.40	0.49	0.39	0.46	0.37	0.46	0.37	0.39
BGE	0.50	0.44	0.50	0.44	0.46	0.40	0.46	0.40	0.35	0.43	0.34	0.45	0.40	0.53	0.40	0.52	0.41	0.52	0.41	0.43
SF	0.44	0.42	0.45	0.42	0.31	0.41	0.31	0.40	0.34	0.30	0.33	0.34	0.42	0.41	0.41	0.35	0.36	0.35	0.36	0.30
all-MPNet	0.13	0.30	0.12	0.29	0.14	0.36	0.13	0.35	0.29	0.07	0.27	0.22	0.31	0.10	0.30	0.12	0.33	0.11	0.32	0.09
all-MPNet-Wiki	0.34	0.33	0.33	0.32	0.32	0.43	0.32	0.42	0.34	0.23	0.32	0.38	0.47	0.39	0.46	0.41	0.42	0.41	0.42	0.25
GloVe	0.23	0.39	0.22	0.38	0.45	0.60	0.46	0.60	0.45	0.19	0.44	0.47	0.60	0.47	0.61	0.48	0.65	0.48	0.65	0.21

Figure 8 shows the variability of cosine similarity scales across models and datasets after normalization.

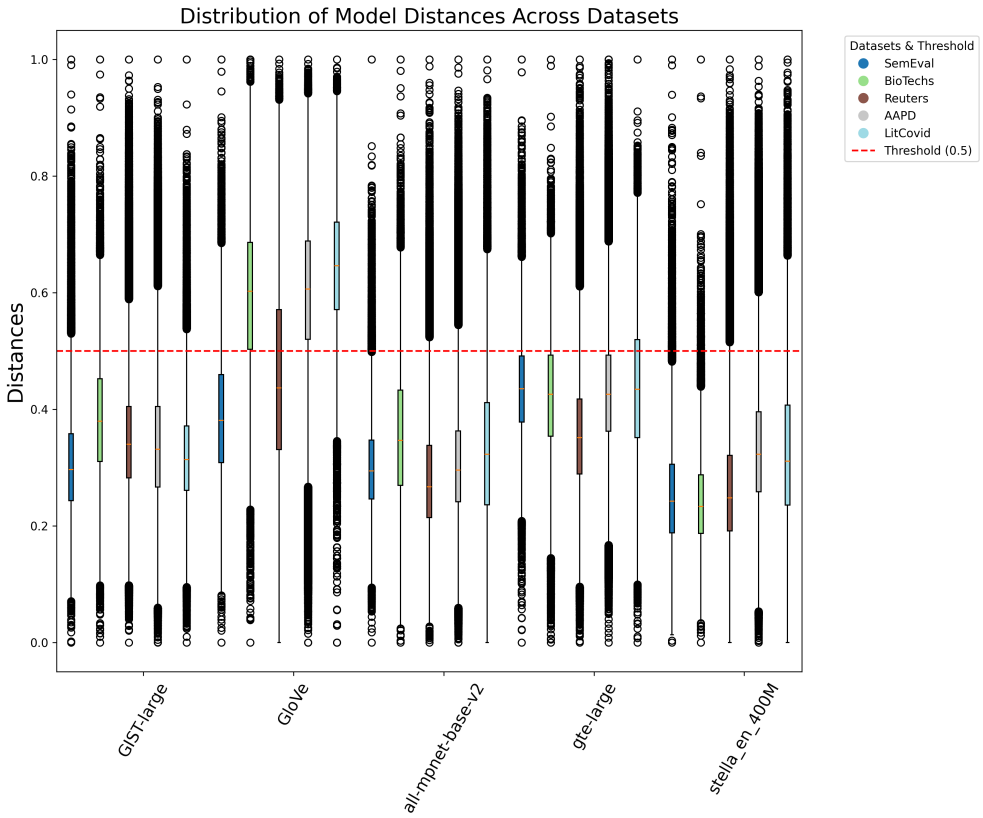

Figure 8: Bar charts showing the variation in normalized cosine similarity distributions between individual models and domains.

