Zero-Shot Multi-Label Topic Inference with Sentence Encoders & LLMs

Souvika Sarkar[1], Dongji Feng[2], Shubhra Kanti Karmaker Santu[1]
[1] Big Data Intelligence (BDI) Lab, Auburn University
[2] MCS department, Gustavus Adolphus College
szs0239@auburn.edu, djfeng@gustavus.edu, sks0086@auburn.edu

Abstract

In this paper, we conducted a comprehensive study with the latest Sentence Encoders and Large Language Models (LLMs) on the challenging task of "definition-wild zero-shot topic inference", where users define or provide the topics of interest in *real-time*. Through extensive experimentation on seven diverse data sets, we observed that LLMs, such as ChatGPT-3.5 and PaLM, demonstrated superior generality compared to other LLMs, e.g., BLOOM and GPT-NeoX. Furthermore, Sentence-BERT, a BERT-based classical sentence encoder, outperformed PaLM and achieved performance comparable to ChatGPT-3.5.

1 Introduction

Topic modeling and inference have been widely studied in the NLP literature (Alghamdi and Alfalqi, 2015; Jelodar et al., 2019; Chauhan and Shah, 2021). In this paper, we focus on *Zero-shot* approaches (Yin et al., 2019; Xie et al., 2016; Veeranna et al., 2016) for inferring topics from documents where both the document and topics were never seen by a model previously. For developing *Zero-shot* methods, we exclusively focus on leveraging the recent powerful Sentence Encoders and Large Language Models (LLMs) due to their recent success in a wide variety of NLP tasks.

The problem of *zero-shot topic inference* can be described using an intuitive example where the end user (possibly a domain expert) is actively involved in the inference process. Consider that the domain expert is analyzing a large volume of health articles and wants to automatically infer topics from those articles, including topics like "Autoimmune Disorders", "Heart health", "Arthritis", etc. For this real-life use case, the user will provide the collection of documents as well as a set of topics to be used as labels for categorizing the documents. Additionally, the user may also provide a list of relevant keywords/clues associated with each topic, which can be used as expert guidance for the inference process. The *zero-shot topic inference* algorithm then infers topics for each document.

Naturally, the *zero-shot topic inference* is a hard task, and only limited previous works studied this problem (Yin et al., 2019; Xie et al., 2016; Veeranna et al., 2016). However, with the recent developments in LLMs and pre-trained sentence embeddings like Cer et al. (2018b); Conneau et al. (2017a); Scao et al. (2022), we have observed significant performance boosts in many downstream zero-shot NLP tasks. Inspired by these, we decided to explore zero-shot methods by leveraging various sentence encoders [InferSent (Conneau et al., 2017a), Language-Agnostic SEntence Representations (LASER) (Artetxe and Schwenk, 2019), Sentence-BERT (SBERT) (Reimers and Gurevych, 2019a), and Universal Sentence Encoder (USE) (Cer et al., 2018b)] and recent LLMs [BLOOM (Scao et al., 2022), PaLM (Chowdhery et al., 2022), GPT-NeoX (Black et al., 2022), ChatGPT (Brown et al., 2020)] for topic inference. It is important to highlight that for all of our experiments with LLMs (except ChatGPT-3.5), we did not follow a prompting approach. Instead, to make an apple-to-apple comparison with classic sentence encoders, we generated sentence embeddings from these LLMs (except ChatGPT-3.5) and used the embeddings directly to infer topics.

In summary, we conducted extensive experiments with classic Sentence Encoders and LLMs on the *zero-shot topic inference* task and, consequently, established a comprehensive benchmark for future work in this direction. Experiment results with multiple real-world data sets, including online product reviews, news articles, and health-related blog articles, show that among all the models *ChatGPT-3.5* is superior in terms of generality compared to others, while *Sentence-BERT* performs exceptionally well and surpasses LLMs such as PaLM, BLOOM, and GPT-NeoX.

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 16218–16233
December 6-10, 2023

2 Related Work

This work is built upon prior research from multiple areas, including Topic Modeling and Categorization (Blei et al., 2003; Wang et al., 2011; Iwata et al., 2009), Text Annotation (Ogren, 2006; Zlabinger, 2019; Bijoy et al., 2021), Zero-Shot Learning (Veeranna et al., 2016; Yin et al., 2019), Sentence embeddings (Casanueva et al., 2020; Cer et al., 2018a), Large Language Models (Scao et al., 2022; Brown et al., 2020) etc. A brief discussion on each area and how this work is positioned concerning the state-of-the-art is as follows.

2.1 Topic Modeling and Inference

Classical Unsupervised Topic Models: Classical Topic Models, such as PLSA and LDA, emerged in the late 90s. PLSA was proposed by Hofmann et al. (Hofmann, 1999). LDA, introduced by Blei et al. (2003), extended PLSA by incorporating a generative model at the document level and remains widely used. Subsequently, several works, including Wang et al. (2011); Du et al. (2013); He et al. (2016); Hingmire and Chakraborti (2014), explored different aspects/issues of topic modeling.

Supervised Topic Inference: Studies such as Tuarob et al. (2015); Bundschus et al. (2009) have demonstrated the feasibility of supervised learning to categorize topics using well-annotated training data. Iwata et al. (2009) proposed a topic model for analyzing content-related categories in noisy annotated discrete data. Poursabzi-Sangdeh and Boyd-Graber (2015) combined document classification and topic modeling to uncover semantic structures. Engels et al. (2010) employed a latent topic model for the automatic categorization of videos with the associated text. In the field of neural text classification, researchers like Meng et al. (2018) addressed the challenge of limited training data. Additionally, Hassan et al. (2020) introduced a supervised classification for sexual violence report tracking.

Zero-Shot Topic Inference: Various topic modeling-based approaches have been explored for zero-shot classification for the English language (Karmaker Santu et al., 2016). Similarly, Li et al. (2018); Zha and Li (2019) worked towards a dataless text classification. Veeranna et al. (2016), adopted pre-trained word embedding for measuring semantic similarity between a label and documents. Further endeavor has been spent on zero-shot learning using semantic embedding by (Hascoet et al., 2019; Zhang et al., 2019; Xie and Virtanen, 2021; Rios and Kavuluru, 2018; Yin et al., 2019; Xia et al., 2018; Zhang et al., 2019; Pushp and Srivastava, 2017; Puri and Catanzaro, 2019; Yogatama et al., 2017; Pushp and Srivastava, 2017; Chen et al., 2021; Gong and Eldardiry, 2021).

2.2 Sentence Embedding

Powerful sentence encoders have demonstrated their effectiveness in various NLP tasks, such as Intent Classification Casanueva et al. (2020), Fake-News Detection Majumder and Das (2020), Duplicate Record Identification Lattar et al. (2020), Humor Detection Annamoradnejad (2020), Ad-Hoc monitoring Sarkar et al. (2023), and COVID-19 Trending Topics Detection Asgari-Chenaghlu et al. (2020). Researchers have explored dual-view approaches Cheng (2021), evaluated sentence embeddings for transfer-learning tasks (Perone et al., 2018; Enayet and Sukthankar, 2020), and examined the limitations of capturing sentence correctness and quality Rivas and Zimmermann (2019); Sarkar et al. (2022). Additionally, sentence embeddings have been utilized for domain-specific embeddings Chen et al. (2019), recommending research articles, and computing semantic similarity between articles (Hassan et al., 2019; Chen et al., 2018; Tang et al., 2018). Some studies have focused on understanding the encoded sentence representations Adi et al. (2017b) and investigating the impact of word frequency and distance on sentence encoding (Adi et al., 2017a).

