Pragmatic Approach to Automatic Topic Labeling with Zero-shot Classification

Zikai ZHOU† and Kei WAKABAYASHI††

† Graduate School of Comprehensive Human Science, University of Tsukuba
1–2 Kasuga, Tsukuba-shi, Ibaraki 355–8550, Japan
†† Faculty of Library, Information and Media Science, University of Tsukuba
1–2 Kasuga, Tsukuba-shi, Ibaraki 355–8550, Japan
E-mail: †s1813024@klis.tsukuba.ac.jp, ††kwakaba@slis.tsukuba.ac.jp

Abstract In natural language processing, topic models are proposed to discern latent semantic structures within a text corpus. Once a topic model is applied, the extraction of topic terms becomes essential for interpreting the corpus's latent structure. The cognitive challenges of understanding a list of top terms lead to an interest in automatic topic labeling, which involves generating meaningful labels for topics and enhancing their interpretability. However, existing topic labeling methods often focus on individual topics, neglecting the overall corpus context and topic relationships. To address this, we propose a pragmatic approach that evaluates the ambiguity degree of topic labels through a listener model, which is instantiated by a zero-shot classification method. This model serves as a metric to assess the quality of label sets in the context of the entire corpus. Additionally, we introduce an automatic topic labeling pipeline employing combinatorial optimization to refine label sets for a given topic distribution. Experimental results demonstrate the correlation between our pragmatic approach and human evaluation, showcasing the effectiveness of our pipeline in automatic topic labeling. Our pragmatic approach emphasizes the importance of considering the holistic corpus context for more accurate and interpretable topic labeling.

Key words automatic topic labeling, pragmatic analysis

1 Introduction

In natural language processing (NLP), a topic typically refers to a specific subject or theme that is discussed in a context, which could be also considered as a latent semantic group of text. Topic models are proposed to find those latent semantic groups by analyzing a collection of documents, and automatically finding groups of words that frequently co-occur in the same document as topics. Understanding topics in text can be valuable for various applications, such as document summarization, sentiment analysis, information retrieval, and more. Traditional topic models such as Latent Semantic Analysis (PLSA) [1] and Latent Dirichlet Allocation (LDA) [2] are based on the Bayesian probabilistic model. Modern topic models such as Neural Variational Document Model (NVDM) [3], Bidirectional Adversarial Topic (BAT) model [4], and BERTopic [5] leverage modern neural networks and even large language models (LLM) to generate expressive topics.

In practice, topic terms are presented after a topic model is applied to the corpus. Topic terms are words or phrases defining and characterizing topics generated by topic modeling algorithms. Generally, these terms are representative of the underlying themes or subjects of topics and could help people interpret the topics and understand the latent structure of the corpus. For example, a topic related to computer science may have topic terms such as "computer", "programming", "algorithm", "code", etc.. In Latent Dirichlet Allocation, topic terms are the words with the highest probability of being associated with a particular topic. However, the cognitive load of understanding a whole list of (in practice, top 10 or top 20) terms is considered high compared to that of understanding an expressive label. Previous work [6] showed that in the task of information retrieval, humans can identify more documents when textual labels are presented than when keywords are presented. This nature has led to interest in automatic topic labeling, which is the task of outputting meaningful words or phrases as labels for topics.

A topic label is a short text that precisely describes the semantics of what the topic represents. Topic labels are expected to be comprehensible for humans to understand the topics. In definition, the topic labels can be just a concatenation of top words (e.g., "computer, programming, algorithm, code"), while a more concise and precise term that represents

the topic (e.g., , "Computer Science") is preferred.

There have been many automatic topic labeling researches. The main task of automatic topic labeling is associated with the task of finding proper label candidates and ranking them. Mei et al. [7] first introduced automatic topic labeling to LDA topics by extracting bigrams and ranking them with KL-divergence. Many automatic topic labeling methods follow this idea, such as Lau et al. [8] using the title of English Wikipedia pages as label candidates and then applying the ranking method with lexical association feature to the corpus. On the other hand, with the advancement of computing power and the development of language models based on transformers, direct approaches to label generation have been studied. Alokaili et al. [9] proposed a sequence-to-sequence model that takes topic terms as input and output corresponding labels, and Popa et al. [10] fine-tuned a pre-trained BART [11] model to generate labels from topics.