Appendix B Computation Times

The total computation time, calculated as the sum of fine-tuning or threshold calibration and inference process across all datasets, is summarised in Table 11. It should be noted that, though iterating over thresholds for uniform thresholds is evidently much faster, more computational resources are spent on calculating the macro-averaged F_1 score for each threshold as opposed to the F_1 score of the positive class for label-specific thresholds. The batch size (8) remained consistent across all experiments.

Table 11: Sum of Training + Inference Times (in hours) across Datasets, with Consistent Batch Size (8).

Model	Lbl. rep.	Thresh.	Train + Inf. (hrs)
RoBERTa (sample)	/	/	5.44
Stella	lbl. name	lbl.	0.24
Stella	kw	lbl.	0.25
Stella	kw	unif.	0.26

Appendix C Evaluation of Label Representations

Table 12 compares the performance of different label representations methods (out-of-the-boxlabel names and adjusted label names). The results are obtained using the best performing model for each dataset. Table 13 shows a per-class comparison between the two representation methods for the LitCovid dataset. Table 14 compares the results with label names and generated keywords as label representations.

Table 12: Comparison between the Original Label Names and Adjusted Label Names (Best Model for each Dataset).

	Lbl Names		**Adj. Lbl Names**	
Dataset	maF_1	miF_1	maF_1	miF_1
BioTech	24.36	36.86	27.15	36.97
Reuters	46.53	60.79	47.63	62.69
LitCovid	54.82	58.07	61.68	58.30

Table 13: Comparison between the Original Label Names and Adjusted Label Names on the LitCovid dataset (Stella)

Label Name	**pr**	**rec**	**F1**	**Adj. Label Name**	**pr**	**rec**	**F1**
Case Report	45.51	59.96	51.75	Case Report	45.51	59.96	51.75
Diagnosis	56.37	**72.70**	63.50	Diagnostic Methods	**64.29**	65.91	**65.09**
Epidemic Forecasting	59.26	75.00	66.21	Epidemic Forecasting	59.26	75.00	66.21
Mechanism	27.43	54.52	36.49	Biological Mechanisms	**58.99**	**70.92**	**64.41**
Prevention	61.26	**80.15**	69.44	Prevention Strategies	**67.90**	79.67	**73.31**
Transmission	**33.88**	**48.05**	**39.74**	Transmission Dynamics	27.32	43.75	33.63
Treatment	**49.86**	65.52	**56.63**	Medical Treatments	38.28	**90.03**	53.72
micro avg	49.85	69.54	58.07	micro avg	**51.69**	**76.45**	**61.68**
macro avg	47.65	65.13	54.82	macro avg	**51.65**	**69.32**	**58.30**

Table 14: Comparison between Label Names and Keyword Embeddings as Label Representations for Optimized Label-Specific Thresholds.

