Structural Contrastive Representation Learning for Zero-shot Multi-label Text Classification

Tianyi Zhang[1*], **Zhaozhuo Xu**[1*], **Tharun Medini**[2], **Anshumali Shrivastava**[1 2]
[1]Department of Computer Science, Rice University
[2]ThirdAI Corp
Houston, Texas, USA
tz21@rice.edu, zx22@rice.edu, tharun@thirdai.com, anshumali@rice.edu

Abstract

Zero-shot multi-label text classification (ZMTC) is a fundamental task in natural language processing with applications in the cold start problem of recommendation systems. Ideally, one would learn an expressive representation of both input text and label features so that ZMTC is transformed into a nearest neighbor search problem. However, the existing representation learning approaches for ZMTC struggle with accuracy as well as poor training efficiency. Firstly, the input text is structural, consisting of both short title sentences and long content paragraphs. It is challenging to model the correlation between short label descriptions and long structural input documents. Secondly, the enormous label space in ZMTC forces the existing approaches to perform multi-stage learning with label engineering. As a result, the training overhead is significant. In this paper, we address both problems by introducing an end-to-end structural contrastive representation learning approach. We propose a randomized text segmentation (RTS) technique to generate high-quality contrastive pairs. This RTS technique allows us to model title-content correlation. Additionally, we simplify the multi-stage ZMTC learning strategy by avoiding label engineering. Extensive experiments demonstrate that our approach leads to up to 2.33% improvement in precision@1 and $5.94\times$ speedup in training time on publicly available datasets. Our code is available publicly[†].

1 Introduction

Zero-shot multi-label text classification (ZMTC) (Chalkidis et al., 2020; Xiong et al., 2022; Song et al., 2021; Liu et al., 2021a; Lupart et al., 2022; Zhang et al., 2022) defines the following problem: given a set of documents with no labels and the full label description for each class, we would like to correctly classify unseen documents to these classes. ZMTC approaches can be leveraged to solve the cold start problem in e-commerce systems (Li et al., 2019; Chang et al., 2021). For instance, we can accurately retrieve newly added products with learning-based retrieval systems. With ZMTC, we do not have to worry about whether we have enough supervised data for the retrieval system. Without retraining the semantic matching model, ZMTC is capable of mapping between the customer query and its matched product descriptions even if they are recently added.

Challenges of ZMTC: We observe two major challenges in ZMTC. Learning the mapping between input text and its associated class descriptions is hard. In practice, both input text and class description can be a structural document with a short title sentence and a long descriptive paragraph (Bhatia et al., 2016). As a result, representation learning of multi-categorical input text and class description is likely hard. Secondly, the label space is enormous. Practitioners deploy ZMTC to tasks with number of classes in millions or even billions (Medini et al., 2019, 2020; Liu et al., 2021b; Dahiya et al., 2021). The explicit zero-short learning approaches that require learning softmax classifiers (Pourpanah et al., 2020) would become prohibitive due to the expensive overhead in computing billion-scale embeddings.

Exploiting Representation Learning in ZMTC: One ideal way of tackling ZMTC is to transform it into a nearest neighbor search problem. By learning meaningful representations for both input and class text, the ZMTC becomes a similarity search over embeddings. However, the existing representation approaches do not tackle the two major ZMTC challenges completely. The current ZMTC methods (Chalkidis et al.,

[*] Equal contribution.

[†] https://github.com/tonyzhang617/structural-contrastive-representation-learning

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4937–4947
December 7-11, 2022 ©2022 Association for Computational Linguistics

2020; Xiong et al., 2022) divide the structural document into a set of sentences. Next, they perform sentence-level representation learning by modeling the pairwise similarity of input and label documents. This representation learning method neglects the paragraph-level information and the structural relationship between the title and contents. Moreover, the current ZMTC approaches index the label space into clusters (Xiong et al., 2022), trees (Gupta et al., 2021) or graphs (Chen et al., 2021) to reduce the computation. However, this procedure results in multi-stage training, which is generally hard to optimize end-to-end for ZMTC training.

Our Proposal: This paper proposes an end-to-end structural contrastive representation learning approach for ZMTC. We propose a novel randomized text segmentation (RTS) method. We start by creating random chunks of the document contents into subsequences. Next, we pair the generated chunks with the document title as well as other chunks from the description to form positive pairs. The pairs are then used for contrastive representation learning. Our novel approach of combining titles and text introduces a data-dependent way that trains the model to associate segments of the description with the title. As a result, the relationship between the short title sentence and the long content paragraphs is baked into the representation. We can think of it as a novel self-supervised auxiliary task. This method allows us to learn the representation without label engineering. In other words, we transform ZMTC into learning the similarity between different types and modalities of text within documents. Specifically, our proposal enables us to represent the long paragraph with random subsequence sampling. Our extensive experiments indicate that our approach leads to up to 2.33% improvement in precision@1 and $5.94\times$ speedup in training time on state-of-the-art large-scale ZMTC benchmarks.