However, the ambiguity of topics generated by topic models has caused trouble for automatic topic labeling. For example, consider two different topics {*Twitter, ChatGPT, 5G, Apple watch*} and {*Northern lights, SpaceX, NASA, Gravitational waves*}. While it is reasonable to label them as "Technology" and "Science" respectively, a more distinctive label set would be "Information Technology" and "Space and Astronomy". Automatic topic labeling methods often focus solely on a single topic while ignoring the integrity of the whole corpus and topic relationship, which could bring problems to humans interpreting the topics. This problem drives us to handle topic labeling in a pragmatic manner, that labels should be generated in the context of the entire corpus, and could bring the least misconception among topics. To achieve this goal, we propose a pragmatic automatic topic labeling method by leveraging zero-shot classification of text as the evaluation of the distinctness degree of the topic labels. The overview of our proposal is shown in Figure 1. The listener model could be used to compute a metric for evaluating the distinctness of the topic label sets for a given topic distribution. Furthermore, we construct an automatic topic labeling pipeline that takes label candidates as input and applies combinatorial optimization to find the proper topic label set to the corpus.

The contributions of this paper are summarized as follows:

(1) We propose a pragmatic listener model by leveraging zero-shot classification of text as the evaluation of the distinctness degree of the topic labels.

(2) We validate the distinctness of label sets generated by existing automatic topic labeling methods by conducting crowd-sourcing tasks on Amazon Mechanical Turk. The evaluation of our pragmatic listener model on these label sets shows a correlation with the crowd-sourcing results.

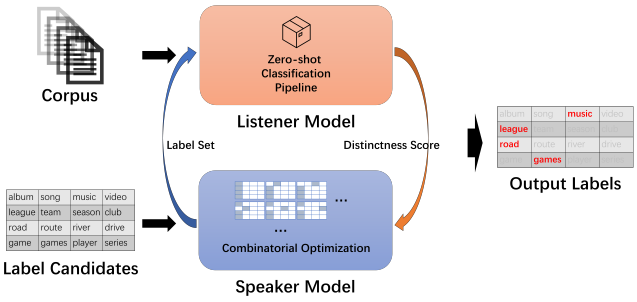

Figure 1 Overview: a pragmatic listener-speaker architecture leveraging combinatorial optimization and zero-shot classification of text for automatic topic labeling.

(3) We propose a pragmatic automatic topic labeling pipeline in the listener-speaker architecture. The experiments show that our pipeline has improved the distinctness of the topic label set.

2 Related Work

2.1 Topic Modeling

Latent Dirichlet Allocation (LDA) [2] is a probabilistic topic model that assumes documents are mixtures of topics and topics are mixtures of words. It adds a prior distribution to the document-topic and topic-word distributions on the previous work of Latent Semantic Analysis (PLSA). By applying the idea of topic distributions, other probabilistic topic models have been proposed, such as the Correlated Topic Model (CTM) [12] and the Structured Topic Model (STM) [13]. Some other approaches, such as NVDM [3] and DocNADE [14], use modern neural networks to generate similar distributions. In recent years, neural topic models have been cooperating with other methods such as graph neural networks [15], reinforcement learning [16], contrastive learning [17], and generative adversarial networks [18] to improve their performance.

Aside from probabilistic topic models, some other approaches interpret topics as clusters of documents. Top2Vec [19] leverages joint document and word semantic embeddings to find topic vectors. Top2Vec employs a Doc2Vec [20] model for embeddings and applies a clustering algorithm to the document embeddings to generate topic vectors. Sia et al. [21] introduced a similar framework that leverages pre-trained embeddings. This framework provides flexibility in the model choice on embedding models, dimension reduction methods, and topic term extraction. BERTopic [5] leverages embeddings provided by BERT [22], a pre-trained transformer-based language model, while using a class-based variation of the TF-IDF to generate coherent topic representations instead of cluster centroids.

Our proposed automatic topic labeling model has the ben-

efit of not restricting a specific type of topic model, since the labeling process is separate from the topic modeling process and does not require any special model structure.

2.2 Automatic Topic Labeling

Automatic topic labeling has experienced significant progress over the years, evolving from unsupervised to supervised approaches and incorporating the latest advancements in natural language processing (NLP). Pioneering work by Mei et al. [7] established a foundation by introducing methods to extract bigrams and rank them using KL-divergence, framing the practice of label generation from within the original corpus itself. Subsequent research such as Lau et al. [8] and Bhatia et al. [23] expanded upon this idea by employing external corpora like Wikipedia and leveraging supervised learning techniques to rank candidates.

Some other approaches leverage automatic summarization technology for labeling topic models, aiming to enhance the informativeness and meaningfulness of topic labels. Unlike other approaches, these methods mainly produce topic labels in sentence form and focus on more informative expressions of topics rather than common phrase-level labels. Cano et al. [24] introduced an independent summarization framework for identifying dominant words, which only relies on the identification of dominant terms in documents related to the latent topic. Following the idea, other approaches [25] [26] [27] [28] leverage different summarization and ranking methods and achieve better results.

As the field advanced, word embedding techniques such as Word2Vec [29] and Doc2Vec [20] were employed, enhancing the process of computing semantic similarity between topics and potential labels. The introduction of transformers [30] marked a paradigm shift in the field. The encoder-decoder architecture and the pre-training on vast corpora of text brought about a generation of transformer-based models [22] [31] [11] that excel in transfer learning, enabling unprecedented performance on a wide array of NLP tasks, including topic labeling.