2.3 Large Language Model (LLM) & Prompts

Recent research has extensively studied the potential of LLMs like ChatGPT, BLOOM, GPT, etc., for a wide range of applications. For example, researchers have shown the utility of ChatGPT in healthcare education (Sallam, 2023), programming bug solving (Surameery and Shakor, 2023), and machine translation (Jiao et al., 2023). Some works in the direction of prompt engineering direction are: White et al. (2023) presents a catalog of patterns to improve the outputs of LLM conversations, Reynolds and McDonell (2021) discuss methods of prompt programming, Jang et al. (2023) evaluated llm with negated prompts.

2.4 Difference from Previous Works

Despite the extensive research conducted in this field, there remains a noticeable gap in the systematic exploration of the potential of recent LLMs and sentence encoders for the target task. Specif-

ically, the utilization of LLMs for *zero-shot topic inference* has been largely unexplored, while existing studies on leveraging sentence encoders for text-topic similarity have primarily concentrated on a single encoder, limiting the scope of the investigation. In contrast, our work provides a comprehensive comparative analysis by evaluating multiple state-of-the-art sentence encoders as well as LLMs, considering various techniques to encode topics and documents. Additionally, we introduce novel approaches to incorporate user-provided auxiliary information for topic encoding, leading to improved inference results.

3 Problem Statement

The traditional *Topic Inference* task is defined as:

Definition 1 *Given a collection of documents D and a set of **pre-defined** topics T, infer one or more topics in T FOR each document $d \in D$.*

Thanks to the **pre-defined** set of topics T, the traditional *Topic Inference* task can benefit from fine-tuning based on a carefully designed training set for supervised learning. On the other hand, we follow the idea of *Definition-Wild Zero-Shot-Text Classification* coined by Yin et al. (2019), which is as follows:

Definition 2 *Definition-Wild 0SHOT-TC aims at learning a classifier f(·): X → Y, where classifier f(·) never sees Y-specific labeled data in its model development.*

Extending on top of *Definition-Wild (0SHOT-TC)*, we formalize our task from the user's standpoint in the following fashion:

Definition 3 *Given a collection of documents $D = \{d_1, d_2, ..., d_n\}$, a user x and a set of **user-defined** topics $T_x = \{t_1, t_2, ..., t_m\}$ provided in **real-time**, annotate each document $d_i \in D$ with zero or more topics from T_x **without any further fine-tuning**.*

In this dynamic setting, different users may provide varying sets of topics for the same dataset based on their specific application needs and goals. Customized training datasets in advance are no longer feasible as the target topics are provided in real time. We assume that each topic t is represented by a word or phrase, and users can include additional topic-related keywords K_t. Essentially, our ad-hoc problem assumes that the user, typically a domain expert in a specific field (e.g., a cardiologist or a business analyst), provides the documents, target topic, and optional keywords in real time.

Topics in a document may not be explicitly mentioned but rather implied through related keywords. For example, a document discussing "Mental Health" may not contain the exact phrase but may reference related terms like "Depression", "Anxiety", and "Antidepressant". These implicit topics are equally significant and should be annotated alongside explicit topics. While user-provided keywords help, it is challenging to create a comprehensive list capturing all possible descriptions. Also, the presence of a keyword does not guarantee the document's sole focus on that topic. Therefore, keywords alone cannot accurately infer topics; they merely serve as clues from the user.

4 Method for Zero-shot Topic Inference

In this section, we discuss the *zero-shot topic inference* approach we studied in this paper. The end-to-end inference process is shown in Fig 1.

1. The end user provides the inputs, i.e., article text, custom-defined topics, and optional keywords.
2. The article, topics, and keywords are individually inputted into the sentence encoder model, where we employ various *mid and large sentence encoders* (refer to Sec. 5.2).
3. Next, Two separate embedding vectors are generated by sentence encoders:
   - Article Embedding: The input article is encoded using three different approaches, which are further elaborated in section 4.1.
   - Topic Embedding: The candidate topics are embedded using four different approaches. The details are provided in section 4.2.
4. After obtaining the two embeddings, we compute their semantic similarity. We measure the similarity using cosine similarity between the embeddings. Subsequently, topics are assigned to the article based on the cosine similarity, employing a user-defined threshold. For a comprehensive analysis, we conducted experiments with various thresholds ranging from 0 to 1.
5. The output of the *zero-shot topic inference* framework is the set of the inferred topic(s).

4.1 Article Embedding

For article embedding, we adopted three methods as narrated in Table 1.

4.2 Topic Embedding

For generating topic embedding, we adopted four approaches, including and excluding the auxiliary information provided by the user, to do a comparative study. The details of topic embedding are

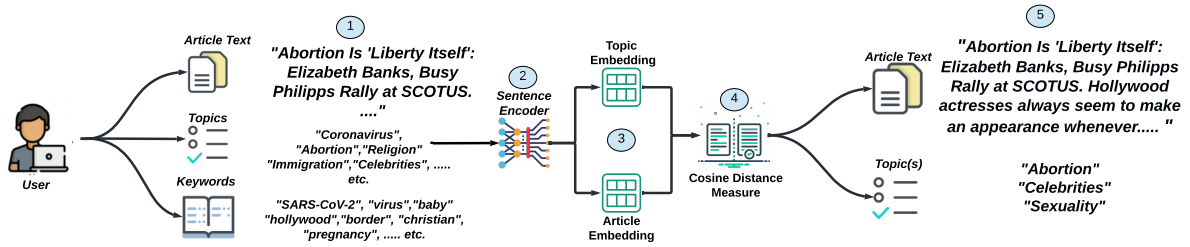

Figure 1: Steps for *Zero-shot Multi-Label Topic Inference* process, leveraging sentence encoders.

Embedding Approach	Description
Entire Article (EA)	Encode the entire article using Sentence Encoders at once, including articles that are long paragraphs and consist of more than one sentence.
Sentence Embedding Average (SEA)	Split the article into sentences, then encode each sentence, and at the end, average all sentence embedding to generate article embedding.
Individual Sentence Embedding (ISE)	Split the input article into sentences and encode each sentence separately. Then, unlike averaging (Sentence Embedding Average), use the individual sentence embeddings for similarity calculation with topic embedding.

Table 1: Three different ways of encoding an input article using sentence encoders.

Embedding Approach	Description
Topic-Name-Only	Encode only the topic name/phrase.
Topic + Keywords	Encode both topic name and keywords, then average all embeddings to generate the final topic embedding.
Topic + Keyword + Definition	Extract the topic's and keyword's definitions from WordNet, encode these definitions separately using sentence encoders, and then average all embeddings to generate the final topic embedding. For example, instead of encoding the keyword "campaign", we generated embedding of its definition, "a race between candidates for elective office".
Explicit-Mentions	First, extract all the articles explicitly mentioning the topic/phrase using algorithm 1 for all topics. Then, for each topic, generate embeddings of all articles that are explicitly annotated/labeled with that topic, then average them to obtain the ultimate topic embedding.

Table 2: Four different ways of encoding a topic using sentence encoders.

given in Table 2. As part of topic embedding using auxiliary information (Embedding approach "Explicit-Mentions"), we performed a rudimentary annotation on the dataset to find explicit mentions of the topics, which is discussed in algorithm 1.