2 Related Work

2.1 Zero-Shot Multi-label Text Classification

Zero-shot multi-label text classification (ZMTC) is a standard natural language processing (NLP) task with practical significance. In recommendation systems, efficient ZMTC leads users to new products (Li et al., 2019; Chang et al., 2021). In medical document analysis, ZMTC is the tool for tagging medical subject headings to a stream of related papers (Lupart et al., 2022). Current ZMTC methods focus on label modeling by shrinking the large label space for more expressivity and better efficiency. For instance, Chalkidis et al. (2020) use a hierarchy of labels to help improve the ZMTC performance. Xiong et al. (2022) introduce a multi-scale label clustering to help the learning of both text and label representations. Liu et al. (2021a) introduce reasoning in the label hierarchy modeling to boost the effectiveness of per-trained language models in ZMTC. Zhang et al. (2022) introduce meta-data such as label synonyms in contrastive learning for better ZMTC. In this paper, we aim at a label-engineering-free approach of ZMTC. We focus our research on modeling the correlation between the title and contents of documents. As a result, we directly generate meaningful representations for both input text and labels so that ZMTC can be solved with efficient near neighbor search engines (Johnson et al., 2019).

2.2 Contrastive Learning

Inspired by the recent success of contrastive representation learning methods in the field of computer vision (Chen et al., 2020; Khosla et al., 2020; He et al., 2020), multiple contrastive learning approaches have been proposed for sentence representation learning in NLP. Wu et al. (2020) leverage multiple data augmentation techniques for better sentence representation learning. Zhang et al. (2020) attempt to maximize mutual information between sentence-level and token-level representations. Giorgi et al. (2021) sample spans of text as positive pairs for contrastive learning. Gao et al. (2021) use different dropout masks as data augmentation. Aside from sentence representation learning, document representation learning is also seeing contrastive learning approaches gaining traction. Xu et al. (2021) propose to represent documents as a graph attention network, in which each passage is a vertex, and perform contrastive learning on pairs of passage subsets to learn document representations. Luo et al. (2021) use data augmentation techniques such as synonym substitution and back-translation for better document representation.

Contrastive learning approaches have also been applied to ZMTC problems in prior works. Xu et al. (2022) propose to iteratively train the query encoder and document encoder using training pairs constructed with the Inverse Cloze Task (Lee et al., 2019) and dropout (Srivastava et al., 2014), and expand the set of negative instances with a cache queue. Xiong et al. (2022) construct positive pairs with the Inverse Cloze Task and augment the set of positive instances with pseudo-labels constructed with unsupervised clustering and TF-IDF. However, these contrastive learning approaches do not focus on modeling the structural information of both input and label documents. Moreover, the learning framework has multiple stages, making the training inefficient.

3 Methodology

In this section, we introduce our proposed structural contrastive learning approach for ZMTC. We start with our problem settings. Next, we introduce our approach of representation learning for structural text with title and content. Finally, we highlight the proposed randomized text segmentation with more intuition.

3.1 Problem Setting

Notations: In this paper, we denote $X = \{(t_1, c_1), \ldots, (t_{|X|}, c_{|X|})\}$ as a set of documents. Every $(t_i, c_i) \in X$ is a title-content pair where t_i and c_i represents title and content text, respectively. Let $Y = \{y_1, \ldots, y_{|Y|}\}$ be the set of labels. Each $y_i \in Y$ can be a short sentence description or a structural document with a title and contents. Each $(t_i, c_i) \in X$ corresponds to a subset of labels in Y. The set of documents X is split into disjoint subsets X_{train} and X_{test} for training and evaluation, respectively. We summarize the notations in Table 1.

The multi-label text classification problem is the problem of matching documents to their most relevant labels in a large pool of labels. ZMTC is an important subtask for this problem that focuses on unseen labels. In the ZMTC setup, we have access to X_{train} and Y for training a model to classify documents to labels. We would like to correctly classify each unseen document in X_{test} to labels in Y with the trained model. Due to the zero-shot nature of the problem, we do not have access to M, the ground truth mappings of documents to labels, for training. This problem formulation is general enough that many real-world problems can be modeled after, for example, predicting which items are similar to an item on an e-commerce website (Chang et al., 2021), predicting which categories an article belongs to on an online encyclopedia (Bhatia et al., 2016), or predicting medical subject headings for COVID-19 related articles (Lupart et al., 2022).

3.2 Learning Text Representation

In this section, we introduce our structural contrastive representation learning approach for ZMTC. We present an overview of our method in Figure 1. For document data containing both title and content, we start with randomized text segmentation to generate subsequences for better paragraph-level representation learning of long text. Next, we pair the generated text segments with titles or each other to construct positive pairs and train the model using a contrastive representation learning framework. As a result, we obtain representation for both input and label text so that ZMTC becomes a nearest neighbor search problem. In the following subsections, we start by introducing the randomized text segmentation technique. Next, we introduce our contrastive learning framework.

3.2.1 Randomized Text Segmentation

We perform randomized text segmentation (RTS) on the contents to divide a long text into non-overlapping contiguous subsequences. We use these subsequences to generate positive pairs for contrastive representation learning.