		SemEval			BioTech			Reuters			AAPD			LitCovid		
model	**rep.**	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$
GIST-Large	lbl.	45.15	53.37	**68.18**	22.56	32.48	20.73	**31.59**	**38.02**	50.55	41.06	**49.97**	64.3	**51.34**	56.31	44.96
	kw	**47.59**	**55.00**	67.51	**23.29**	**32.99**	**28.08**	29.31	36.63	**52.10**	**41.78**	46.84	**64.7**	48.31	**56.76**	**61.07**
GTE-Large	lbl.	41.20	49.85	**63.42**	**26.18**	**36.63**	26.77	**34.47**	**41.31**	48.76	35.95	40.24	61.0	**48.79**	**55.71**	51.16
	kw	**44.01**	**50.87**	61.95	24.13	35.65	**26.77**	30.39	41.10	**48.86**	**37.41**	**40.13**	**62.0**	47.76	55.14	**60.65**
Stella	lbl.	41.20	**48.96**	**65.91**	**26.10**	34.45	23.10	**37.14**	**55.54**	69.04	41.65	46.60	56.8	**58.30**	61.68	18.67
	kw	**42.54**	48.66	61.83	26.02	**35.27**	**28.87**	33.53	47.20	**69.24**	**42.41**	**48.16**	**64.1**	57.71	**63.88**	**55.47**
UAE-Large	lbl.	**44.05**	**53.07**	**64.34**	**22.55**	**32.49**	22.57	**34.01**	35.37	46.44	36.40	**44.24**	62.5	**45.77**	**53.89**	39.77
	kw	43.96	51.65	63.82	22.42	31.78	**25.20**	28.30	**37.08**	**48.03**	**38.60**	43.86	**64.0**	42.11	53.05	**56.61**
Mxbai	lbl.	**44.42**	**54.52**	64.50	21.74	31.03	23.88	**32.88**	33.84	47.14	38.84	**47.18**	62.8	**48.48**	**54.97**	42.15
	kw	43.93	50.82	**64.68**	**25.23**	**33.75**	**27.82**	28.10	**34.32**	**50.88**	**39.79**	44.92	**63.6**	45.44	54.94	**59.22**
BGE	lbl.	42.10	48.62	57.32	19.72	**30.75**	17.85	**32.97**	**43.43**	51.04	34.77	40.61	49.6	**48.08**	**56.53**	53.55
	kw	**42.68**	**48.97**	**61.64**	**21.63**	30.41	**21.52**	26.07	36.85	**55.44**	**37.05**	**43.97**	**61.4**	45.75	54.27	**62.40**
SF	lbl.	**42.31**	**53.20**	61.46	**22.16**	31.12	16.54	**32.74**	**43.03**	45.05	**33.73**	40.75	**66.2**	**49.19**	56.56	35.60
	kw	41.03	50.41	**64.41**	21.02	**32.51**	**25.46**	28.93	42.65	**51.21**	33.26	**43.08**	63.1	47.22	**56.99**	**61.82**
all-MPNet	lbl.	38.73	49.77	**59.71**	20.71	25.37	17.85	**30.49**	**41.76**	49.72	37.06	41.95	62.1	**53.41**	**59.98**	19.79
	kw	**40.59**	**51.91**	57.04	**22.26**	**26.07**	**22.05**	27.52	39.29	**59.21**	**39.70**	**46.95**	**66.9**	51.32	56.68	**46.91**
all-MPNet-Wiki	lbl.	37.90	46.13	51.40	21.04	26.86	19.69	**28.57**	**48.70**	**61.40**	37.38	45.11	61.2	49.37	57.36	17.17
	kw	**38.65**	**46.87**	**56.55**	**22.62**	**27.95**	**19.69**	25.48	39.80	60.77	**39.43**	**45.13**	**65.0**	**51.14**	**58.00**	**37.81**
GloVe	lbl.	36.22	42.53	28.69	20.45	28.46	7.61	**18.70**	**23.88**	28.48	19.00	25.14	12.0	38.43	44.26	**43.31**
	kw	**37.17**	**44.18**	**31.91**	**20.94**	**30.97**	**10.24**	15.90	23.05	**29.90**	**22.07**	**30.00**	**32.3**	**44.78**	**56.67**	28.71

Appendix D Complete Classification Results

Table 15 contains the classification results for all thresholding methods for all models.

Table 15: Classification Results on All Datasets. An Asterisk (*) Indicates Best Performance with Label Names.