Algorithm 1 Article Annotation using Explicit Mention

1: **Input:** Article text, Topic names and Keywords
2: **Output:** Articles labeled with explicit topics
3: **for** each article text **do**
4: check whether the topic name or set (at least 3) of the informative keywords are present or not in the corresponding article text
5: **if** present **then** label the article with the explicit topic
6: **end if**
7: **end for**

4.3 Zero-shot Topic Inference

Once we obtain all the embeddings, we measure cosine similarity between article and topic embeddings and, accordingly, infer topics. For instance, considering article a and topics $t \in T$ as well as considering "Entire Article" (EA) and "Topic-Name-Only" (TNO) as the embedding approach for article and topic, respectively, the inference of topic works as follows:

$$\hat{t} = \underset{t \in T}{\text{argmax}} \{\text{cosine\_similarity}\,(EA(a), TNO(t))\}$$

Where topic t belongs to a set of input topics T. $EA(a)$ represents the embedding of article a using the "Entire Article" embedding approach, and $TNO(t)$ represents the embedding of topic t using the "Topic-Name-Only" embedding approach. Note that, the combination of other articles and topic embeddings can be expressed in a similar way and hence, omitted due to lack of space.

5 Experimental Design

5.1 Datasets

Although the idea of our goal task is inspired from *Definition-Wild Zero-Shot-Text Classification* coined by Yin et al. (2019), we realized that the dataset introduced in the paper is suitable for Single-Label Multi-class classification, whereas our *zero-shot topic inference* setup is a Multi-label classification problem (more than one topic is as-

sociated with the input article). Hence, in our experiments, we mainly focused on curating/using the following datasets. (A) *Large datasets* with a higher number of articles for inference and relatively longer text (News and Medical-Blog collected from the web), and (B) *Small datasets* which contain fewer articles (<2000) for inference and are relatively shorter in length.

***Large Datasets*:** The large datasets were published by Sarkar and Karmaker (2022), which are collection of publicly available online news[1] and medical-blog articles[2]. Each article is already labeled with one or more ground-truth topics and stored in JSON objects. Some statistics about these datasets are summarised in Table 3.

***Small Datasets*:** The small datasets are originally a set of 5 different online product reviews; these were initially collected from Hu and Liu (2004) and re-annotated by Karmaker Santu et al. (2016). Unlike large datasets, the product reviews are shorter in length and contain more topics than the larger two datasets (see Table 3).

Dataset	# of Articles	Avg. Article length	Topics	Topics/ article
Medical	2066	693	18	1.128
News	8940	589	12	0.805
Cellular phone	587	16	23	1.058
Digital camera 1	642	18	24	1.069
Digital camera 2	380	17	20	1.039
DVD player	839	15	23	0.781
Mp3 player	1811	17	21	0.956

Table 3: Statistics on *Large* and *Small* datasets

Topic Name	Keywords
Addiction	Opioids, Alcohol, Drug
Headache	Migraine, Sinus, Chronic pain
Heart Health	Hypertension, Stroke, Cardiovascular
Mental Health	Depression, Anxiety, Antidepressant
Women's Health	Pregnancy, Breast, Birth

Table 4: Topics and keywords from the Medical dataset

In zero-shot learning, the auxiliary information about topics is provided by the end user (e.g., domain experts) conducting the inference task in the form of keywords/textual descriptions. In this section, we have shown some topics and corresponding keyword details from the Medical dataset (Table 4); due to lack of space, we provided more examples in the appendix A.1.

[1]https://newsbusters.org/

[2]https://www.health.harvard.edu/

5.2 Baseline and Sentence Encoder Models

As baselines, we used constrained topic modeling and a classical word embedding-based approach.

Generative Feature Language Models (GFLM) were proposed by (Karmaker Santu et al., 2016)). The paper suggested an approach based on generative feature language models that can mine the implicit topics effectively through unsupervised statistical learning. The parameters are optimized automatically using an Expectation-Maximization algorithm. Details on the method have been discussed in the appendix A.2.

Classical word embeddings are a popular way to encode text data into a dense real-valued vector representation. In order to implement a zero-shot classifier, we encoded both the input document and the target topics using pre-trained word embeddings and then computed vector similarity between the input document encoding and each target topic encoding separately. The implementation of the classical word-embedding-based zero-shot approach is very similar to our setup (discussed in section 4) with the following differences:

1. Instead of sentence encoders in step 2, pre-trained Glove embedding is used.
2. Articles are represented in two different ways.
   a) *Average Sentence Level Embedding:* For each input article, we encode the article by averaging the pre-trained embeddings (e.g., Glove) of each word present in that article.
   b) *Dictionary of Word Embeddings*: Extract word embedding of all words in an article, and instead of taking the average, we save them individually as a key-value pair.
3. For semantic similarity between Article and Topic embeddings, we used two metrics: 1) Euclidean distance and 2) Cosine Similarity.

The rest of the process, i.e., step 4 and step 5 are the same as discussed in section 4.

Sentence Encoders: We leveraged contemporary sentence encoders for the mentioned task. We refer to the traditional sentence encoders as *mid sentence encoders* (**MSE**), especially for their size, such as 1) InferSent (Conneau et al., 2017a), 2) Language-Agnostic SEntence Representations (LASER) (Artetxe and Schwenk, 2019), 3) Sentence-BERT (SBERT) (Reimers and Gurevych, 2019a), 4) Universal Sentence Encoder (USE) (Cer et al., 2018b). We utilized different large language models as sentence encoders, harnessing their embeddings, and referred to them as the *large sentence*

Prompt Design	
System setup	
The AI assistant has been designed to understand and categorize user input by the given topics. When processing user input, the assistant must predict the topics from one of the following pre-defined options: *"Addiction", "Alcohol", "Arthritis", "Brain and cognitive health", "Breast Cancer", "Cancer", "Children's Health", "Exercise and Fitness", "Headache", "Healthy Eating", "Heart Health", "Mental Health", "Osteoporosis", "Pain Management", "Prostate Knowledge", "Sleep", "Smoking cessation", "Women's Health"*. It is essential to note that an article may have multiple topics associated. If the user input is not relevant to any topics, the assistant should print nothing, indicating that the input does not align with the available categories. The agent MUST response with the following JSON format: {**"Topics": ["List of topics"]**}	
User	**Taking into account the given article:** "Perhaps as many as one in every 5 American adults will get a prescription for a painkiller this year, and many more will buy over-the-counter medicines without a prescription. These drugs can do wonders; getting rid of pain can seem like a miracle, but sometimes there's a high price to be paid. Remember the heavily marketed COX-2 inhibitors? Rofecoxib, sold as Vioxx, and valdecoxib, sold as Bextra, were taken off the market in 2004 and 2005, respectively, after studies linked them to an increased risk of heart attack and stroke. The nonsteroidal anti-inflammatory drugs (NSAIDs), like aspirin, ibuprofen (sold as Advil and Motrin), and naproxen (sold as Aleve) seem like safe bets.....", predict the category or topics of this article from the list of mentioned topics.
ChatGPT	"**Topics**": ["Arthritis", "Heart Health", "Pain Management"]
Directive: Taking into account the given article {article text}, predict the category or topic(s) of this article from the list of mentioned {topics}. Please remember to only respond in the predefined JSON format without any additional information.	

Table 5: Prompt design details for the *zero-shot topic inference* on Medical dataset.

encoder (**LSE**) due to their training on extensive text datasets. The specific models we employed were: 1) BLOOM (Scao et al., 2022), 2) GPT-NeoX (Black et al., 2022), and 3) PaLM (Chowdhery et al., 2022). We would like to mention that we did not perform fine-tuning or parameter tuning on top of the pre-trained sentence encoders and LLMs. We have provided brief descriptions of all the models in appendix A.3.

5.3 Sample ChatGPT Prompt

Additionally, we thoroughly examined the performance of ChatGPT-3.5 on the task. Due to the lack of access to the model's embeddings, we were unable to adopt the embedding-similarity-based classification approach as presented in Section 4. Instead, by utilizing the API, we adopted a prompting approach to perform the *zero-shot topic inference* task. Details of the ChatGPT prompt are presented in Table 5. For evaluation, we recorded the responses of ChatGPT given these prompts.

