EDEntail: An Entailment-based Few-shot Text Classification with Extensional Definition

Zhu Zixiao[1,3] **Junlang Qian**[2] **Zijian Feng**[1,3] **Hanzhang Zhou**[1,3] **Kezhi Mao**[2]*
[1]Institute of Catastrophe Risk Management, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore
[2]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[3]Future Resilient Systems Programme, Singapore-ETH Centre, CREATE campus, Singapore
{zixiao001, junlang001, feng0119, hanzhang001}@e.ntu.edu.sg, ekzmao@ntu.edu.sg

Abstract

Few-shot text classification has seen significant advancements, particularly with entailment-based methods, which typically use either class labels or intensional definitions of class labels in hypotheses for label semantics expression. In this paper, we propose EDEntail, a method that employs extensional definition (EDef) of class labels in hypotheses, aiming to express the semantics of class labels more explicitly. To achieve the above goal, we develop an algorithm to gather and select extensional descriptive words of class labels and then order and format them into a sequence to form hypotheses. Our method has been evaluated and compared with state-of-the-art models on five classification datasets. The results demonstrate that our approach surpasses the supervised-learning methods and prompt-based methods under the few-shot setting, which underlines the potential of using an extensional definition of class labels for entailment-based few-shot text classification. Our code is available at https://github.com/MidiyaZhu/EDEntail.

1 Introduction

Entailment-based text classification formulates heterogeneous classification tasks into a unified textual entailment problem (Dagan et al., 2005; Zhang et al., 2023a). Unlike traditional classification models that often encode class labels into numerical vectors such as one-hot vectors without considering label semantics (Zhang et al., 2018), the entailment-based approaches express the semantics of class labels in the hypothesis and classify the input texts through semantic entailment matching between the input texts (premise) and the hypothesis.

In hypothesis construction, class and subclass labels are commonly used as descriptive words for label semantics representation (Schopf et al., 2020). The intensional definition of a class label by WordNet, which specifies the necessary and sufficient conditions of the class label (Cook, 2009), is also frequently utilized to provide label information (Yin et al., 2019). While such use of label semantic representations in hypothesis might be effective for some tasks, it may fall short in others because **a single descriptive word or an intensional definition may not encapsulate all the semantic meanings within a label's domain, however, an effective hypothesis is expected to deduce various premises from the label domain.** As shown in rows 1-4 in Figure 1, hypotheses constructed from the premises' class label fail to properly entail the given four premises in both language model and human cognition. Besides, **descriptive words that exhibit clear semantic entailment relationships in human cognition may exhibit dissimilarity in the word embedding space** (Zhu and Mao, 2023). The relationship learned from the same descriptive word might vary much in the word embedding space with different contexts (see Section 4.7), hindering entailment feature learning. These limitations may weaken the entailment-based text classification performance, particularly in zero-shot or few-shot scenarios, when the test set lacks alignment with the hypothesis or when test samples fail to transfer the relation learned from the training samples.

To address the above limitations in label semantics expression, we propose to construct hypotheses using extensional definition of class labels. An extensional definition, which gives the meaning of a term by listing all descriptive words that fall under this term [1], provides explicit and diverse information of label semantics, facilitating semantic matching between different premises and a hypothesis.

As illustrated in the last row of Figure 1, the label extensional definition-based hypothesis can

*Corresponding author

[1]https://en.wikipedia.org/wiki/Extensional_and_intensional_definitions

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1124–1137

Figure 1: The entailment probability for four negative premises from SST-2 using four hypotheses: class label, subclass label, intensional definition, and extensional definition. Red means an entailment probability over 50% (entailment), while blue is below 50% (contradiction). PLM values are given by 'roberta-large-mnli' and Human values are averaged from 20 questionnaires.

entail all premises in both PLM and human cognition. This not only broadens the semantics of class labels but also ensures computational efficiency by avoiding multi-time subclasses matching. At the linguistic level, descriptive words in the extensional definition can collaboratively reduce polysemy issues by refining semantic meanings with each other. At the machine-learning level, descriptive words in the extensional definition can boost embedding consistency among each other through the contextual learning capacity (see Section 4.7).