The contents c of a document is a finite sequence of terms, $c = (w_1, \ldots, w_{|c|})$, where each term w is a textual entity such as a word. We segment the contents into non-overlapping contiguous subsequences by sampling lengths $l_1, l_2, \ldots$ of the subsequences from the discrete uniform distribution $\mathcal{U}(L_{\text{min}}, L_{\text{max}})$, where L_{min} and L_{max} are hyperparameters. We keep sampling from the distribution until we obtain k sampled lengths from the distribution such that $\sum_{i=1}^{k} l_i \geq |c|$ and $\sum_{i=1}^{k-1} l_i < |c|$. Then, we segment the contents into k subsequences $(w_1,\ldots,w_{l_1}),(w_{l_1+1},\ldots,w_{l_1+l_2}),\ldots,(w_{1+\sum_{i<k} l_i},\ldots,w_{|c|})$. To prevent the last subsequence from being too short in length, we merge the last two subsequences by concatenation if the length of the last subsequence is less than $\frac{L_{\text{min}}}{2}$.

The process of randomized segmentation of contents is repeated independently every epoch. The subsequences obtained through segmenting the same text can be completely different for different epochs due to the independent sampling at each epoch.

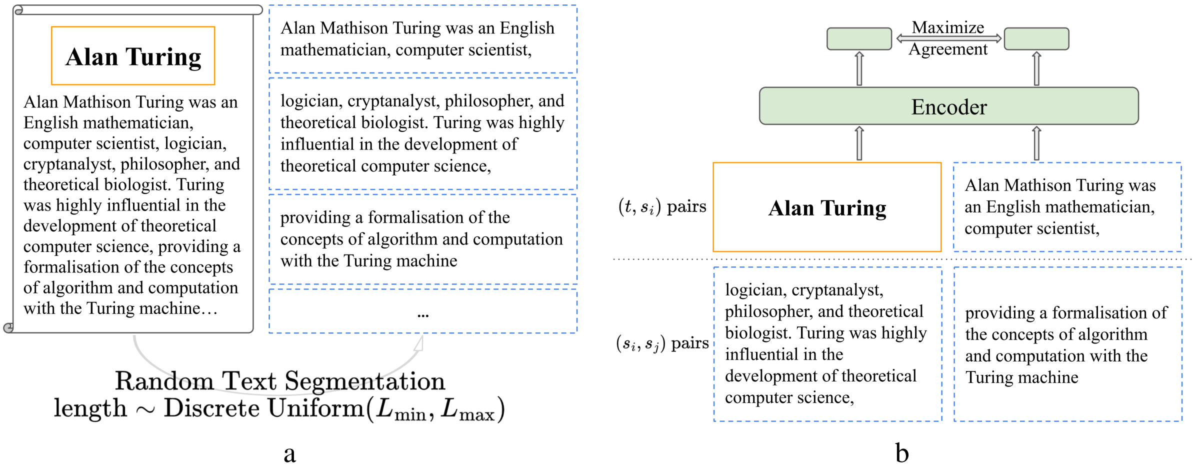
Alan Turing
Alan Mathison Turing was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine…
Alan Mathison Turing was an English mathematician, computer scientist,
logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science,
providing a formalisation of the concepts of algorithm and computation with the Turing machine
...
Random Text Segmentation
length ~ Discrete Uniform($L_{\min}, L_{\max}$)
a
Maximize Agreement
Encoder
(t, s_i) pairs
Alan Turing
Alan Mathison Turing was an English mathematician, computer scientist,
(s_i, s_j) pairs
logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science,
providing a formalisation of the concepts of algorithm and computation with the Turing machine
b

Figure 1: a) Randomized Text Segmentation breaks up the contents of a document into non-overlapping subsequences with lengths sampled from the discrete uniform distribution. b) Exploit the structure of document data by constructing (t, s_i) and (s_i, s_j) positive pairs for contrastive learning.

Notation	Definition
t	document title
c	document contents
y	label description
w	a textual entity such as a word
$s_1, s_2, \dots$	subsequences obtained from RTS
$l_1, l_2, \dots$	lengths of $s_1, s_2, \dots$
L_{min}	the minimum length of a RTS subsequence
L_{max}	the maximum length of a RTS subsequence
X_{train}	training set of documents
X_{test}	test set of documents
Y	label set
M	ground truth mapping from documents to labels

Table 1: Main notations used in this paper

3.2.2 Positive Pair Construction

In this section, we introduce how to construct positive pairs for contrastive representation learning given the RTS subsequences and the short title of the document. We construct two types of positive pairs for each input document and one or two types of positive pairs for each label document.

We perform RTS on the contents of every document. Given a document with content c and title t, we use RTS to obtain k subsequences $s_1, \dots, s_k$ of c. Next, we construct two types of positive pairs:

1. For each subsequence s_i, we pair it with t as (t, s_i). There would be k such pairs.
2. For each subsquence s_i, we pair it with another subsequence s_j where $i \neq j$. We form $\lceil \frac{k}{2} \rceil$ pairs of (s_i, s_j) by sampling pairs from $\{s_1, \dots, s_k\}$ without replacement, and pair the last remaining one with s_1 if k is odd.

For the label set Y, if it only contains a short description for each class, we directly construct $|Y|$ positive pairs of (y_i, y_i) and use dropout noise to prevent representation collapse (Gao et al., 2021). On the other hand, if elements in Y have both a short title and long contents, we apply the same pair construction method on labels as input documents. It is worth noting that we do not model the correlation between the input document and the labels in the zero-shot learning setup. We directly use the pairs for training in a contrastive learning framework with a language model as the encoder. This procedure is end-to-end learning with only one training stage. Moreover, it does not involve any label engineering such as clustering.