5.4 Evaluation Metric

To measure the performance of each *zero-shot topic inference* approach, we use three popular metrics available in the literature: Precision, Recall, and F_1 score. First, for each article, the model inferred topic(s) were compared against the list of "gold" topic(s) to compute the true positive, false positive, and false negative statistics for that article. Then, all such statistics for all the articles in a dataset were aggregated and used to compute the final Precision, Recall, and micro-averaged F_1 score.

Dataset	GFLM -S	GFLM -W	Classical Embedding (Glove)			
			Euclid. Word	Cosine Word	Euclid. Sent.	Cosine Sent.
Medical	0.532	0.530	0.212	0.154	0.105	0.142
News	0.494	0.492	0.141	0.171	0.113	0.115
Cellular phone	0.497	0.504	0.082	0.074	0.084	0.068
Digital cam. 1	0.460	0.471	0.120	0.118	0.142	0.127
Digital cam. 2	0.494	0.497	0.084	0.091	0.078	0.095
DVD player	0.473	0.486	0.096	0.100	0.096	0.108
Mp3 player	0.509	0.514	0.058	0.066	0.053	0.069

Table 6: F_1 score for Topic Modeling based baselines, GFLM-S, GFLM-W and Classical Embedding based baselines, Euclidean Word, Euclidean Sentence, Cosine Word, Cosine Sentence.

6 Performance Analysis and Findings

In this section, we present performance details of sentence encoders using various article and topic encoding techniques (refer to Table 1 and 2). The evaluation includes reporting the F_1 score for all sentence encoders. Table 6 contains baseline results for all datasets, while Table 7 shows performance for *Small datasets* using four topic embedding techniques and four *mid sentence encoders (MSE)*. For *Small datasets*, which mainly consist of single sentences, we considered "Entire Article" as the article embedding. Table 8 provides details on *Large datasets*, including twelve combinations of topic embedding techniques and three article em-

Topic Embedding	Sentence Encoder	Small Datasets				
		Cellular phone	Digital cam. 1	Digital cam. 2	DVD player	Mp3 player
Topic-Name-Only	InferSent	0.079	0.065	0.077	0.046	0.065
	LASER	0.091	0.087	0.101	0.076	0.097
	SBERT	0.418	0.427	0.520	0.295	0.373
	USE	**0.435**	**0.432**	**0.579**	**0.379**	**0.424**
Topic + Keywords	InferSent	0.077	0.063	0.080	0.045	0.055
	LASER	0.093	0.091	0.107	0.095	0.094
	SBERT	**0.549**	**0.503**	**0.554**	**0.478**	**0.433**
	USE	0.511	0.477	0.501	0.442	0.398
Topic + Keyword + Definition	InferSent	0.091	0.086	0.083	0.061	0.091
	LASER	0.192	0.212	0.100	0.247	0.165
	SBERT	0.220	**0.273**	**0.321**	**0.325**	0.277
	USE	**0.228**	0.266	0.236	0.261	**0.294**
Explicit-Mentions	InferSent	0.346	0.312	0.356	0.354	0.254
	LASER	0.293	0.337	0.370	0.323	0.280
	SBERT	**0.520**	**0.500**	**0.603**	**0.501**	**0.521**
	USE	0.488	0.457	0.593	0.449	0.486

Table 7: F_1 score for the *zero-shot topic inference* task for *Small datasets* (Cellular phone, Digital camera 1, Digital camera 2, DVD player, Mp3 player). Performance comparison of four *mid sentence encoders* over various topic embedding procedures for "Article Embedding" type.

beddings for *mid sentence encoders (MSE)*. Based on the performance of the *mid sentence encoders (MSE)*, we selected the best-performing article embedding ("Entire Article") and topic embeddings ("Topic + Keyword", "Explicit-Mentions") and continued our experiment with *large sentence encoders (LSE)*, as indicated in Table 9. Below, we summarize our findings.

1. Overall, ChatGPT (prompt-based) outruns all the encoders and baselines for all datasets except *Digital camera 2* dataset where SBERT attains the best result. Among others, Sentence-BERT (SBERT) performed close to ChatGPT. PaLM and USE performed somewhat mediocre; however, the remaining models performed poorly over both datasets and could not outrun the baseline as well. For qualitative analysis of the classified data, we picked a review from the Digital camera 1 (*Small*) dataset, which is associated with ground truth ***"Size"***, ***"Lens"***, ***"Photo"***. We observed that BLOOM and GPT-NeoX annotated the review with many incorrect topics, e.g. ***"Video"***, ***"Feature"***, ***"Manual"***, ***"Weight"***, ***"Focus"***, ***"Mode"*** etc. PaLM annotated the same review with correct and some other topics which are semantically correlated to the correct topics, for instances ***"Weight"*** (highly correlated with ***"Size"***), ***"Focus"***(highly correlated with ***"Lens"***), ***"Picture"***(highly correlated with ***"Photo"***). On the other hand, for the same review, ChatGPT inferred correct topics ***"Size"***, ***"Lens"***, ***"Photo"*** and an incorrect topic ***"Video"***, thus achieve best F_1 Score among all. Due to space limitation, we have added the case study with *Large* datasets in the appendix A.4.
2. Even though USE and PaLM could not beat ChatGPT and SBERT, they attained a score very close to the baseline methods (GFLM). Another intriguing observation is that PaLM's performance showed a significant improvement for the "Explicit-Mention" topic embedding compared to the "Topic + Keywords" topic embedding.
3. Based on our observations, the topic embedding techniques of "Topic+Keywords" and "Explicit-Mentions" exhibited superior performance compared to other methods. These embeddings, which incorporate user guidance through topic keywords, significantly improved the accuracy of real-time zero-shot topic inference. As a result, we employed these embeddings only when experimenting with *large sentence encoders (LSE)*.
4. The "Entire Article" approach excelled among other techniques for article embedding, making it the preferred choice when utilizing large sentence encoders (LSE). The "Sentence Embedding Average" method followed next in performance, while the "Individual Sentence Embedding" approach proved to be less promising.
5. "Explicit-Mentions" topic embedding with "Entire Article" as the article embedding attained the best score, followed by "Topic + Keywords" topic embedding paired with "Entire Article".
6. F_1 score obtained by InferSent, LASER,

Dataset ->		Medical											
Topic Embedding ->		Topic Name Only			Topic+Keywords			Topic+Keyword+Def'n			Explicit-Mentions		
Article Embedding ->		EA	SEA	ISE	EA	SEA	ISE	EA	SEA	ISE	EA	SEA	ISE
Sentence Encoder	InferSent	0.128	0.146	0.120	0.102	0.105	0.119	0.140	0.132	0.131	0.154	0.217	0.227
	LASER	0.120	0.142	0.134	0.124	0.122	0.121	0.125	0.124	0.185	0.187	0.139	0.136
	SBERT	**0.565**	**0.571**	**0.547**	**0.579**	**0.541**	**0.471**	**0.460**	**0.465**	**0.420**	**0.594**	**0.556**	**0.534**
	USE	0.488	0.516	0.429	0.500	0.484	0.340	0.390	0.409	0.375	0.520	0.504	0.468
Dataset ->		**News**											
Topic Embedding ->		**Topic Name Only**			**Topic+Keywords**			**Topic+Keyword+Def'n**			**Explicit-Mentions**		
Article Embedding ->		**EA**	**SEA**	**ISE**	**EA**	**SEA**	**ISE**	**EA**	**SEA**	**ISE**	**EA**	**SEA**	**ISE**
Sentence Encoder	InferSent	0.105	0.116	0.099	0.217	0.127	0.110	0.129	0.141	0.117	0.234	0.161	0.144
	LASER	0.171	0.180	0.154	0.181	0.176	0.135	0.126	0.127	0.128	0.130	0.136	0.134
	SBERT	**0.425**	0.408	**0.447**	**0.488**	**0.458**	**0.374**	0.406	0.386	0.378	**0.511**	**0.416**	**0.404**
	USE	0.419	**0.426**	0.367	0.461	0.418	0.281	**0.420**	**0.390**	**0.391**	0.446	0.371	0.368

Table 8: F_1 Score for the *zero-shot topic inference* task for *Large datasets* (Medical and News). Performance comparison of four *mid sentence encoders* over various topic embedding and article embedding techniques.