Most recently, direct labeling techniques exploiting the power of transformer models have been explored. Alokaili et al. [9] introduced a sequence-to-sequence model for generating labels based on topic terms. Popa et al. [10] demonstrated the effectiveness of fine-tuning a pre-trained BART [11] model for extracting labels directly from topics. These approaches mark a shift towards leveraging pre-existing language models, which have been trained on general language understanding, to tailor them to the specific task of automatic topic labeling. Through fine-tuning these models on targeted datasets, research has enabled more accurate and contextually appropriate labels, even with smaller and domain-specific data, pushing the boundaries of what's possible in automatic topic labeling.

Among all the research mentioned above, no research focuses on inter-topic relationships when assigning labels to topics. When similar topics are presented to an existing automatic topic labeling method, one may output ambiguous labels, which could bring problems to humans interpreting the topics. Our proposal on the other hand takes the entire topic distribution as input and focuses on assigning pragmatic labels to topics in the holistic corpus context.

2.3 Pragmatic Analysis

A fundamental concept in computational pragmatics is that speaker and listener agents function within a cooperative framework, while both gain advantages by engaging in reasoning about the intentions and actions of others within this context. William et al. [32] firstly used pragmatic reasoning alone with weighted inference to address ambiguity and generate clarification requests in a human-robot dialog task. Daniel et al. [33] introduced a pragmatic listener-speaker model to reason why speakers produce certain instructions, and how listeners will react to them, and showed that such pragmatic inference improves both the listener and speaker models in diverse settings. Further work [34] extended the idea of pragmatics in other NLP systems, examining how task goals, environmental contexts, and communicative affordances in each study enhance linguistic meaning, and offering suggestions for future grounded task design aimed at organically eliciting pragmatic phenomena.

Our proposal involves the idea of pragmatic analysis by constructing a listener-speaker architecture that works cooperatively to generate distinctive topic labels for each topic.

3 Proposal

3.1 Speaker Model

Consider a topic distribution of $T = \{t_1, t_2, \ldots, t_m\}$ where m is the number of topics. For i-th topic t_i, the label candidates of the topic is $L_i = \{l_{i,1}, l_{i,2}, \ldots, l_{i,n_i}\}$ where n_i is the number of label candidates of topic t_i. The input of the speaker model will be $S = (L_1, L_2, \ldots, L_m)$, which is the set of all label candidates. The set of all possible combinations could be presented as $\Phi = L_1 \times L_2 \times \cdots \times L_m$.

A possible output label assignment $\phi \in \Phi$ for topic distribution T will be $\phi = (l_1, l_2, \ldots, l_i)$ where $l_i \in L_i$. The total number of possible label sets could be calculated as

$$|\Phi| = \prod_{i=1}^{m} n_i, \tag{1}$$

which will be a very large number when m and n_i grow. The speaker model should address the issue of exploring the large combinatorial space in terms of computational efficiency. In Section 3.3, an efficiency-aware method based on a combi-

natorial optimization technique is presented.

3.2 Listener Model

The listener model will evaluate each ϕ with a distinctness score function $f(D, \phi)$ by classifying the corpus D on ϕ. For document $d \in D$, the gold label $y_{true}(d)$ is the index of the topic assigned to the document by the topic model. The classification result $y_{pred}(d, \phi)$ will be granted by passing d and ϕ to a zero-shot text classification model, which is the corresponding index of the topic of the classification result label l_{pred}. The classification metric would consider the prediction correct if and only if $y_{true}(d) = y_{pred}(d, \phi)$.

The distinctness score could then be calculated as:

$$f(D, \phi) = \frac{1}{|D|} \sum_{i=1}^{m} n(D, i) F1(D, \phi, i), \tag{2}$$

$$n(D, i) = |\{d \in D \mid y_{true}(d) = i\}| \tag{3}$$

$$F1(D, \phi, i) = \frac{2 \cdot precision(i) \cdot recall(i)}{precision(i) + recall(i)}, \tag{4}$$

$$precision(i) = \frac{TP(i)}{TP(i) + FP(i)}, \tag{5}$$

$$recall(i) = \frac{TP(i)}{TP(i) + FN(i)}, \tag{6}$$

$$TP(i) = |\{d \in D \mid y_{pred}(d, \phi) = i, y_{true} = i\}|, \tag{7}$$

$$FP(i) = |\{d \in D \mid y_{pred}(d, \phi) = i, y_{true} \neq i\}|, \tag{8}$$

$$FN(i) = |\{d \in D \mid y_{pred}(d, \phi) \neq i, y_{true} = i\}|. \tag{9}$$

which is the weighted F1-score of the classification on corpus D. The distinctness score represents how well the given label assignment ϕ induces the correct classification from the pre-trained zero-shot classification model. The zero-shot classification model is assumed to represent a common sense association of a document with a topic label. A higher distinctness score is expected to indicate that the given topic labels ϕ are more natural for labels to classify documents.