3.2.3 Training Loss

In this section, we introduce the contrastive loss we used for representation learning with positive pairs we constructed. Let $E(\cdot)$ denote a encoder with pre-trained weights. This encoder transforms input text into an embedding with fixed dimensions. We choose MPNet (Song et al., 2020) as the encoder, and use the `[CLS]` representation as the text embedding. Next, following the contrastive learning framework in (Chen et al., 2020), in each iteration, we sample a batch of positive pairs $\{(x_i, \hat{x}_i) | i \in \{1, \dots, b\}\}$ with size b and minimize the following loss:

$$\mathcal{L} = -\frac{1}{b} \sum_{i=1}^{b} \log \frac{e^{f(E(x_i), E(\hat{x}_i))/\tau}}{\sum_{j=1}^{b} e^{f(E(x_i), E(\hat{x}_j))/\tau}} \qquad (1)$$

where $f(x, \hat{x}) = \frac{x \cdot \hat{x}}{\|x\| \|\hat{x}\|}$ is the cosine similarity and τ is the temperature hyperparameter. We train $E(\cdot)$ for a certain number of epochs and update its weights to minimize the loss.

3.2.4 Inference

Once the training is finished, we perform inference with a nearest neighbor search framework. We first encode the labels into a set of embeddings $\{E(y)|y \in Y\}$. Next, given an input document with title t and contents c. We concatenate them as $t \bigoplus c$ and generate the document embedding $E(t \bigoplus c)$. Finally, we query and retrieve the k-nearest neighbor embeddings of $E(t \bigoplus c)$ in the set $E(Y)$. Here we use the same cosine similarity as our distance metric.

k-nearest neighbor search for dense embedding vectors can be greatly accelerated using the FAISS engine (Johnson et al., 2019). As a result, we obtain an efficient workflow for ZMTC.

3.3 Discussion

Motivation: The motivation of our method is to leverage the inherent structure in the data to generate high-quality pairs for contrastive learning. A document is a title-content pair, where the title is short and expresses the main topic of the document, and the contents are long and describe multiple concepts of the topic in detail. With randomized text segmentation, we break up the long contents into short segments, each of which consists of one or two constituent concepts of the topic. By pairing these segments with the title or other segments for contrastive learning, the model captures the semantic similarity between texts from different categories within the same document. Moreover, the model learns to produce high-quality representations for both input documents and labels. Furthermore, by independently repeating the RTS process every epoch, the model is trained on a different set of pairs every epoch. This prevents the model from memorizing the training pairs and overfitting, and encourages the model to capture the underlying semantic similarity of concepts within the document.

RTS vs Sentence-level Separation: Previous approaches of contrastive learning for document data break up documents into natural sentences by splitting text at appropriate punctuations (Xiong et al., 2022; Lee et al., 2019). However, this method has multiple downsides. Natural sentences are not ideal training data for contrastive learning, since they may be too short to capture enough context, and they are static. Moreover, the model is prone to memorizing the training data or overfitting, since it is trained on the same set of pairs every epoch. Our method produces training data of much better quality and variety, and enables the model to learn underlying patterns in the data that are otherwise difficult or impossible to recognize. We will demonstrate this empirically in 4.5.

Choices of Hyperparameter: Based on the motivation of our proposed method, we describe a method of setting hyperparameter values for L_{min} and L_{max}. We set $L_{\text{min}} = l$ and $L_{\text{max}} = 2l$ such that, with high probability, a subsequence of length l, which is randomly sampled from the contents of any document in the dataset, would capture enough context for one to recognize an idea or a concept described in the document.

4 Experiment

In this section, we evaluate the performance of our method and compare it with competitive baselines on 4 ZMTC datasets. There are 2 product recommendation datasets, 1 article recommendation dataset, and 1 article categorization dataset in our experiment. We choose these datasets for simulating the cold start problem in large-scale recommendation systems, information retrieval tasks in search engines, and natural language processing of unseen documents. In the experimental evaluation, we would like to answer the following questions: (1) Does our approach of RTS and pair construction improve the ZMTC accuracy? (2) Does our single-stage training improve the efficiency of ZMTC? (3) How do different sequence pairs affect the ZMTC performance?

Dataset	$\lvert X_{\text{train}}\rvert$	$\lvert X_{\text{test}}\rvert$	$\lvert Y\rvert$
LF-Amazon-131K	294,805	134,835	131,073
LF-WikiSeeAlso-320K	693,082	177,515	312,330
LF-Wikipedia-500K	1,813,391	783,743	501,070
LF-Amazon-1M	914,179	1,465,767	960,106

Table 2: Statistics of the datasets used for evaluation. $|X_{\text{train}}|$, $|X_{\text{test}}|$, $|Y|$ denote the number of training instances, the number of test instances, and the number of labels, respectively.