LSE ->	Bloom		GPTNeo		PaLM		ChatGPT
Topic Embedding ->	Topic+KWD	Expl.-Ment.	Topic+KWD	Expl.-Ment.	Topic+KWD	Expl.-Ment.	Prompt
Medical	0.259	0.308	0.259	0.268	0.295	0.392	**0.606**
News	0.301	0.329	0.286	0.274	0.387	0.410	**0.521**
Cellular Phone	0.258	0.268	0.215	0.269	0.296	0.565	**0.576**
Digital cam. 1	0.259	0.286	0.222	0.253	0.314	0.441	**0.641**
Digital cam. 2	0.224	0.260	0.194	0.273	0.363	0.486	**0.562**
DVD player	0.281	0.309	0.225	0.291	0.268	0.506	**0.533**
Mp3 player	0.246	0.284	0.216	0.241	0.304	0.479	**0.571**

Table 9: F_1 Score for the *zero-shot topic inference* task for *all datasets*. Performance comparison of three *large sentence encoders (LSE)* over various topic embedding procedures and ChatGPT prompt results.

BLOOM, and GPT-NeoX indicates that they failed to generalize over unseen datasets and, therefore, may not be a good choice for *zero-shot topic inference*.

7. Despite the observation stated in (6), we would like to point out that the inclusion of user guidance in the inference process boosted the performance of InferSent and LASER. For example, "Topic-Name Only" embedding achieved around 7% F_1 score (Average over all datasets); however, with "Explicit-Mentions" embedding, F_1 score reached 30% (average over all datasets).
8. For small datasets, "Topic-Name-Only" embedding presented an interesting case. Here, USE performed better than SBERT. This suggests that, for the product review domain, if additional keywords for each topic are unavailable, USE may be a better choice than SBERT. However, a detailed investigation is warranted to determine the root cause for this result.

Considering the real-time nature of our task, it is crucial to consider inference time when selecting the appropriate approach. In order to analyze computation time, we logged the duration taken by different encoders for article and topic embeddings. Due to space constraints, we have included the timing information for the best-performing article and topic embedding techniques in the main paper, while the timings for other embeddings such as "Sentence Embedding Average", "Individual Sentence Embedding", "Topic-Name-Only", and "Topic + Keyword + Definition" are provided in the appendix A.5. The generation time (in seconds) for each model is reported in Tables 11 and 10. Major observations from these tables are as follows.

1. USE is the fastest of all encoders for generating embeddings, followed by SBERT.
2. "Explicit Mentions" took more time for processing since, for "Explicit Mentions", the encoder needs to traverse the whole dataset.
3. The difference in article embedding time is more conspicuous on the *Large datasets* as they contain a longer and higher number of articles. USE, SBERT and PaLM-based embeddings clearly win over InferSent, LASER, BLOOM, and GPT-NeoX in terms of time as well.
4. The high processing time over *Large datasets* suggests that InferSent, LASER, BLOOM, and GPT-NeoX are unsuitable for real-time inference if the dataset is big or articles are long.

In essence, comprehensive performance and run-

Encoder Type	Topic+Keywords							Explicit-Mentions						
	MSE				LSE			MSE				LSE		
	Infer.	LASER	SBERT	USE	BLOOM	GPT-NeoX	PaLM	Infer.	LASER	SBERT	USE	BLOOM	GPT-NeoX	PaLM
Medical	1.964	1.024	1.720	2.183	22.213	93.557	14.177	730.655	708.560	19.453	21.154	2422.647	14013.319	128.255
News	1.673	0.801	1.479	0.809	19.121	55.454	12.034	2885.003	1438.609	61.226	36.005	9587.733	21247.567	390.947
Cellular phone	0.622	0.456	0.587	0.985	12.546	38.937	4.920	10.323	6.726	7.540	9.025	161.338	258.812	63.464
Digital cam. 1	0.621	0.490	0.416	0.914	10.854	41.174	3.317	13.002	9.707	9.597	8.865	177.990	317.446	71.092
Digital cam. 2	0.659	0.405	0.313	0.846	9.692	46.301	2.513	6.780	5.844	4.665	5.669	102.491	217.594	38.603
DVD player	0.682	0.422	0.373	0.833	10.380	49.443	2.814	9.695	10.822	6.802	7.271	146.187	234.948	60.409
Mp3 player	0.574	0.707	0.775	1.167	14.378	88.985	6.620	29.245	18.380	19.833	21.239	423.877	712.045	167.419

Table 10: Time comparison for generating topic embedding by different sentence encoders *(mid & large)* for *Small and Large data sets (Time unit in seconds).*

Total Time for Computing Embedding for Entire Article							
Sentence Encoder	**Medical**	**News**	**Cell phone**	**Digital cam. 1**	**Digital cam. 2**	**DVD player**	**Mp3 player**
InferSent	902.8	3867.3	8.6	7.8	4.6	10.6	21.4
LASER	514.9	1919.1	6.9	8.2	4.4	9.9	15.3
SBERT	28.8	88.6	6.7	6.9	4.4	9.6	18.7
USE	27.4	64.2	5.6	6.7	4.1	8.2	17.4
BLOOM	2341.4	9716.9	112.5	122.1	73.2	156.7	345.4
GPT-NeoX	16451.5	20741.6	170.6	293.2	118.9	236.2	559.3
PaLM	177.01	750.7	48.9	53.4	31.9	70.9	151.9

Table 11: Time comparison for generating article embedding by different sentence encoders (mid & large) for Small and Large datasets (Time unit in seconds).

time analysis show that a) auxiliary information helps in achieving better performance in real-time *zero-shot topic inference* task, b) even though the recent LLMs and sentence encoders are designed to be fairly general, aiming for seamless transfer learning, not all of them serve the purpose accurately, c) the processing time varies a lot across different sentence encoders and should be considered seriously while using these encoders in real-time tasks.

7 Conclusion

The task of *zero-shot topic inference* is both fundamental and challenging. Considering the challenge of *zero-shot topic inference* and the unexplored potential of recent sentence encoders and large language models (LLMs) in this area, we investigated their ability to generalize for this task alongside traditional sentence encoders.

In our real-time zero-shot topic inference task, we found varying performance among popular sentence encoders and LLMs. Among the *mid sentence encoders*, Sentence-BERT showed good performance on unseen data, while USE achieved decent accuracy. However, InferSent and LASER didn't perform at par with USE and SBERT. Among the *large sentence encoders*, ChatGPT performed the best, followed by PaLM. However, GPT-NeoX and BLOOM didn't generalize effectively for the task. We also introduced innovative approaches to incorporate user guidance, improving topic inference accuracy. Additionally, we conducted a thorough analysis of execution time, revealing that both the F_1 score and the efficiency of certain models (specifically BLOOM, GPT-NeoX, InferSent, and LASER) raise concerns when considering their suitability for the task.