To implement the extensional definition-based hypothesis construction, we develop a generation method to ensure the descriptive words selected are representative and concise, considering that there could exist too many example words in the extensional definition of a class label and inclusion of all of them in hypotheses is impractical while randomly selection may result in insufficient coverage of the example words.

Our contributions are summarized below:

1. By analyzing current entailment-based methods, we identify that the semantic expression of labels within the existing hypothesis construction tends to be limited. This results in narrow coverage of label information and word embedding inconsistency in feature learning, hindering the performance of premise-hypothesis entailment.

2. We present **EDEntail**, an entailment-based approach to few-shot text classification that utilizes extensional definitions of class labels. A systematic method for extensional definition generation is designed to provide diverse informative label signals in hypothesis construction for premise-hypothesis entailment relationship enhancement.

3. Extensive experiments across various classification datasets demonstrate that EDEntail outperforms other state-of-the-art models in few-shot settings.

2 Related Work

Meta-task exhibits significant potential in zero-shot or few-shot text classification tasks. It can be classified into the generative method and the discriminative method (Zhang et al., 2023a).

Generative methods treat every task as a text-to-text generation problem. Prompt-based method as a generation method that treats the meta-task as a masked language modeling (MLM) problem (Schick and Schütze, 2021; Gao et al., 2021). The MLM model predicts the masked token and then maps the predicted token to the label space through verbalizers. The prompt-based methods exhibit remarkable success in few-shot classification tasks (Zhao et al., 2021; Zhang et al., 2023b). Although knowledge can be incorporated into prompt verbalizer to enhance projection performance (Hu et al., 2022), the MLM model prediction may fall out of all possible associated candidates in verbalizer projection (Zhang et al., 2023a).

Discriminative methods like entailment-based method formulates meta-task under the framework of Natural Language Inference, which aims to determine the relationship between the premise and the hypothesis as 'entailment', 'contradiction', or 'neutral' (Yang et al., 2023). Recent studies in entailment-based methods include language model training (Devlin et al., 2018; Liu et al., 2019; Pàmies et al., 2023), pseudo-label training (Ge et al., 2023; Gera et al., 2022), classifier training (Xia et al., 2022; Zhang et al., 2023a; Wang et al., 2022b), and hypothesis engineering. The latter

delves into the effective use of class label names (Plaza-del Arco et al., 2022) or intensional definitions (Lamanov et al., 2022), or strategies for incorporating the above label information into ensemble models (Basile et al., 2021). One challenge here is the relationship learning between premise and hypothesis with extremely limited training sources (Mayer et al., 2023), and another challenge is the construction of hypotheses suitable for all tasks (Gera et al., 2022). Enhancing classification performance and stability under limited training data is a focal research point (Ma et al., 2021; Min et al., 2022). In this paper, we propose a new method for hypothesis construction, aiming to address the aforementioned challenges in entailment-based text classification.

3 Proposed Methods

In this section, we introduce our proposed method, **EDEntail**, as depicted in Figure 2. This method comprises three key modules: extensional definition (**EDef**) generation by gathering and selecting relevant and representative descriptive words; hypothesis construction by ordering and formatting words of extensional definition; and entailment reformulation for few-shot text classification.

3.1 EDef Generation

Vocabulary Construction To establish EDef, we require a vocabulary for each class label that contains descriptive words to offer additional classification prompts.

Our goal for constructing the vocabulary is to elicit label semantics in an extensional and comprehensive manner. We found that the available open-sourced vocabularies are noisy with overlapping words between different classes. Thus, simply crawling descriptive words based on the concepts (like ConceptNet) might cause polysemy concerns in label names. Additionally, the WordNet synsets might cause irrelevant words upon extending the synonym search (e.g., 'anger' leading to 'temper' and then 'humour').

To build a high-quality vocabulary and ensure generalizability, we employ dictionary resources and ChatGPT[2]. We presume the class label names can describe the classification task and initialize them as the prompt words for descriptive words searching. We first extract the definitions of the class label words specific to our tasks from the Oxford English Dictionary[3]. Combining these definitions, we configure prompts within ChatGPT to obtain the label vocabulary. The prompts input for ChatGPT is listed in Appendix A. The vocabulary of each class label is assembled from words produced by ChatGPT with its corresponding prompts.