4.1 Datasets

We conduct our experiments on 4 publicly available datasets for multi-label text classification. Table 2 presents the statistics of the datasets. All 4 datasets have a very large set of labels, ranging from 131K to 960K in size, which enables us to accurately evaluate the model performance since real-world ZMTC tasks usually have an enormous label space. We obtain LF-Amazon-131K, LF-WikiSeeAlso-320K, and LF-Wikipedia-500K

Dataset	Learning Rate	L_{min}	L_{max}	Epochs	Batch Size	Training Pairs Used
LF-Amazon-131K	5×10^{-5}	40	80	10	384	$(t, s_i), (s_i, s_j), (y, y)$
LF-Amazon-1M	5×10^{-6}	40	80	10	384	$(t, s_i), (s_i, s_j), (y, y)$
LF-WikiSeeAlso-320K	5×10^{-8}	80	160	5	256	$(s_i, s_j), (y, y)$
LF-Wikipedia-500K	5×10^{-8}	80	160	5	256	$(s_i, s_j), (y, y)$

Table 3: Best hyperparameters and training settings for each dataset.

datasets from the extreme classification repository (Bhatia et al., 2016). The LF-Amazon-1M is available in (Gupta et al., 2021). All 4 datasets use data collected from real-world applications; LF-Amazon-131K and LF-Amazon-1M contain item-to-item recommendation data from the e-commerce website Amazon, LF-WikiSeeAlso-320K contains data for related articles from the encyclopedic website Wikipedia, and LF-Wikipedia-500K contains article categorization data from Wikipedia. Since the datasets use data from large-scale recommendation systems, they are ideal for evaluating the real-world performance of models.

4.2 Settings

4.2.1 Testbed

We implement our approach with PyTorch (Paszke et al., 2019). Our experiments are conducted on a machine with 4 NVIDIA Tesla V100 32GB GPU and 2 24-core/48-thread Intel Xeon Gold 5220R CPUs with 1.5TB of RAM.

4.2.2 Evaluation Metrics

We adopt precision at p or $P@p, p \in \{1, 3, 5\}$ and recall at r or $R@r, r \in \{1, 3, 5, 10, 100\}$ as the evaluation metrics for ZMTC tasks, which are defined as:

$$\begin{aligned} P@p &= \frac{\sum_{i=1}^{n} \sum_{y \in Y_i^{\text{pred}}} \mathbb{1}_i^{\text{top-}p}(y)}{np}, \\ R@r &= \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{y \in Y_i^{\text{pred}}} \mathbb{1}_i^{\text{top-}r}(y)}{\sum_{y \in Y} \mathbb{1}_i^{\text{top-}r}(y)}, \end{aligned} \tag{2}$$

where n is the number of documents evaluated, Y_i^{pred} is the set of predicted labels for the ith document, and $\mathbb{1}_i^{\text{top-}p}(\cdot)$ is an indicator function indicating whether a predicted label is a ground truth top-p label for the ith document. The precision and recall metrics are frequently used for this setup in prior works (Xiong et al., 2022; Reddi et al., 2019; Chang et al., 2021).

4.2.3 Hyperparameter

The best hyperparameters we found in our experiments for each dataset are shown in table 3. We adopt the same training procedure for each dataset. We finetune the base version of MPNet (Song et al., 2020) with positive pairs constructed with our proposed method for 5 or 10 epochs with the AdamW optimizer (Loshchilov and Hutter, 2019) with decreasing learning rate to optimize the loss function in Equation 1 with $\tau = 0.05$. The learning rate decays $10\times$ over epochs on a linear schedule.

We carry out grid search for the best learning rate in $\{5 \times 10^{-5}, 5 \times 10^{-6}, 5 \times 10^{-8}\}$ and $(L_{\text{min}}, L_{\text{min}}) \in \{(40, 80), (80, 160)\}$. We use fixed batch sizes that are large enough to take full advantage of GPU memory. We train the models for 5 or 10 epochs, depending on the size of the dataset.

Datasets that share the same source of data share almost identical hyperparameters; LF-WikiSeeAlso-320K and LF-Wikipedia-500K use identical hyperparameters, since their data are both sampled from Wikipedia. We avoid using (t, s_i) pairs for training on Wikipedia datasets, for the following reasons. For a Wikipedia article, the text in the short title is usually frequently repeated throughout the contents. Therefore, maximizing the agreement of the title with content subsequences is unnecessary and redundant. Additionally, training with (t, s_i) pairs will cause the encoder to focus solely on the title keywords in the content subsequences, instead of capturing the semantic similarity of text segments.

4.3 Baselines

We provide an overview of the baseline methods evaluated. All methods except XR-Linear encodes documents and labels into embedding vectors, and retrieves the labels with the most similar embeddings in terms of cosine similarity for a document. XR-Linear retrieves labels by querying a hierarchical tree structure.

- **MACLR**: (Xiong et al., 2022) A multi-stage contrastive learning method that uses clustering and TF-IDF to construct pseudo-labels.
- **TF-IDF**: (Ramos et al., 2003) represents input and label documents as sparse TF-IDF feature vectors.
- **GloVe**: (Pennington et al., 2014) represents input and label documents as Glove embeddings.