3.3 Combinatorial Optimization

The goal of the proposed pragmatic listener-speaker method is to find the label set with the highest distinctness score $\phi_{best} = \arg\max f(D, \phi)$. To brute force through $\prod_i^m n_i$ combinations is impossible when m and n_i are large. In order to find the ϕ_{best}, we leverage local search [35] for combinatorial optimization.

Let (Φ, f) be the combinatorial optimization problem, where the score function f is a mapping from the set of solutions Φ to $\mathbb{R}$. We define a neighborhood function $\mathcal{N}(\phi)$ by changing one label l_i in label set ϕ to another label candidate l_i' for that topic:

$$\mathcal{N}(\phi = (l_1, \ldots, l_m)) = \bigcup_{i \in [1..m]} \{(l_1', \ldots, l_m') \mid l_i' \in L_i \wedge \forall j \in [1..m] \backslash i (l_j' = l_j)\}. \tag{10}$$

A local optimum is a solution $\hat{\phi}$ where the score of it is no worse than its neighbors:

$$f(\hat{\phi}) \geqq f(\phi) \quad \forall \phi \in \mathcal{N}(\hat{\phi}). \tag{11}$$

We use the simple local search algorithm by iterating through neighbors for each topic label until a local optimum is found. The pseudo-code is shown in Algorithm 1.

Algorithm 1 Local search for topic labels.

```
function LOCAL_SEARCH(S, D, m, f)
    φ̂ ← RandomSampling(S)
    BestScore ← f(D, φ̂)
    Checked ← [False for x in range(0, m)]
    repeat
        for i in range(0, m) do φ ← φ̂
            for candidate in S[i] do φ[i] ← candidate
                if f(D, φ) > BestScore then
                    φ̂ ← φ
                    BestScore ← f(D, φ)
                    Checked ← [False for x in range(0, m)]
                end if
            end for
            Checked[i] ← True
        end for
    until False not in Checked
    return φ̂
end function
```

3.4 Pragmatic Topic Label Pipeline

The pragmatic topic label pipeline could leverage different automatic topic labeling methods in an ensemble learning manner. After applying topic modeling to corpus D and obtaining topic distribution T, they will be passed into off-the-shelf automatic labeling methods. The label candidates $ATL_j(D, T) = (s_{j,1}, \ldots, s_{j,m})$ will be obtained, and the sets from different automatic topic labeling methods will then be concatenated to generate the input for the listener-speaker model,

$$S = (\bigcup_j s_{j,1}, \ldots, \bigcup_j s_{j,m}) \tag{12}$$

where ATL is an off-the-shelf automatic labeling method and j is the number of different automatic topic labeling methods used.

The pipeline uses the zero-shot classification of text [36] for the distinctness score calculation. Zero-shot classification involves predicting a class that the model has not encountered during training. This approach could be thought of as transfer learning by utilizing a pre-trained language model. The model is provided with a prompt and a natural language text sequence that describes the task, classification target, and label set. In practice, it is considered reasonable to utilize language models that are pre-trained on natural language inference (NLI) tasks.

After the combinatorial optimization process through the

pragmatic listener-speaker model, the optimized label set would then be outputted as the final label set for the topic.

4 Experiments and Discussion

4.1 Experiment Settings

4.1.1 Corpus

We use the WikiText-2 dataset [37] for evaluation. The WikiText-2 dataset is extracted from the set of verified English articles on Wikipedia. It was introduced in 2016 and contains approximately 2.5 million tokens in total. The articles are in their raw form which retains numbers, cases, and punctuations. Since topic modeling is an unsupervised machine learning method, all three splits (test, train, validation) are used in topic modeling. In experiments, each paragraph is treated as a document in the training step since the articles are too long to fit in the language model used in the topic model and the zero-shot classification pipeline.

4.1.2 Topic Modeling

We choose BERTopic [5] as our topic modeling method. BERTopic leverages transformers and c-TF-IDF to create dense clusters, allowing for easily interpretable topics whilst keeping important words in the topic descriptions. It has the benefits of being one of the state-of-the-art topic modeling methods on various corpus, competence in different settings, and user-friendly APIs.

The BERTopic model generates topic representations through the following steps.

（1） **Embedding representation**: Each document is converted to its embedding representation using a pre-trained language model. We used the `sentence-transformers/all-mpnet-base-v2` model from Huggingface [38], which is a pre-trained Sentence-BERT [31] model based on MPNet [39].

（2） **Dimension reduction**: The dimensionality of the resulting document embeddings is reduced using UMAP [40] to optimize the clustering process.