		SemEval			BioTech			Reuters			AAPD			LitCovid		
model	**thr**	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$	**ma**F_1	**mi**F_1	$P@1$
GIST-Large	0.5	42.2	48.58	67.51	18.6	19.3	28.1	32.8	37.92	50.55	28.46	30.02	64.7	35.35	42.95	61.07
	n0.5	42.94	49.53	67.51	15.43	19.0	28.1	12.3	15.81	50.55	30.89	32.64	64.7	40.76	48.47	61.07
	unif	44.99	51.39	67.51	18.54	21.5	28.1	**32.8**	37.92	50.55	38.21	43.11	64.7	40.32	47.41	61.07
	lbl	**47.59**	**55.0**	**67.51**	**23.29**	**33.0**	**28.1**	31.6*	**38.02***	**50.55***	**41.06***	**49.97***	**64.3***	**51.34***	**56.31***	**44.96***
GTE-Large	0.5	37.25	39.42	61.95	13.95	16.7	26.8	8.3	11.44	48.76	17.51*	18.91*	61.0*	38.82	44.07	60.65
	n0.5	41.23	45.84	61.95	13.32	15.9	26.8	8.86	12.2	48.76	23.98	27.02	62.0	39.76	47.55	60.65
	unif	41.35	46.79	61.95	19.56	21.0	26.8	33.2*	38.13*	48.76*	33.88	39.49	62.0	43.64*	48.53*	51.16*
	lbl	**44.01**	**50.87**	**61.95**	**26.18***	**36.6***	**26.8***	**34.5***	**41.31***	**48.76***	**37.41**	**40.13**	**62.0**	**48.79***	**55.71***	**51.16***
Stella	0.5	28.91	32.37	61.83	6.29	2.34	28.9	29.2	29.41	69.04	35.19	43.68	64.1	29.2	28.36	55.47
	n0.5	25.62	28.45	61.83	18.03	19.9	28.9	20.9	29.83	69.04	31.81*	35.07*	56.8*	43.68	46.43	55.47
	unif	41.54	45.9	61.83	18.25	19.4	28.9	36.5*	54.26*	69.04*	35.19	43.68	64.1	44.5	47.4	55.47
	lbl	**42.54**	**48.66**	**61.83**	**26.02**	**35.3**	**28.9**	**37.1***	**55.54***	**69.04***	**42.41**	**48.16**	**64.1**	**57.71**	**63.88**	**55.47**
UAE-Large	0.5	42.08*	48.1*	64.34*	16.85	20.7	25.2	22.7	27.39	46.44	12.85*	14.09*	62.5*	31.44	33.96	56.61
	n0.5	40.98	45.65	63.82	15.15	18.4	25.2	12.1	16.0	46.44	21.78	22.68	64.0	35.58	41.7	56.61
	unif	42.1	48.69	63.82	17.95	20.8	25.2	31.4*	32.4*	46.44*	33.28	38.15	64.0	35.36	40.97	56.61
	lbl	**44.05***	**53.07***	**64.34***	**22.55**	**32.49**	**25.2**	**34.0***	**35.37***	**46.44***	**38.6**	**43.86**	**64.0**	**45.77***	**53.89***	**39.77***
Mxbai	0.5	42.89*	47.76*	64.5*	17.91*	19.31*	23.88*	22.14*	24.85*	47.14*	15.35*	16.67*	62.8*	33.29*	35.67*	42.15*
	n0.5	39.86	46.43	64.5	15.79	19.34	27.82	9.98*	11.62*	47.14*	23.5	24.44	63.6	37.33	43.37	59.22
	unif	42.89	49.16	64.68	20.2	22.7	27.8	30.6*	**35.08***	47.14*	34.99	39.72	63.6	38.81*	40.49*	42.15*
	lbl	**44.42***	**54.52***	**64.50***	**25.23**	**33.8**	**27.8**	**32.9***	33.84*	**47.14***	**39.79**	**44.92**	**63.6**	**48.48***	**54.97***	**42.15***