8 Limitations

We acknowledge a limitation in our study 1) regarding the restricted access to various large language models (LLMs) such as LaMDA, Gato, and LLaMA (downloadable upon approval). Consequently, we were unable to fully utilize these models in our experiments. Additionally, due to limited availability, we had to rely on the API for evaluating the performance of models like ChatGPT. As a result, it is important to note that ChatGPT to the other three large language models is not a direct comparison. 2) Despite the promising results, we feel a limitation of our work is the reliance on keywords, i.e., the performance of the real-time zero-shot greatly depends on the choice of keywords; without appropriate keywords, the approach may suffer. In our future work, we will work towards mitigating this constraint. We believe that the ever-increasing scale of the data in different areas, new types of contents, topics, etc. will encourage the community to focus more towards *zero-shot topic inference* for categorization, and annotation and also motivate researchers to pursue research in this important direction.

9 Acknowledgements

This work has been partially supported by the National Science Foundation (NSF) Standard Grant Award #2302974 and Air Force Office of Scientific Research Grant/Cooperative Agreement Award #FA9550-23-1-0426. We would also like to thank Auburn University College of Engineering and the Department of CSSE for their continuous support through Student Fellowships and Faculty Startup Grants.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017a. Analysis of sentence embedding models using prediction tasks in natural language processing. *IBM Journal of Research and Development*, 61(4/5):3–1.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017b. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Rubayyi Alghamdi and Khalid Alfalqi. 2015. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 6(1).

Issa Annamoradnejad. 2020. Colbert: Using BERT sentence embedding for humor detection. *CoRR*, abs/2004.12765.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Meysam Asgari-Chenaghlu, Narjes Nikzad-Khasmakhi, and Shervin Minaee. 2020. Covid-transformer: Detecting COVID-19 trending topics on twitter using universal sentence encoder. *CoRR*, abs/2009.03947.

Biddut Sarker Bijoy, Syeda Jannatus Saba, Souvika Sarkar, Md Saiful Islam, Sheikh Rabiul Islam, Mohammad Ruhul Amin, and Shubhra Kanti Karmaker Santu. 2021. Covid19α: Interactive spatio-temporal visualization of covid-19 symptoms through tweet analysis. In *26th International Conference on Intelligent User Interfaces - Companion*, IUI '21 Companion, page 28–30, New York, NY, USA. Association for Computing Machinery.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

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

Markus Bundschus, Volker Tresp, and Hans-Peter Kriegel. 2009. Topic models for semantically annotated document collections. In *NIPS workshop: Applications for Topic Models: Text and Beyond*, pages 1–4.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. *CoRR*, abs/2003.04807.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018a. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 169–174. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018b. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Uttam Chauhan and Apurva Shah. 2021. Topic modeling using latent dirichlet allocation: A survey. *ACM Computing Surveys (CSUR)*, 54(7):1–35.

Qi Chen, Wei Wang, Kaizhu Huang, and Frans Coenen. 2021. Zero-shot text classification via knowledge graph embedding for social media data. *IEEE Internet of Things Journal*.

Qingyu Chen, Jingcheng Du, Sun Kim, W John Wilbur, and Zhiyong Lu. 2018. Combining rich features and deep learning for finding similar sentences in electronic medical records. *Proceedings of the BioCreative/OHNLP Challenge*, pages 5–8.

Qingyu Chen, Yifan Peng, and Zhiyong Lu. 2019. Biosentvec: creating sentence embeddings for biomedical texts. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–5. IEEE.

Xingyi Cheng. 2021. *Dual-View Distilled BERT for Sentence Embedding*, page 2151–2155. Association for Computing Machinery, New York, NY, USA.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017a. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 670–680. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017b. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Lan Du, Wray Buntine, and Mark Johnson. 2013. Topic segmentation with a structured topic model. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200.

Ayesha Enayet and Gita Sukthankar. 2020. A transfer learning approach for dialogue act classification of github issue comments. *arXiv preprint arXiv:2011.04867*.

Chris Engels, Koen Deschacht, Jan Hendrik Becker, Tinne Tuytelaars, Sien Moens, and Luc J Van Gool. 2010. Automatic annotation of unique locations from video and text. In *BMVC*, pages 1–11.

Jiaying Gong and Hoda Eldardiry. 2021. *Zero-Shot Relation Classification from Side Information*, page 576–585. Association for Computing Machinery, New York, NY, USA.

Tristan Hascoet, Yasuo Ariki, and Tetsuya Takiguchi. 2019. Semantic embeddings of generic objects for zero-shot learning. *EURASIP Journal on Image and Video Processing*, 2019(1):1–14.

Hebatallah A Mohamed Hassan, Giuseppe Sansonetti, Fabio Gasparetti, Alessandro Micarelli, and Joeran Beel. 2019. Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation? In *RecSys (Late-Breaking Results)*, pages 6–10.

Naeemul Hassan, Amrit Poudel, Jason Hale, Claire Hubacek, Khandaker Tasnim Huq, Shubhra Kanti Karmaker Santu, and Syed Ishtiaque Ahmed. 2020. Towards automated sexual violence report tracking. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 250–259.

Tieke He, Hongzhi Yin, Zhenyu Chen, Xiaofang Zhou, Shazia Sadiq, and Bin Luo. 2016. A spatial-temporal topic model for the semantic annotation of pois in lbsns. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1–24.

Swapnil Hingmire and Sutanu Chakraborti. 2014. Topic labeled text classification: A weakly supervised approach. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 385–394, New York, NY, USA. Association for Computing Machinery.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. 2009. Modeling social annotation data with content relevance using a topic model. In *Advances in Neural Information Processing Systems*, pages 835–843.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2023. Can large language models truly understand prompts? a case study with negated prompts. In *Transfer Learning for Natural Language Processing Workshop*, pages 52–62. PMLR.

Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. 2019. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*.

Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. 2016. Generative feature language models for mining implicit features from customer reviews. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 929–938.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28.

Hafsa Lattar, Aicha Ben Salem, and Henda Hajjami Ben Ghezala. 2020. Duplicate record detection approach based on sentence embeddings. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 269–274. IEEE.

Ximing Li, Changchun Li, Jinjin Chi, Jihong Ouyang, and Chenliang Li. 2018. Dataless text classification: A topic modeling approach with document manifold. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 973–982, New York, NY, USA. Association for Computing Machinery.

Soumayan Bandhu Majumder and Dipankar Das. 2020. Detecting fake news spreaders on twitter using universal sentence encoder. In *CLEF*.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 983–992, New York, NY, USA. Association for Computing Machinery.

Philip Ogren. 2006. Knowtator: a protégé plug-in for annotated corpus construction. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Demonstrations*, pages 273–275.

Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Forough Poursabzi-Sangdeh and Jordan Boyd-Graber. 2015. Speeding document annotation with topic models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–132.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. 2017. Train once, test anywhere: Zero-shot learning for text classification. *arXiv preprint arXiv:1712.05972*.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

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

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

Pablo Rivas and Marcus Zimmermann. 2019. Empirical study of sentence embeddings for english sentences quality assessment. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 331–336. IEEE.

Malik Sallam. 2023. Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, volume 11, page 887. MDPI.

Souvika Sarkar, Biddut Sarker Bijoy, Syeda Jannatus Saba, Dongji Feng, Yash Mahajan, Mohammad Ruhul Amin, Sheikh Rabiul Islam, and Shubhra Kanti Karmaker ("Santu"). 2023. Ad-hoc monitoring of covid-19 global research trends for well-informed policy making. *ACM Trans. Intell. Syst. Technol.*, 14(2).

Souvika Sarkar, Dongji Feng, and Shubhra Kanti Karmaker Santu. 2022. Exploring universal sentence encoders for zero-shot text classification. In *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. Association for Computational Linguistics.