（3） **Document clustering**: The reduced embeddings are clustered using HDBSCAN [41]. We apply hierarchical clustering to set the number of topics to 40. Although HDBSCAN is a soft-clustering approach allowing noise to be modeled as outliers, in order to avoid too many outliers affecting the consistency of the model and match the topic representation of traditional topic models, we apply outlier reduction using firstly class TF-IDF (c-TF-IDF) representations with threshold = 0.05 and then the topic distributions.

（4） **Topic representation**: The c-TF-IDF score for each term is calculated by treating each document cluster as a big document. This procedure models the importance of words in clusters and generates topic-word distributions for each cluster of documents.

4.1.3 Zero-shot Classification

We use the Huggingface zero-shot classification pipeline [42] in our listener model. We choose the `valhalla/distil bart-mnli-12-1` model as the pre-trained language model, which is the distilled version of `bart-large-mnli` BART [11] model. The batch number is set to 1.

4.1.4 Automatic Topic Labeling Baselines

To evaluate our proposed pragmatic model and generate label candidates for the automatic topic labeling pipeline, we choose the following automatic topic labeling methods as baselines.

- **Top terms**: The topic representations generated by the BERTopic model are used as simple topic labels. The top 4 words of each topic are concatenated into one label.
- **KPE**: Following the idea of utilizing automatic summarization technology for labeling topic models while trying to restrict the lexical length of the labels, we apply YAKE [43], an off-the-shelf key-phrase extraction (KPE) method on top 50 documents of each topic. The max lexical length of the generated labels is set to 3.
- **NETL**: We choose Neural Embeddings Automatic Labeling (NETL) [23] as the baseline which stands for the methods that employ external corpora like Wikipedia and leverage supervised learning techniques to rank candidates. The top 10 words of each topic are passed to the model to generate label candidates.
- **LLM**: Regarding the advance of large language models, we leverage ChatGPT [44] as automatic topic labeling methods by simply passing the top 10 words to the `gpt3.5-turbo-1106` model and prompting to ask the model to generate topic labels based on them.
- **Human labels**: An expert in topic modeling is asked to assign a topic label after reading the top 10 words and top 20 documents for each topic.

4.2 Listener Model Evaluation

To evaluate the zero-shot distinctness score in the proposed pragmatic listener model, we conduct crowd-sourcing topic labeling tasks on Amazon Mechanical Turk (AMT). Participants in the AMT tasks are provided with a document in the corpus and five topic labels. The topic labels are the labels of the five closest topics to the document (including the corresponding topic), generated by one of the baselines. Participants are asked to select the most relevant label for the text. After filtering out short documents, 30 documents are sampled from each of the 40 topics to generate a batch of 1200 tasks. For each baseline method, one batch is published. A demo of the AMT tasks is shown in Figure 2. The result is gathered after the tasks are completed in AMT crowd-sourcing. After the result is gathered, we calculate the F1-score in crowd-sourcing of each topic among each la-