Descriptive Words Clustering To maximize the effectiveness of EDef in conveying semantics of label, we employ the K-means algorithm (Hartigan and Wong, 1979) to cluster descriptive words for each class label vocabulary in the embedding space. This reduces the number of descriptive words but guarantees comprehensive expression of the semantics of class labels.

Firstly, we obtain the [CLS] embedding of each descriptive word using Roberta-large with single-word input. Then, all the embeddings are clustered using the K-means clustering algorithm. For the i-th cluster U_i, the word that is the closest to the cluster centre is chosen as the representative word of the i-th cluster as shown in Eqn 1, where R_i is the centre of the i-th cluster, w is the word in U_i and $dist$ is the Euclidean distance.

$$O_i = argmin_{w \in U_i} dist(w, R_i) \tag{1}$$

This procedure is iterated multiple times, say ten times, and the ultimately selected extensional descriptive words are those representative ones with the silhouette score S close to 1 as shown in Eqn 2, where a_i is the average distance from O_i to other words in the same cluster, b_i is the minimum average distance from O_i to other words in the different clusters, and n is the cluster number.

$$S = average \sum_{i=1}^{n} \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2}$$

The extensional descriptive words results of each label l are saved in a set (l, n) with each cluster number n defined in the K-means algorithm.

3.2 Hypothesis Construction

EDef Words Ordering After selecting representative words for extensional definition, the words are arranged based on a certain order to build hypotheses. The order of words in the sequence should, to the greatest extent, activate keywords in the premise. The knowledge used for ordering can be obtained from either the local dataset and the language model, or external sources. We next introduce two ordering methods.

[2] https://chat.openai.com/(free 3.5 version)

[3] https://www.oed.com

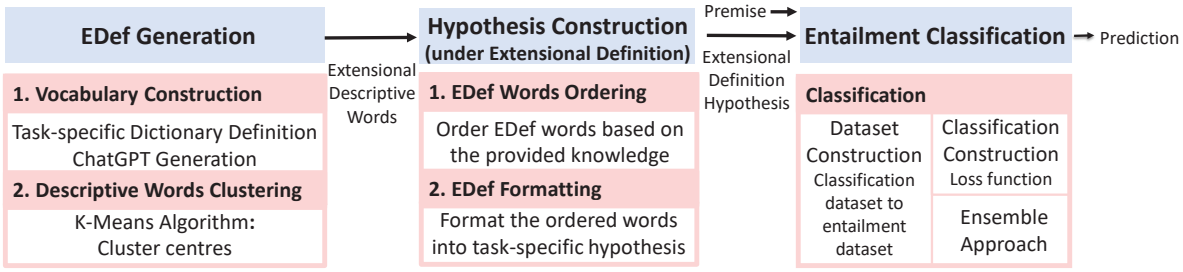


Figure 2: Overall architecture of EDEntail; it utilizes EDef in the hypothesis for entailment classification learning.

Ordering Based on Entailment Knowledge For local knowledge, considering that word features differ across datasets, we utilize the pre-trained language model's entailment knowledge between the dataset and relevant words to match the extensional definition with the specific dataset.

For each extensional descriptive word in (l, n), we first encode the word into the hypothesis, and then derive the average zero-shot entailment probability by entailing it with the pruned set $D^l \subset D_{train}$, where D_{train} is the few-shot training set. Finally, we order the extensional descriptive words in each (l, n) from the largest probability to the smallest entailment probability.

Ordering Based on Frequency Knowledge For external knowledge, we evaluate various sources like frequency, silhouette scores, and LLM prompting. Based on robustness and linguistic reliability, we choose word frequency knowledge for word ordering, which is 'the one common feature of nearly all measures of lexical prevalence created to date' (Egbert and Burch, 2023) and is vital in human word ranking experiments (Battig and Montague, 1969). We chose Google Ngram for its convenience and superior performance.

For each extensional descriptive word in set of (l, n), we search its latest usage frequency in Google Ngram Viewer[4] and order extensional descriptive words in each (l, n) from the most frequent to the least frequent one.