BGE	0.5	39.51*	42.56*	57.32*	15.57*	17.18*	17.85*	11.4*	13.34	51.04	11.03*	11.69*	49.0*	34.94*	38.4*	53.55*
	n0.5	40.35*	43.83*	57.32*	15.13*	17.3*	17.85*	8.36	9.95	51.04	23.39	24.09	49.6	35.4*	39.3*	53.55*
	unif	39.31	44.93	61.64	16.77	19.5	21.5	32.1*	35.82*	51.04*	31.62	36.64	61.4	40.53*	45.24*	53.55*
	lbl	**42.68**	**48.97**	**61.64**	**21.63**	**30.4**	**21.5**	**33.0***	**43.43***	**51.04***	**37.05**	**43.97**	**61.4**	**48.08***	**56.53***	**53.55***
SF	0.5	38.32*	44.47*	61.46*	15.06	12.7	25.5	32.1	31.11	45.05	27.54*	32.99*	66.2*	36.87	44.21	61.82
	n0.5	38.85*	44.87*	61.46*	15.33	19.8	25.5	15.0	18.81	45.05	18.3	20.52	66.2	39.08	46.47	61.82
	unif	41.29	47.77	70.17	15.74	18.4	22.8	31.8*	32.42*	45.05*	31.29	35.02	63.1	42.03	49.18	61.82
	lbl	**42.31***	**53.20***	**64.41***	**20.74**	**34.6**	**22.8**	**32.7***	**43.03***	**45.05***	**33.26***	**43.08**	**63.1**	**47.22**	**56.99**	**61.82**
all-MPNet	0.5	1.38	1.64	57.04	5.96	2.08	22.05	9.62*	6.34*	49.72*	10.46	10.33	66.9	8.64	4.64	46.91
	n0.5	29.16	33.88	57.04	14.58	18.46	22.05	20.8*	22.77*	49.72*	32.73	38.62	66.9	36.31	42.64	46.91
	unif	39.14	44.28	57.04	12.59	16.0	22.05	27.2*	34.12*	49.52*	35.88	43.13	66.9	41.0*	43.94*	20.12*
	lbl	**40.59**	**51.91**	**57.04**	**22.26**	**26.07**	**22.05**	**30.6**	**39.71**	**49.52**	**39.68**	**46.96**	**66.9**	**53.56**	**60.15**	**20.12**
all-MPNet-Wiki	0.5	29.76	33.91	56.55	16.67	19.07	19.69	22.14*	31.68*	61.4*	29.2	32.72	61.2	35.84	37.36	37.81
	n0.5	36.12	40.78	56.55	15.22	17.11	19.69	9.85*	12.7*	61.4*	20.29	21.34	61.2	36.66*	41.65*	17.17*
	unif	37.08	41.04	56.55	16.13	19.0	19.7	26.2*	43.43*	60.77*	37.66	42.32	66.1	43.15*	48.63*	23.99*
	lbl	**38.65**	**46.87**	**56.55**	**22.62**	**27.95**	**19.7**	**28.3***	**49.23***	**60.77***	**39.63**	**47.46**	**66.1**	**53.05**	**60.46**	**43.69**
GloVe	0.5	18.64	20.06	31.91	11.44*	14.52*	7.61*	10.3	9.58	28.48	12.35*	13.98*	12	35.61*	43.86*	43.31*
	n0.5	32.42	34.71	31.91	12.73*	14.62*	7.61*	5.97	5.26	28.48	9.13	9.86	32.3	30.54*	34.22*	43.31*
	unif	34.25	35.94	31.91	14.2	16.0	10.2	9.27*	7.38*	28.48*	16.62	18.01	32.3	35.63*	43.39*	43.31*
	lbl	**37.17**	**44.18**	**31.91**	**20.94**	**30.97**	**10.24**	**18.7***	**23.88***	**28.48***	**22.07**	**30.0**	**32.3**	**44.78**	**56.67**	**28.71**