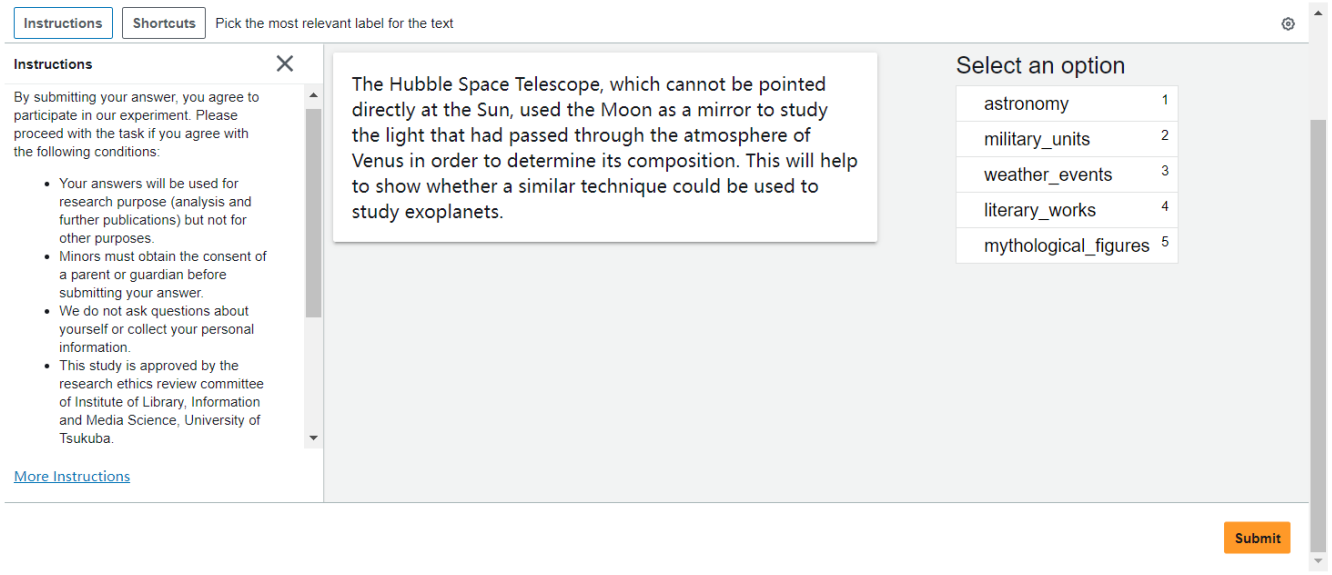

Figure 2 A demo of the Amazon Mechanical Turk crowd-sourcing tasks. Participants are asked to select the most relevant label to the text among five choices.

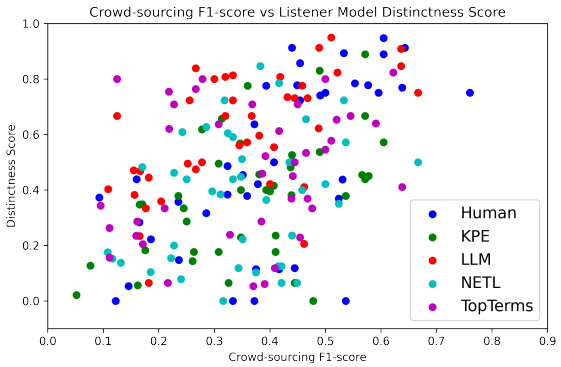

Figure 3 A scatter plot of topic labels evaluation on each topic labeling baseline. Each point on the graph stands for a topic.

bel set, as long as the weighted overall F1-score of each label set.

The same sampled corpus and label sets are then passed into the proposed listener model for evaluation. The distinctness score of each topic among each label set and the overall distinctness score are gathered. We conduct a correlation analysis between the crowd-sourcing result and the listener model result. The result is shown in Figure 3 and Table 1. The correlation between the crowd-sourcing F1-score and distinctness score is $r = 0.4299$, with $p < 0.0001$, consistent with our hypothesis that there is a positive correlation between the crowd-sourcing F1-score and distinctness score.

4. 3 Automatic Labeling Pipeline Evaluation

To evaluate our proposed automatic labeling pipeline, we utilize our proposed pipeline to generate the label set for the corpus. Four out of five of the baselines (**Top Terms**, **KPE**, **NETL**, **LLM**) are used to generate label candidates for the automatic labeling pipeline. We run two different settings on our proposed pipeline.

Table 1 The overall score (weighted-average) of each topic labeling baseline compared between crowd-sourcing F1-score and distinctness score.

	Crowd-sourcing F1-score	**Distinctness Score**
Human Labels	0.4092	0.4903
KPE	0.3653	0.3782
LLM	0.3475	0.6061
NETL	0.3466	0.3722
Top Terms	0.3597	0.4658

- **ppl_4**: We use the exact labels outputted by baselines in Section 4. 2, comprising 4 candidates for each topic. 30 documents from each of the 40 topics are sampled from the corpus to generate a subcorpus of 1200 documents.(注1)
- **ppl_10**: Top 3 candidate labels outputted by each of **KPE**, **NETL**, **LLM** and the label generated by **Top Terms** comprise 10 candidates for each topic. The same subcorpus used in **ppl_4** is used.

After the two label sets are generated by the proposed pipeline, we evaluate them by conducting the same AMT crowd-sourcing tasks in Section 4. 2. To remove the effect of uninterpretable topics, we filter out topics which answer accuracy is lower than 0.4 on **Human labels** label sets in crowd-sourcing tasks. The crowd-sourcing F1-score results on baselines and proposed methods after filtering are shown

(注1) : We use hyperparameter `random_seed` to make sure the documents are sampled differently to the subcorpus in Section 4. 2

in Table 2.

Table 2 Crowd-sourcing F1-score on filtered topics.

	Crowd-sourcing F1-score
Baselines	
Human	0.4792
KPE	0.3508
LLM	0.3645
NETL	0.3654
Top Terms	0.3398
Proposed Method	
ppl_4	0.3172
ppl_10	0.3762

4.4 Discussion

According to the result in Table 1, in terms of crowd-sourcing F1-score, **Human labels** performs the best, obtaining a score of 0.4092. **KPE** and **Top Terms** perform a bit weaker, while **LLM** and **NETL** perform the worst. This may be due to the fact that both **KPE** and **Top Terms** labels are generated from the corpus, while **LLM** and **NETL** labels are generated from the top terms. The labels of **LLM** and **NETL** are more general terms of subjects, while the labels of **KPE** and **Top Terms** tend to consist of specific objects. Additionally, the labels of **KPE** and **Top Terms** contain more words on average compared to other baselines. This nature helps participants choose the correct labels once the words in labels appear in the document, even if the document is weakly related to the topic.