Dataset	Metric	Method								
		SBERT	GloVe	XR-Linear	SimCSE	TF-IDF	ICT	MPNet	MACLR	RTS
LF-Amazon-131K	P@1	1.86	3.67	7.56	10.13	12.38	13.82	13.94	18.13	**18.74**
	P@3	1.44	2.78	7.84	8.61	11.50	11.41	11.41	**15.42**	15.30
	P@5	1.14	2.15	7.30	6.69	9.14	8.90	8.82	11.93	**11.96**
	R@1	1.01	2.05	4.05	5.61	6.91	7.76	7.82	10.35	**10.64**
	R@3	2.22	4.33	12.11	13.39	18.14	18.09	18.08	**24.45**	24.16
	R@5	2.88	5.44	18.32	16.84	23.21	22.80	22.58	30.43	**30.45**
	R@10	4.01	7.23	29.17	21.27	29.32	28.94	27.91	37.28	**38.19**
	R@100	10.18	14.17	40.39	35.81	45.04	47.40	43.39	54.99	**59.34**
LF-WikiSeeAlso-320K	P@1	1.71	3.86	4.73	9.03	10.71	10.76	13.75	16.31	**18.64**
	P@3	1.27	2.76	4.27	6.64	8.90	10.05	11.93	13.53	**15.14**
	P@5	1.06	2.21	3.90	5.22	7.15	8.12	9.58	10.78	**12.07**
	R@1	1.08	2.12	2.23	4.99	5.92	6.12	8.14	9.71	**10.86**
	R@3	2.16	4.11	5.83	9.89	13.03	14.32	17.77	20.39	**22.68**
	R@5	2.90	5.22	8.64	12.34	16.48	18.05	22.21	25.37	**28.29**
	R@10	4.17	6.95	14.18	15.93	21.60	23.01	28.11	32.05	**35.47**
	R@100	10.76	15.33	36.93	30.11	42.55	39.77	45.91	53.83	**57.30**
LF-Wikipedia-500K	P@1	0.17	2.19	10.67	14.32	20.30	17.74	22.46	28.44	**30.67**
	P@3	0.15	1.52	8.77	6.84	12.98	9.67	12.87	17.75	**19.03**
	P@5	0.13	1.23	7.61	4.55	9.96	7.06	9.49	13.53	**14.34**
	R@1	0.05	0.85	3.69	4.24	7.25	7.35	8.74	10.40	**10.58**
	R@3	0.13	1.66	8.58	8.03	12.91	11.60	14.07	18.16	**18.48**
	R@5	0.18	2.18	12.11	11.26	15.98	13.84	16.76	22.38	**22.51**
	R@10	0.30	3.10	19.80	14.35	20.31	17.19	20.64	**28.52**	28.23
	R@100	1.29	8.52	31.02	27.68	38.16	31.08	34.72	**50.09**	48.00
LF-Amazon-1M	P@1	2.82	4.05	5.19	3.33	7.68	8.66	8.29	9.58	**10.00**
	P@3	2.87	4.07	5.48	3.69	9.20	9.26	8.87	10.41	**10.95**
	P@5	2.13	3.07	5.26	2.74	7.23	7.13	6.80	8.03	**8.41**
	R@1	2.03	2.91	3.63	2.38	5.61	6.30	6.04	**7.38**	7.34
	R@3	5.91	8.42	11.30	7.66	19.30	19.45	18.64	22.01	**23.09**
	R@5	7.21	10.44	17.94	9.38	24.92	24.60	23.51	27.72	**29.14**
	R@10	8.80	12.90	31.18	11.43	31.76	30.73	29.35	34.48	**36.30**
	R@100	14.22	21.18	43.79	18.54	51.79	48.42	46.15	55.23	**55.84**
Average of All	P@1	1.64	3.44	7.04	9.20	12.77	12.74	14.61	18.11	**19.51**
	P@3	1.43	2.78	6.59	6.45	10.64	10.10	11.27	14.28	**15.11**
	P@5	1.11	2.17	6.02	4.80	8.37	7.80	8.67	11.07	**11.70**
	R@1	1.04	1.98	3.40	4.31	6.42	6.88	7.69	9.46	**9.86**
	R@3	2.61	4.63	9.45	9.74	15.84	15.86	17.14	21.25	**22.10**
	R@5	3.29	5.82	14.25	12.46	20.15	19.82	21.27	26.47	**27.60**
	R@10	4.32	7.54	23.58	15.75	25.75	24.97	26.50	33.08	**34.55**
	R@100	9.11	14.80	38.03	28.03	44.38	41.67	42.54	53.53	**55.12**

Table 4: Precision and recall metrics of our method and other baselines on 4 datasets for ZMTC. For each metric, the best value is **bolded** and the second best value is underlined. RTS achieves the state-of-the-art results on most of the metrics, with substantial improvements to some metrics over previous best.

- **Sentence BERT (SBERT)**: (Reimers and Gurevych, 2019) a BERT model trained on extra data to specialize in producing high-quality sentence representations.
- **SimCSE**: (Gao et al., 2021) An unsupervised contrastive learning method that constructs positive pairs by pairing a sentence in the training corpus with itself and using dropout as data augmentation, and finetunes a BERT model with such pairs.
- **MPNet**: (Song et al., 2020) represent input and label documents with MPNet, a BERT model pre-trained with the masked and permuted training objective.
- **XR-Linear**: (Yu et al., 2022) A model that organizes labels into a hierarchical tree and constructs pseudo-labels with TF-IDF to overcome the lack of training supervision.
- **Inverse Cloze Task (ICT)**: (Lee et al., 2019) A BERT model trained with the ICT objective for title prediction.