Appendix E Classification Results with Euclidean Distance

Table 16 contains the results obtained using the Euclidean distance as a distance metric.

Table 16: Results with Euclidean Distance

		SemEval			BioTech			Reuters			AAPD			LitCovid		
model	**thr**	**ma**F_1	**mi**F_1	P@1	**ma**F_1	**mi**F_1	P@1	**ma**F_1	**mi**F_1	P@1	**ma**F_1	**mi**F_1	P@1	**ma**F_1	**mi**F_1	P@1
GIST-Large	unif	43.32	49.96	68.18	14.65	17.10	20.73	**31.99**	33.04	50.55	33.76	40.11	64.3	38.38	41.56	1.59
	lbl	**45.29**	**54.16**	**68.18**	**17.00**	**22.41**	**20.73**	27.96	**37.91**	**50.55**	**40.01**	**46.73**	**64.3**	**41.31**	**51.26**	**1.59**
GTE-Large	unif	39.08	45.51	62.44	16.19	18.96	27.82	**33.24**	37.06	47.60	29.61	33.91	60.3	44.29	48.45	1.62
	lbl	**39.82**	**45.27**	**62.44**	**19.41**	**26.37**	**27.82**	33.00	**44.38**	**47.60**	**30.89**	**34.52**	**60.3**	**48.76**	**55.68**	**1.62**
Stella	unif	40.60	42.46	52.68	11.78	12.93	20.21	31.05	52.38	72.28	31.63	35.15	47.9	35.58	36.95	6.84
	lbl	**41.66**	**49.52**	**52.68**	**13.26**	**17.09**	**20.21**	**37.06**	**56.72**	**72.28**	**42.92**	**48.97**	**47.9**	**54.70**	**58.84**	**6.84**
UAE-Large	unif	41.49	46.50	62.29	12.28	14.34	26.51	**31.28**	31.04	36.95	29.81	30.60	49.1	35.87	39.79	3.41
	lbl	**42.01**	**49.67**	**62.29**	**14.05**	**19.11**	**26.51**	28.98	**32.98**	**36.95**	**36.46**	**41.86**	**49.1**	**41.39**	**48.57**	**3.41**
Mxbai	unif	41.58	46.11	61.34	14.09	16.06	25.20	**28.64**	29.98	39.03	30.82	31.25	52.6	36.59	41.16	3.05
	lbl	**42.33**	**50.45**	**61.34**	**16.80**	**22.20**	**25.20**	27.26	**31.58**	**39.03**	**37.02**	**42.81**	**52.6**	**42.44**	**51.74**	**3.05**
BGE	unif	40.32	43.76	57.29	12.46	14.54	17.85	**32.27**	36.14	51.04	27.71	33.49	49.6	32.54	35.52	4.20
	lbl	**41.87**	**50.55**	**57.29**	**14.91**	**19.63**	**17.85**	29.98	**43.39**	**51.04**	**33.89**	**42.03**	**49.6**	**37.57**	**48.23**	**4.20**
SF	unif	40.53	45.19	61.46	13.47	15.79	16.54	**31.94**	30.93	45.05	23.59	27.56	66.2	41.07	42.22	2.04
	lbl	**41.24**	**53.88**	**61.46**	**15.56**	**21.07**	**16.54**	31.42	**45.45**	**45.05**	**25.15**	**27.36**	**66.2**	**48.94**	**56.60**	**2.04**
all-MPNet	unif	37.92	43.51	59.71	13.94	16.33	17.85	25.50	29.59	49.72	34.47	43.24	62.1	41.01	43.86	2.37
	lbl	**38.37**	**49.57**	**59.71**	**21.42**	**22.87**	**17.85**	**27.82**	**37.54**	**49.72**	**37.20**	**42.10**	**62.1**	**53.53**	**60.10**	**2.37**
all-MPNet-Wiki	unif	35.64	37.74	47.01	15.98	17.77	21.26	26.01	38.33	56.30	34.36	39.34	57.2	36.00	40.01	3.40
	lbl	**37.11**	**44.22**	**47.01**	**12.16**	**26.58**	**21.26**	**26.48**	**48.28**	**56.30**	**36.38**	**46.51**	**57.2**	**42.14**	**53.73**	**3.40**
GloVe	unif	33.85	35.90	19.70	10.56	11.56	7.35	6.73	2.94	0.00	9.14	9.68	4.7	33.70	32.45	8.82
	lbl	**33.22**	**42.73**	**19.70**	**18.19**	**23.53**	**7.35**	**13.34**	**9.69**	**0.00**	**13.33**	**15.16**	**4.7**	**38.31**	**44.13**	**8.82**

References

[1] Adane Nega Tarekegn, Mohib Ullah, and Faouzi Alaya Cheikh. Deep learning for multi-label learning: A comprehensive survey, 2024.

[2] Nikmah Isnaini, Adiwijaya, Mohamad Syahrul Mubarok, and Muhammad Yuslan Abu Bakar. A multi-label classification on topics of indonesian news using k-nearest neighbor. *Journal of Physics: Conference Series*, 1192(1):012027, mar 2019.

[3] Emre Deniz, Hasan Erbay, and Mustafa Coşar. Multi-label classification of e-commerce customer reviews via machine learning. *Axioms*, 11(9), 2022.

[4] Jiawen Deng and Fuji Ren. Multi-label emotion detection via emotion-specified feature extraction and emotion correlation learning. *IEEE Transactions on Affective Computing*, 14(1):475–486, 2023.