In terms of the distinctness score outputted by the proposed listener model, the labels of **LLM** score the highest of 0.6061, followed by **Human Labels** of 0.4903 and **Top Terms** of 0.4658. This may be because while being the most similar label with **Human Labels**, **LLM** produces multiple concepts in a single label (e.g., "Historical architecture" compared to "Archaeology, Historical Sites"), which helps the zero-shot classification in matching the documents. This could also be proved by the score of **Top Terms** since each word could be interpreted as an independent concept. On the other hand, the zero-shot classification method tends to give higher scores to common-sense labels (e.g., "Historical architecture" compared to "sun temple"), making the score of **KPE** and **NETL** relatively low. This may be due to the choice of the language model or the prompt used in the zero-shot classification pipeline, which required further investigation.

Nonetheless, despite the different aspects between the crowd-sourcing F1-score and the distinctness score assigned by the proposed method, the correlation analysis shows a strong belief that there is a positive correlation between the crowd-sourcing F1-score and distinctness score, addresses that the proposed listener model could serve as a metric to assess the quality of label sets in the context of the entire corpus.

During the listener model evaluation, we noticed that some of the **Human labels** have exceptionally low accuracy in crowd-sourcing tasks, some of them even lower than the expected accuracy of random choices. This phenomenon warns us of the possibility that the topic model could output improper topics, that even humans could not assign proper labels to them. We consider this as a problem of topic modeling and therefore exclude these topics by filtering out them with crowd-sourcing accuracy on **Human labels** during the automatic labeling pipeline evaluation.

Before we discuss the result of the automatic labeling pipeline evaluation, two deficiencies in the proposed automatic topic labeling pipeline need to be addressed. First, the problem of the local optimum. Since we chose local search for combinatorial optimization, there is a possibility that the outputted label set is a local optimum, rather than the global optimum with the highest distinctness score. Approaches such as simulated annealing, tabu search, and genetic algorithms are designed to solve this problem. However, all of those approaches require branching for worse neighbors or finding multiple optimums in order to overcome the problem of local optimum. This brings us to the second problem of the computational cost. As mentioned in Section 3.4, zero-shot classification involves querying modern language models with prompts. The computational cost of such querying is extremely high compared to traditional classification methods. At the same time, combinatorial optimization executes a huge amount of such classification in order to compare the scores between neighbors. This nature makes the computational cost of the proposed automatic topic labeling pipeline significantly high, which is the reason for sampling a sub-corpus for the pipeline, restricting the number of label candidates, and using a distilled version of the language model during the experiments. As for now, there is no proper solution other than increasing the computational power or reconstructing the off-the-shelf zero-shot classification method for the proposed pipeline.

After uninterpretable topics are filtered out, according to Table 2, the best score is 0.4792 of **Human labels**, followed by 0.3762 of the proposed method **ppl_10**. **ppl_4** obtain the worst score of 0.3172. This may indicate that the number of candidates provided to the pipeline affects the performance by a large margin, even if those candidates are not been chosen as the best choice according to the off-the-shelf automatic topic labeling methods. This phenomenon shows that in this specific problem setting of combinatorial optimization, a better solution is more likely to be found with the

naive local search method in a larger combination set. On the other hand, the proposed method **ppl_10** outperforms other baselines, showcasing the effectiveness of our proposed automatic topic labeling pipeline.

5 Conclusion

In this research, we proposed a pragmatic automatic topic labeling approach that evaluates the ambiguity degree of topic labels in the holistic corpus. The pragmatic automatic topic labeling approach consists of two parts, the listener model and the speaker model, leverage zero-shot classification and combinatorial optimization respectively, to output the pragmatically proper label set among the input label candidates. Our crowd-sourcing experiments demonstrated the correlation between our pragmatic approach and human evaluation, proving that the listener model serves as a metric to assess the quality of label set in the aspect of distinctness. Our automatic topic labeling pipeline experiments showed that the proposed method is effective in improving the quality of the topic label set under certain circumstances.

The main drawback of this work is that the proposed method relies on the zero-shot classification of text, whose computational cost weighs on the efficiency of the entire pipeline. As this research explores the idea of the pragmatic approach in automatic topic labeling, future work should focus on exploring other methods or training a lightweight designated machine learning model to achieve the same or better performance in this task.

Acknowledgment

This work was supported by JSPS KAKENHI #21H03552, #22K12039 and JST CREST #JPMJCR22M2.

References

[1] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pp. 50–57, New York, NY, USA, August 1999. Association for Computing Machinery.

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

[3] Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 1727–1736. JMLR.org, 2016.