4.4 Main Results

Accuracy: Table 4 shows the evaluation results on 4 ZMTC tasks. We report the precision and recall of the baselines from (Xiong et al., 2022). Our method attains the best results on most of the metrics, and substantially improves over previous state-of-the-art results on some. P@1 is improved from 16.31% to 18.64% on LF-WikiSeeAlso-320K and

Method	Ablation Settings					Precision			Recall				
	Model	Segmentation	(t, s_i)	(s_i, s_j)	(l, l)	@1	@3	@5	@1	@3	@5	@10	@100
Original RTS	MPNet	RTS (40, 80)	✓	✓	✓	**18.74**	**15.30**	**11.96**	**10.64**	**24.16**	**30.45**	**38.19**	**59.34**
Fixed segmentation	MPNet	Fixed (60)	✓	✓	✓	17.26	14.07	11.01	9.91	22.41	28.21	35.21	54.66
Natural segmentation	MPNet	Natural	✓	✓	✓	17.34	14.15	11.01	9.82	22.40	28.12	35.26	56.18
RTS with BERT	BERT	RTS (40, 80)	✓	✓	✓	18.57	15.01	11.75	10.57	23.75	29.94	37.56	58.57
No labels	MPNet	RTS (40, 80)	✓	✓	✗	18.37	14.86	11.53	10.45	23.53	29.41	37.00	57.82
(t, s_i) pairs only	MPNet	RTS (40, 80)	✓	✗	✗	17.50	14.61	11.35	10.03	23.30	29.10	35.92	55.03

Table 5: Experimental results of the ablation study. We study the impact of fixed and natural sentence segmentation, different pre-trained models, and different positive pairs.

Dataset	MACLR	RTS	Speedup
LF-Amazon-131K	18.08	6.06	2.98×
LF-WikiSeeAlso-320K	29.08	4.94	5.89×
LF-Wikipedia-500K	31.58	9.20	3.43×
LF-Amazon-1M	34.75	5.85	5.94×
Total	113.49	26.05	4.36×

Table 6: Training time (in hours) comparison between RTS and MACLR, the previous state-of-the-art method for ZMTC. RTS achieves significant training speedup, up to a factor of 5.94×, on all datasets.

from 28.44% to 30.67% on LF-Wikipedia-500K, and R@100 is improved from 54.99% to 59.34% on LF-Amazon-131K and from 53.83% to 57.30% on LF-WikiSeeAlso-320K. For the average metrics of all datasets, all precision and recall metrics are improved over previous state-of-the-art results, especially P@1 and R@100, which are improved by 1.4% and 1.59%, respectively. The results answer the first question, our approach can consistently improve the ZMTC performance on different tasks.

Efficiency: We compare the training time of RTS with the previous state-of-the-art method MACLR on ZMTC tasks. The training time statistics are shown in Table 6. We test both methods with the same hardware configuration. For MACLR, we use the code and the best hyperparameters provided in (Xiong et al., 2022). We the model with our method until the evaluation metric P@1 reaches the highest P@1 achieved by MACLR. The results answer the second question: our proposed ZMTC method achieves 2.98 × to 5.94× speedup in training.

4.5 Ablation Study

In this section, we answer the third question and perform an ablation study. We investigate the impact of segmentation methods, pretraining, and types of positive pairs on model accuracy. All ablation experiments are based on LF-Amazon-131K, and done with the same hyperparameters described in Section 4.2.3 to ensure a fair comparison. Detailed results of the ablation study are shown in Table 5.

Segmentation Methods: We study the impact of different text segmentation methods by comparing RTS, natural, and fixed segmentation. Natural segmentation breaks up long text into natural sentences, while fixed segmentation breaks it up into subsequences of fixed length. For fixed segmentation, we choose 60 as the length of each subsequence as it is the average length used in RTS. Natural and fixed segmentation methods perform similarly, while RTS outperforms both natural and fixed segmentation and achieves 1.4% − 1.48% better precision@1.

Pretraining: We compare using BERT (Devlin et al., 2019) and MPNet (Song et al., 2020) as the starting points for training to study the impact of pretraining. We compare the base version of BERT and MPNet, which have the same architecture and model size but different pretraining schemes. MPNet has been shown to outperform BERT on downstream tasks (Song et al., 2020). After the same amount of training time, BERT slightly underperforms MPNet, but it still achieves significantly better results than MPNet trained with naive segmentation methods. A better pretraining scheme produces slightly a better model for ZMTC, but it is not a significant contributing factor.

Positive Pairs: We remove each type of pair for training to investigate the impact of different types of pairs have on model accuracy. First, we remove label pairs (y, y) and train with only (t, s_i) and (s_i, s_j) pairs. The model retains good performance, so the label set is not necessary to produce high-quality models for ZMTC. Then we further remove (s_i, s_j) pairs and train with only (t, s_i) pairs. The resulting model still outperforms the ones trained on all 3 types of pairs with naive segmentation methods, since RTS exploits the structure of document data and enables the model to learn the underlying semantic similarity between segments.