[5] Shu Huang, Wei Peng, Jingxuan Li, and Dongwon Lee. Sentiment and topic analysis on social media: a multi-task multi-label classification approach. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, page 172–181, New York, NY, USA, 2013. Association for Computing Machinery.

[6] Jens Lemmens, Tess Dejaeghere, Tim Kreutz, Jens Van Nooten, Ilia Markov, and Walter Daelemans. Vaccinpraat: Monitoring vaccine skepticism in dutch twitter and facebook comments. *Computational Linguistics in the Netherlands Journal*, 11:173–188, Dec. 2021.

[7] Jens Van Nooten and Walter Daelemans. Improving Dutch vaccine hesitancy monitoring via multi-label data augmentation with GPT-3.5. In Jeremy Barnes, Orphée De Clercq, and Roman Klinger, editors, *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 251–270, Toronto, Canada, July 2023. Association for Computational Linguistics.

[8] Josefien Van Olmen, Jens Van Nooten, Hilde Philips, Annet Sollie, and Walter Daelemans. Predicting covid-19 symptoms from free text in medical records using artificial intelligence: Feasibility study. *JMIR Med Inform*, 10(4):e37771, Apr 2022.

[9] Changzhen Xiong and Yanmei Shan. Subject features and hash codes for multi-label image retrieval. In *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 808–812, 2018.

[10] Qian Wang, Ning Jia, and Toby P. Breckon. A baseline for multi-label image classification using an ensemble of deep convolutional neural networks. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 644–648, 2019.

[11] Rohit Gupta, Mamshad Nayeem Rizve, Jayakrishnan Unnikrishnan, Ashish Tawari, Son Tran, Mubarak Shah, Benjamin Yao, and Trishul Chilimbi. Open vocabulary multi-label video classification, 2024.

[12] Manjunath Mulimani and Annamaria Mesaros. Class-incremental learning for multi-label audio classification, 2024.

[13] A. F. Giraldo-Forero, J. A. Jaramillo-Garzón, and C. G. Castellanos-Domínguez. A comparison of multi-label techniques based on problem transformation for protein functional prediction. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2688–2691, 2013.

[14] Balázs Szalkai and Vince Grolmusz. Near perfect protein multi-label classification with deep neural networks. *Methods*, 132:50–56, 2018. Comparison and Visualization Methods for High-Dimensional Biological Data.

[15] Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: dataless classification. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 830–835. AAAI Press, 2008.

[16] Andriy Kosar, Guy De Pauw, and Walter Daelemans. Unsupervised text classification with neural word embeddings. *Computational Linguistics in the Netherlands Journal*, 12:165–181, Dec. 2022.

[17] Andriy Kosar, Guy De Pauw, and Walter Daelemans. Advancing topical text classification: A novel distance-based method with contextual embeddings. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 586–597, Varna, Bulgaria, September 2023. INCOMA Ltd., Shoumen, Bulgaria.

[18] Sappadla Prateek Veeranna, Jinseok Nam, Eneldo Loza Mencıa, and Johannes Fürnkranz. Using semantic similarity for multi-label zero-shot classification of text documents. In *Proceeding of european symposium on artificial neural networks, computational intelligence and machine learning. bruges, belgium: Elsevier*, pages 423–428, 2016.

[19] Nikolaos Mylonas, Stamatis Karlos, and Grigorios Tsoumakas. Zero-shot classification of biomedical articles with emerging mesh descriptors. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, page 175–184, New York, NY, USA, 2020. Association for Computing Machinery.

[20] Ghulam Mustafa, Muhammad Usman, Lisu Yu, Muhammad Tanvir Afzal, Muhammad Sulaiman, and Abdul Shahid. Multi-label classification of research articles using word2vec and identification of similarity threshold. *Scientific Reports*, 11(1):21900, 2021.

[21] Souvika Sarkar, Dongji Feng, and Shubhra Kanti Karmaker Santu. Zero-shot multi-label topic inference with sentence encoders and LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16218–16233, Singapore, December 2023. Association for Computational Linguistics.

[22] Souvika Sarkar, Dongji Feng, and Shubhra Kanti Karmaker Santu. Exploring universal sentence encoders for zero-shot text classification. In Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 135–147, Online only, November 2022. Association for Computational Linguistics.