Souvika Sarkar and Shubhra Kanti Santu Karmaker. 2022. Concept annotation from users perspective: A new challenge.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Nigar M Shafiq Surameery and Mohammed Y Shakor. 2023. Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, 3(01):17–22.

Xin Tang, Shanbo Cheng, Loc Do, Zhiyu Min, Feng Ji, Heng Yu, Ji Zhang, and Haiqin Chen. 2018. Improving multilingual semantic textual similarity with shared sentence encoder for low-resource languages. *arXiv preprint arXiv:1810.08740*.

Suppawong Tuarob, Line C Pouchard, Prasenjit Mitra, and C Lee Giles. 2015. A generalized topic modeling approach for automatic document annotation. *International Journal on Digital Libraries*, 16(2):111–128.

Sappadla Prateek Veeranna, Jinseok Nam, EL Mencía, and J Furnkranz. 2016. 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.

Hongning Wang, Duo Zhang, and ChengXiang Zhai. 2011. Structural topic model for latent topical structure analysis. In *ACL*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. Multinli: A corpus for multinatural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1112–1122.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3090–3099. Association for Computational Linguistics.

Huang Xie and Tuomas Virtanen. 2021. Zero-shot audio classification via semantic embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1233–1242.

Sihong Xie, Shaoxiong Wang, and Philip S. Yu. 2016. Active zero-shot learning. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 1889–1892, New York, NY, USA. Association for Computing Machinery.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3912–3921. Association for Computational Linguistics.

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*.

Daochen Zha and Chenliang Li. 2019. Multi-label dataless text classification with topic modeling. *Knowledge and Information Systems*, 61(1):137–160.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1031–1040. Association for Computational Linguistics.

Markus Zlabinger. 2019. Efficient and effective text-annotation through active learning. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1456, New York, NY, USA. Association for Computing Machinery.

A Appendix

A.1 Auxiliary Information Generation

Keyword list for each topic (based on the user's experience and expectations) was not readily available, for which we ideally needed a real user. To address this limitation, we followed the steps discussed in Sarkar and Karmaker (2022). We extracted the *informative* keywords for each topic using the TF-IDF heuristics. Then, a set of keywords for each topic were selected through careful inspection of the top keywords with high TF-IDF scores. In Tables 12 and 13, we have shown some topics and corresponding keywords details from News and Mp3 player dataset respectively.

Topic Name	Keywords
Economy	Recession, Budget, Stock Market
Global Warming	Climate, Planet, Green
Immigration	Border, Immigrants, Detention
Religion	Christian, Religious, Church
Sexuality	Gay, Lgbtq, Transgender

Table 12: Topics and optional keywords from News dataset

Topic Name	Keywords
Screen	Display, Screen saver, Interface
Sound	Audio, Headphone, Earbud
Navigation	Control, Scroll, Flywheel
Battery	Power, Recharge, mAh

Table 13: Topics and optional keywords from Mp3 player dataset

A.2 Details on Generative Feature Language Models

Generative Feature Language Models (GFLM) is unsupervised statistical learning in which parameters are optimized automatically using an Expectation-Maximization algorithm. Once the EM algorithm converges, one knows the topic distributions, i.e.

1. $P(z_{D,w} = t)$: Contribution of topic t for the generation of a particular word.

2. $P(z_{D,w} = B)$: Contribution of background model (mostly stop-words) for the generation of a particular word.

3. $\pi_{D,t}$: to what proportion, a particular document D is generated from some topic-of-interest t.

Based on these quantities, topic distributions within various documents can be inferred in two different ways, which were called **GFLM-Word** (GFLM-W) and **GFLM-Sentence** (GFLM-S).

GFLM-Word: It looks at each word w in the document D and adds a topic t to the inferred topic list if and only if $p(z_{D,w} = t) \times (1 - p(z_{D,w} = B))$ is greater than some threshold θ for at least one word in D. The philosophy behind this formula is that if any particular word w has a small probability of being generated by a background model but has a higher probability of being generated from some topic t, then word w is likely referring to topic t. Here, the decision is made solely by looking at individual words, not the entire document.

GFLM-Sentence: Given a document D, it looks at the contribution of each topic t in the generation of the sentence, i.e., $\pi_{D,t}$ and infers t^* as the topic only if π_{D,t^*} is greater than some user-defined threshold θ. Here, the decision is made at the sentence level, not at the word level.

A.3 Sentence Encoders & LLMs

This section presents a bird's-eye view of the sentence encoders we have used for our experiments.

InferSent (Conneau et al., 2017a) was released by Researchers at Facebook, which employs a supervised method to learn sentence embeddings. InferSent is trained on the Stanford Natural Language Inference (SNLI) corpus and generalizes well to many different tasks[3]. They found that models learned on NLI tasks can perform better than models trained in unsupervised conditions or on other supervised tasks (Conneau et al., 2017b). Furthermore, by exploring various architectures, they showed that a BiLSTM network with max-pooling outperformed the state-of-the-art sentence encoding methods, outperforming existing approaches like SkipThought vectors (Kiros et al., 2015). The model encodes text in 4,096 dimensional vectors.

[3]To use InferSent encoder we download the state-of-the-art fastText embedding and also download the pre-trained model from https://dl.fbaipublicfiles.com/senteval/infersent/infersent2.pkl.

Language-Agnostic Sentence Representations (LASER) (Artetxe and Schwenk, 2019), is a multilingual sentence embedding model that has been trained on over 93 languages. The training data consists of parallel text corpora, meaning texts that are translations of each other, from various sources including news articles, subtitles, and government documents. It was released by Facebook. LASER architecture is the same as neural machine translation: an encoder/decoder approach. It uses one shared encoder for all input languages and a shared decoder to generate the output language. The encoder is a five-layer bidirectional LSTM network. It does not use an attention mechanism but, has a 1,024-dimension vector to represent the input sentence. It is obtained by max-pooling over the last states of the BiLSTM, enabling comparison of sentence representations.

Sentence-BERT (SBERT) (Reimers and Gurevych, 2019b), is a modification of the pre-trained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. SBERT is a so-called twin network that allows it to process two sentences in the same way, simultaneously. BERT makes up the base of this model, to which a pooling layer has been appended. This pooling layer enables to create a fixed-size representation for input sentences of varying lengths. Since the purpose of creating these fixed-size sentence embeddings was to encode their semantics, the authors fine-tuned their network on Semantic Textual Similarity data. SBERT is trained on SNLI (Bowman et al., 2015) and the Multi-Genre NLI (Williams et al., 2018) dataset. The SNLI is a collection of 570,000 sentence pairs annotated with the labels contradiction, entailment, and neutral. MultiNLI contains 430,000 sentence pairs and covers a range of genres of spoken and written text. They combined the Stanford Natural Language Inference (SNLI) dataset with the Multi-Genre NLI (MG-NLI) dataset to create a collection of 1,000,000 sentence pairs. The training task posed by this dataset is to predict the label of each pair, which can be one of "contradiction", "entailment" or "neutral".

In 2018, Researchers at Google released a **Universal Sentence Encoder (USE)** (Cer et al., 2018b) model for sentence-level transfer learning that achieves consistent performance across multiple

NLP tasks. The models take as input English strings and produce as output a fixed dimensional (512) embedding representation of the string. Universal Sentence Encoder is trained on unsupervised training data for the sentence encoding models are drawn from a variety of web sources. The sources are Wikipedia, web news, web question-answer pages and discussion forums. Authors augment unsupervised learning with training on supervised data from the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015). The encoder is shared and trained across a range of unsupervised tasks along with supervised training on the SNLI corpus for tasks like a) Modified Skip-thoughtPermalink, b) Conversational Input-Response PredictionPermalink, c) Natural Language Inference. there are two architectures proposed for USE (Transformer Encoder and Deep Averaging Network). Based on shorter inference time observed in our experiments, we used Deep Averaging Network (DAN)[4] architecture.