5 Conclusion

In this paper, we proposed Randomized Text Segmentation (RTS) and positive pair construction strategies to exploit the structure within document data for end-to-end contrastive learning to advance state-of-the-art results on ZMTC tasks. Our proposed method achieves up to 2.33% improvement on precision@1 and up to 5.94× speedup in training time over previous state-of-the-art. We show that it is feasible to efficiently train high-quality models for challenging ZMTC tasks without having to resort to time-consuming, multi-stage methods with label engineering or methods that utilize inefficient softmax learning. Through extensive ablation experiments, we demonstrate the superiority of RTS over naive segmentation methods, and show that the types of positive pairs we proposed are indeed effective for learning better representation. We believe our work has a substantial impact as it can be applied to tackle many large-scale real-world problems such as cold-start recommendation problems, information retrieval, and medical document categorization and classification.

Limitations

A limitation of our approach is that it relies on complex pretrained transformer-based language models, such as BERT and MPNet, to achieve state-of-the-art results in ZMTC. Transformer-based models are computationally expensive, require specialized hardware such as GPU for training, and are difficult to deploy in large-scale productions. In the future, we would like to explore using simpler models such as embedding models for ZMTC tasks for more efficient training and inference.

Ethics Statement

We use GPUs to train transformer models, which have a notable carbon footprint. However, since our proposed approach improves training efficiency over previous methods by reducing multiple stages of training to one, we hope our work can help save energy in settings such as online recommendation systems.

References

K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code.

Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. An empirical study on large-scale multi-label text classification including few and zero-shot labels. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7503–7515.

Wei-Cheng Chang, Daniel Jiang, Hsiang-Fu Yu, Choon Hui Teo, Jiong Zhang, Kai Zhong, Kedarnath Kolluri, Qie Hu, Nikhil Shandilya, Vyacheslav Ievgrafov, et al. 2021. Extreme multi-label learning for semantic matching in product search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2643–2651.

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*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Kunal Dahiya, Deepak Saini, Anshul Mittal, Ankush Shaw, Kushal Dave, Akshay Soni, Himanshu Jain, Sumeet Agarwal, and Manik Varma. 2021. Deepxml: A deep extreme multi-label learning framework applied to short text documents. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 31–39.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

N. Gupta, S. Bohra, Y. Prabhu, S. Purohit, and M. Varma. 2021. Generalized zero-shot extreme multi-label learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4189–4196.

Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. 2021a. Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1062.

Yang Liu, Hua Cheng, Russell Klopfer, Matthew R Gormley, and Thomas Schaaf. 2021b. Effective convolutional attention network for multi-label clinical document classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5941–5953.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Dongsheng Luo, Wei Cheng, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Bo Zong, Yanchi Liu, Zhengzhang Chen, Dongjin Song, Haifeng Chen, et al. 2021. Unsupervised document embedding via contrastive augmentation. *arXiv preprint arXiv:2103.14542*.

Simon Lupart, Benoit Favre, Vassilina Nikoulina, and Salah Ait-Mokhtar. 2022. Zero-shot and few-shot classification of biomedical articles in context of the covid-19 pandemic. *arXiv preprint arXiv:2201.03017*.

Tharun Medini, Beidi Chen, and Anshumali Shrivastava. 2020. Solar: Sparse orthogonal learned and random embeddings. In *International Conference on Learning Representations*.

Tharun Kumar Reddy Medini, Qixuan Huang, Yiqiu Wang, Vijai Mohan, and Anshumali Shrivastava. 2019. Extreme classification in log memory using count-min sketch: A case study of amazon search with 50m products. *Advances in Neural Information Processing Systems*, 32.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, and Xi-Zhao Wang. 2020. A review of generalized zero-shot learning methods. *arXiv preprint arXiv:2011.08641*.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. New Jersey, USA.

Sashank J Reddi, Satyen Kale, Felix Yu, Daniel Holtmann-Rice, Jiecao Chen, and Sanjiv Kumar. 2019. Stochastic negative mining for learning with large output spaces. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1940–1949. PMLR.

Nils Reimers and Iryna Gurevych. 2019. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Congzheng Song, Shanghang Zhang, Najmeh Sadoughi, Pengtao Xie, and Eric Xing. 2021. Generalized zero-shot text classification for icd coding. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4018–4024.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Yuanhao Xiong, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit S. Dhillon. 2022. Extreme zero shot learning for extreme text classification. In *NAACL 2022*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. LaPraDoR: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3557–3569, Dublin, Ireland. Association for Computational Linguistics.

Peng Xu, Xinchi Chen, Xiaofei Ma, Zhiheng Huang, and Bing Xiang. 2021. Contrastive document representation learning with graph attention networks. In *Findings for EMNLP 2021*.

Hsiang-Fu Yu, Kai Zhong, Jiong Zhang, Wei-Cheng Chang, and Inderjit S Dhillon. 2022. Pecos: Prediction for enormous and correlated output spaces. *Journal of Machine Learning Research*, 23(98):1–32.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610, Online. Association for Computational Linguistics.

Yu Zhang, Zhihong Shen, Chieh-Han Wu, Boya Xie, Junheng Hao, Ye-Yi Wang, Kuansan Wang, and Jiawei Han. 2022. Metadata-induced contrastive learning for zero-shot multi-label text classification. In *Proceedings of the ACM Web Conference 2022*, pages 3162–3173.