[23] Min-Ling Zhang, Yu-Kun Li, Xu-Ying Liu, and Xin Geng. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12(2):191–202, 2018.

[24] Qingwu Fan and Changsheng Qiu. Hierarchical multi-label text classification method based on multi-level decoupling. In *2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE)*, pages 453–457, 2023.

[25] Ling Jia, Jin Fan, Dong Sun, Qingwei Gao, and Yixiang Lu. Research on multi-label classification problems based on neural networks and label correlation. In *2022 41st Chinese Control Conference (CCC)*, pages 7298–7302, 2022.

[26] Xunxin Cai, Meng Xiao, Zhiyuan Ning, and Yuanchun Zhou. Resolving the imbalance issue in hierarchical disciplinary topic inference via llm-based data augmentation. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1424–1429, 2023.

[27] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification, 2021.

[28] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2), apr 2022.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[30] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. In *arxiv*, 2018.

[31] Kai Yin, Chengkai Liu, Ali Mostafavi, and Xia Hu. Crisissense-llm: Instruction fine-tuned large language model for multi-label social media text classification in disaster informatics, 2024.

[32] Usman Malik, Simon Bernard, Alexandre Pauchet, Clément Chatelain, Romain Picot-Clémente, and Jérôme Cortinovis. Pseudo-labeling with large language models for multi-label emotion classification of french tweets. *IEEE Access*, 12:15902–15916, 2024.

[33] Jens Van Nooten and Andriy Kosar. Advancing CSR theme and topic classification: LLMs and training enhancement insights. In Chung-Chi Chen, Xiaomo Liu, Udo Hahn, Armineh Nourbakhsh, Zhiqiang Ma, Charese Smiley, Veronique Hoste, Sanjiv Ranjan Das, Manling Li, Mohammad Ghassemi, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen, editors, *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing*, pages 292–305, Torino, Italia, May 2024. Association for Computational Linguistics.

[34] Fengjun Wang, Moran Beladev, Ofri Kleinfeld, Elina Frayerman, Tal Shachar, Eran Fainman, Karen Lastmann Assaraf, Sarai Mizrachi, and Benjamin Wang. Text2Topic: Multi-label text classification system for efficient topic detection in user generated content with zero-shot capabilities. In Mingxuan Wang and Imed Zitouni, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 93–103, Singapore, December 2023. Association for Computational Linguistics.

[35] Jasmin Bogatinovski, Ljupčo Todorovski, Sašo Džeroski, and Dragi Kocev. Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, 203:117215, 2022.

[36] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[38] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017.

[39] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[40] Irina Radeva, Ivan Popchev, and Miroslava Dimitrova. Similarity thresholds in retrieval-augmented generation. In *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, pages 1–7, 2024.

[41] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. SemEval-2018 task 1: Affect in tweets. In Marianna Apidianaki, Saif M. Mohammad, Jonathan May, Ekaterina Shutova, Steven Bethard, and Marine Carpuat, editors, *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[42] Chidanand Apt'e, Fred Damerau, and Sholom M. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 1994. To appear.

[43] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. SGM: Sequence generation model for multi-label classification. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[44] Qingyu Chen, Alexis Allot, and Zhiyong Lu. LitCovid: an open database of COVID-19 literature. *Nucleic Acids Research*, 49(D1):D1534–D1540, 11 2020.

[45] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multi-label data. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 145–158, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[46] Aivin V. Solatorio. Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning. *arXiv preprint arXiv:2402.16829*, 2024.

[47] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

[48] Xianming Li and Jing Li. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023.

[49] Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. Open source strikes bread - new fluffy embeddings model, 2024.

[50] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[51] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, and Garrett Tanzer. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

[52] Kaitlyn Zhou, Kawin Ethayarajh, Dallas Card, and Dan Jurafsky. Problems with cosine as a measure of embedding similarity for high frequency words, 2022.

[53] Mohammadreza Heydarian, Thomas E. Doyle, and Reza Samavi. Mlcm: Multi-label confusion matrix. *IEEE Access*, 10:19083–19095, 2022.