BLOOM: Scao et al. (2022) introduce BLOOM, a massive language model with 176 billion parameters. BLOOM is trained on 46 natural languages and 13 programming languages and is the result of a collaborative effort involving hundreds of researchers. BLOOM is a causal language model trained to predict the next token in a sentence. This approach has been found effective in capturing reasoning abilities in large language models. BLOOM uses a Transformer architecture composed of an input embeddings layer, 70 Transformer blocks, and an output language-modeling layer. The sequential operation of predicting the next token involves passing the input tokens through each of the 70 BLOOM blocks. To prevent memory overflow, only one block is loaded into RAM at a time. The word embeddings and output language-modeling layer can be loaded on-demand from disk.

Pathways Language Model (PaLM): Chowdhery et al. (2022) introduce the Pathways Language Model (PaLM), a 540-billion parameter model. Large language models have been shown to achieve remarkable performance across a variety of natural language tasks using few-shot learning, which drastically reduces the number of task-specific training examples needed to adapt the model to a particular application. To further understanding of the impact of scale on few-shot learning, authors trained a 540-billion parameter, densely activated, Transformer language model, which they named *Pathways Language Model PaLM*. Developers trained PaLM on 6144 TPU v4 chips using Pathways, a new ML system which enables highly efficient training across multiple TPU Pods. The model demonstrate continued benefits of scaling by achieving state-of-the-art few-shot learning results on hundreds of language understanding and generation benchmarks. On a number of these tasks, PaLM 540B achieves breakthrough performance, outperforming the finetuned state-of-the-art on a suite of multi-step reasoning tasks, and outperforming average human performance on the recently released BIG-bench benchmark. A significant number of BIG-bench tasks showed discontinuous improvements from model scale, meaning that performance steeply increased as authors scaled to their largest model. PaLM also has strong capabilities in multilingual tasks and source code generation.

GPT-NeoX: The GPT-NeoX-20B paper, authored by the Black et al. (2022), introduce an architecture similar to GPT-3 but with notable differences. They utilize rotary positional embeddings for token position encoding instead of learned embeddings and parallelize the attention and feed-forward layers, resulting in a 15% increase in throughput. Unlike GPT-3, GPT-NeoX-20B exclusively employs dense layers. The authors trained GPT-NeoX-20B using EleutherAI's custom codebase (GPT-NeoX) based on Megatron and DeepSpeed, implemented in PyTorch. To address computational limitations, the authors reused the hyperparameters from the GPT-3 paper. In their evaluation, the researchers compared GPT-NeoX-20B's performance to their previous model, GPT-J-6B, as well as Meta's FairSeq 13B and different sizes of GPT-3 on various NLP benchmarks, including LAMBADA, WinoGrande, HendrycksTest, and MATH dataset. While improvements were desired for NLP tasks, GPT-NeoX-20B exhibited exceptional performance in science and math tasks.

ChatGPT: ChatGPT (Brown et al., 2020) is an advanced language model developed by OpenAI. It is designed to generate human-like text responses in a conversational manner. ChatGPT is built upon the GPT (Generative Pre-trained Transformer) architecture, which is a state-of-the-art deep learning model for natural language processing. ChatGPT is trained on a massive amount of text data from the internet to learn patterns, grammar, and context in language. It utilizes a transformer-based

[4]https://tfhub.dev/google/universal-sentence-encoder/4

neural network that consists of multiple layers of self-attention mechanisms and feed-forward neural networks. This architecture allows the model to understand and generate coherent and contextually relevant responses. The primary goal of ChatGPT is to provide natural and engaging interactions with users. It can be used in various applications, such as chatbots, virtual assistants, customer support systems, and more. By inputting a prompt or a message, users can receive a response generated by the model.

Total Time for Computing Article Embedding			
Article Embedding	**Sentence Encoder**	**Large Datasets**	
		Medical	**News**
Sentence Embedding Average	InferSent	1035.469	3807.204
	LASER	639.066	2373.594
	SBERT	548.573	1891.539
	USE	412.942	1448.037
Individual Sentence Embedding	InferSent	1022.728	3778.628
	LASER	631.533	2350.273
	SBERT	553.106	1876.776
	USE	428.725	1410.522

Table 14: Time comparison for generating article embedding by different sentence encoders for *Large datasets. (Time unit in seconds)*

A.4 Case study on *Large Datasets*

Due to space restriction we could not present case study from *Large dataset* in the main paper. Hence we have added our observation in this sections. Upon qualitative analysis of the classified data from Medical dataset, we observed while the input was an article which is originally labeled with topics ***"Heart Health"*** and ***"Mental Health"***; InferSent and LASER seemed to infer the correct topics along with many incorrect topics such as ***"Brain and cognitive health"***, ***"Healthy Eating"***, ***"Women's Health"***, ***"Children's Health"*** etc. The annotated dataset clearly indicates that InferSent and LASER are unable to distinguish between the topics in a zero-shot setting.

Universal Sentence Encoder (USE) annotated the same article as ***"Heart Health"***, ***"Mental Health"*** and ***"Brain and cognitive health"***. Correlation analysis reveals that ***"Mental Health"*** and ***"Brain and cognitive health"*** have high semantic correlation and therefore USE inferred both the topics.

Compared to all these sentence encoders, SentenceBERT performed precisely and inferred the correct topics for the article mentioned earlier.

The case study also corroborate our observations discussed in the paper that InferSent and LASER are unsuited for zero-shot approaches. While Universal Sentence Encoder performs moderate except for semantically correlated topics. SentenceBERT outrun all these encoders and effectively annotate datasets in the zero-shot approaches. Similar to *large sentence encoder*, we observed the same behavior in case of *mid sentence encoder* as well.

	Topic Name Only				**Topic + Keyword + Definition**			
Encoder	**Infer.**	**LASER**	**SBERT**	**USE**	**Infer.**	**LASER**	**SBERT**	**USE**
Medical	0.720	0.602	0.117	0.704	7.963	2.353	1.822	2.082
News	0.391	0.797	0.080	0.760	1.786	1.066	1.455	1.215
Cellular phone	0.350	0.147	0.118	0.706	2.707	0.790	0.570	1.043
Digital cam. 1	0.302	0.408	0.132	0.802	1.913	0.689	0.384	0.948
Digital cam. 2	0.418	0.073	0.109	0.721	1.354	0.534	0.535	1.159
DVD player	0.450	0.360	0.114	0.795	1.673	0.852	0.385	0.816
Mp3 player	0.466	0.190	0.122	0.762	2.654	0.963	0.729	1.217

Table 15: Time comparison for generating topic embedding by different sentence encoders (Time unit in seconds).

A.5 Time Comparison for Article and Topic Embedding

Since Small datasets are mainly consist of single sentences, only "Entire Article" as the article embedding is applicable on them. However, Large datasets are comprises of lengthy articles, hence different Article Embedding applied on them. The time for article embedding generation has been presented in Table 14. As we did not utilize the other two article embedding techniques in the experiment involving the *large sentence encoder*, Table 14 only includes the timing results for the *mid sentence encoder*.

In contrast, the timing results for topic embedding are presented in Table 15. Similar to article embedding, we did not utilize the other two topic embedding techniques in the experiment with the *large sentence encoder*. Therefore, Table 15 exclusively shows the timing results for the *mid sentence encoder* for all datasets.