[4] Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. Neural topic modeling with bidirectional adversarial training. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 340–350, Online, July 2020. Association for Computational Linguistics.

[5] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[6] Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. Representing topics labels for exploring digital libraries. In *IEEE/ACM Joint Conference on Digital Libraries*, pp. 239–248. IEEE, September 2014.

[7] Qiaozhu Mei, Xuehua Shen, and Chengxiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pp. 490–499, New York, NY, USA, August 2007. Association for Computing Machinery.

[8] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1536–1545, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[9] Areej Alokaili, Nikolaos Aletras, and Mark Stevenson. Automatic generation of topic labels. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pp. 1965–1968, New York, NY, USA, July 2020. Association for Computing Machinery.

[10] Cristian Popa and Traian Rebedea. BART-TL: Weakly-supervised topic label generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Stroudsburg, PA, USA, 2021. Association for Computational Linguistics.

[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.

[12] David Blei and John Lafferty. Correlated topic models. *Adv. Neural Inf. Process. Syst.*, Vol. 18, p. 147, 2006.

[13] Lan Du, Wray Buntine, and Mark Johnson. 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*, pp. 190–200. aclweb.org, 2013.

[14] Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 25. Curran Associates, Inc., 2012.

[15] Qingqing Long, Yilun Jin, Guojie Song, Yi Li, and Wei Lin. Graph structural-topic neural network. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 1065–1073, New York, NY, USA, August 2020. Association for Computing Machinery.

[16] Jeremy Costello and Marek Reformat. Reinforcement learning for topic models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4332–4351, July 2023.

[17] Thong Thanh Nguyen and Anh Tuan Luu. Contrastive learning for neural topic model. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, December 2021.

[18] Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. Neural topic modeling with bidirectional adversarial training. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors,

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 340–350, Online, July 2020. Association for Computational Linguistics.

[19] Dimo Angelov. Top2Vec: Distributed representations of topics. August 2020.

[20] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. May 2014.

[21] Suzanna Sia, Ayush Dalmia, and Sabrina J Mielke. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! pp. 1728–1736, November 2020.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[23] Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. Automatic labelling of topics with neural embeddings. December 2016.

[24] Amparo Elizabeth Cano Basave, Yulan He, and Ruifeng Xu. Automatic labelling of topic models learned from Twitter by summarisation. In Kristina Toutanova and Hua Wu, editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 618–624, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[25] Xiaojun Wan and Tianming Wang. Automatic labeling of topic models using text summaries. In Katrin Erk and Noah A Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2297–2305, Berlin, Germany, August 2016. Association for Computational Linguistics.

[26] Mohamad Hardyman Barawi, Chenghua Lin, and Advaith Siddharthan. Automatically labelling Sentiment-Bearing topics with descriptive sentence labels. In *International Conference on Applications of Natural Language to Information Systems*, pp. 299–312. unknown, June 2017.

[27] Dongbin He, Minjuan Wang, Abdul Mateen Khattak, Li Zhang, and Wanlin Gao. Automatic labeling of topic models using graph-based ranking. *IEEE Access*, Vol. 7, pp. 131593–131608, 2019.

[28] Domenic Rosati. Moving beyond word lists: towards abstractive topic labels for human-like topics of scientific documents. October 2022.

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

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. June 2017.

[31] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics.

[32] Tom Williams, Gordon Briggs, Bradley Oosterveld, and Matthias Scheutz. Going beyond literal command-based instructions: extending robotic natural language interaction capabilities. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 1387–1393. AAAI Press, January 2015.

[33] Daniel Fried, Jacob Andreas, and Dan Klein. Unified pragmatic models for generating and following instructions. In *North American Chapter of the Association for Computational Linguistics*, 2017.

[34] Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. Pragmatics in language grounding: Phenomena, tasks, and modeling approaches. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12619–12640, Singapore, December 2023. Association for Computational Linguistics.

[35] Aarts Emile H L. and Jan Karel Lenstra. *Local search in combinatorial optimization*. Princeton Univ. Press, 2003.

[36] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3914–3923, Hong Kong, China, November 2019. Association for Computational Linguistics.

[37] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. 2016.

[38] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M Rush. HuggingFace's transformers: State-of-the-art natural language processing. October 2019.

[39] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, No. Article 1414 in NIPS'20, pp. 16857–16867, Red Hook, NY, USA, December 2020. Curran Associates Inc.

[40] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *J. Open Source Softw.*, Vol. 3, No. 29, p. 861, 2018.

[41] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, Vol. 2, No. 11, p. 205, 2017.

[42] Hugging Face. What is zero-shot classification? - hugging face, 2023.

[43] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. YAKE! keyword extraction from single documents using multiple local features. *Inf. Sci.*, Vol. 509, pp. 257–289, January 2020.

[44] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and Others. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.*, Vol. 33, pp. 1877–1901, 2020.