EDef Formatting To format the ordered descriptive words into the hypotheses, we connect the descriptive words in 4 ways as shown below, where e_j indicates the j-th descriptive word:

1. Connect with comma (EDef-CC):'$e_1, \ldots, e_n$'
2. Connect with space (EDef-CS):'$e_1 \ldots e_n$'
3. Connect with slash (EDef-CL): 'e_1/ $\ldots$/ e_n'
4. Connect with and (EDef-CA): 'e_1 and $\ldots$ and e_n'

We use EDef-CC and EDef-CS because comma and space are the commonly used connectors in writing. We use EDef-CL and EDef-CA because slash / and "and" are often used to denote OR-relationship and AND-relationship, respectively (Woo, 2019). We design OR-relationship connection for tasks that cover coase but encompassing categorical delineations, like sentiment analysis. AND-relationship connection is for tasks focusing on detailed and fine-grained subjects, like emotion recognition

The extensional descriptive words ordered by entailment knowledge or frequency knowledge are encoded into the above formats with the hypothesis defined for each application. For example, for sentiment analysis, a positive EDef-CC hypothesis might be 'a proactive, constructive, relief, encouraging one'.

3.3 Entailment Classification

Since all classification tasks are viewed as entailment tasks and the labels in the entailment-based method are 'entailment' (E), 'contradiction' (C), and 'neutral' (N) instead of classification labels, we need to adapt the three-class entailment approach to accommodate multi-class objectives.

Reformulation We reconstruct the training classification datasets for entailment-based label consistency. A classification dataset by pairing each text with hypotheses formed from an EDef set. For each text-hypothesis pair, it assigns a label of 'entailment' if the EDef's label entails the text's label, and 'contradiction' otherwise. This results in a new dataset tailored for entailment approach analysis.

We reformulate the loss function to ensure that it is applicable in language models pre-trained on entailment datasets as Eqn 3, where $l(N)$, $l(C)$, and $l(E)$ represent the loss components (logits) derived from pre-trained language models (PLMs) after a softmax function. $loss_{BCE}$ is the Binary Cross Entropy Loss. We regard both 'neutral' and 'contradiction' as non-entailment labels that is rep-

[4] https://books.google.com/ngrams/json

resented by the one with the highest logits value from the PLM, aiming to mitigate any potential bias caused by the absence of a class in training [5].

$$loss = loss_{BCE}\left(max\left\{l(N), l(C)\right\}, l(E)\right) \quad (3)$$

The entailment classes map back to classification labels depending on the entailment probability between the text and the class of EDef in the hypothesis. The class with maximum entailment probability is chosen as the prediction of the text.

Ensemble Approach The majority of meta-tasks use a single type of hypothesis or prompt in both training and testing scenarios. They ensemble various models trained under diverse hypotheses or prompts to improve classification performance (Hu et al., 2022; Plaza-del Arco et al., 2022). Instead of ensembling models, we ensemble the hypothesis used in both training and testing sets. Specifically, each sample in the training set is connected with different hypotheses, respectively. Consequently, the testing set's sample pool is similarly expanded through connections with these hypotheses. The next section shows the performance of EDEntail with and without ensembling.

4 Experiments

In this section, we examine EDEntail's capabilities from the following perspectives: (1) The classification performance compared to other advanced classification models (Section 4.3); (2) The classification performance compared with other entailment-based models (Section 4.4); (3) The efficiency of utilizing limited training samples (section 4.5); (4) The performance under zero-shot setting (Section 4.6); (5) The effect of extensional definition on word embedding consistency (section 4.7); and (6) The classification performance compared with large language models (Section 4.8).

4.1 Datasets

We evaluate our method on two sentiment datasets: SST-2 (Socher et al., 2013) and CR (Ding et al., 2008), two emotion recognition datasets: MELD (Poria et al., 2018) (textual data only) and AMAN (Aman and Szpakowicz, 2008), and one question classification dataset: TREC-6 (Li and Roth, 2002).

The hypotheses that we use for each dataset are listed in Table 1. For each task, we provide four types of commonly used hypothesis structures.

[5]We exclude 'neutral' as there is no clear linguistic basis for manually identifying this category and the used PLM also excludes this category in exampled usage in huggingface.

The details of the dataset configuration can be found in Appendix B.

Task	Hypothesis	Class
SST2 CR	A <EDef>piece of work . A <EDef>one. A <EDef>piece. All in all <EDef>.	positive, negative
AMAN MELD	A <EDef>piece of work . A <EDef>one. A <EDef>piece. It was <EDef>!	angry, disgust, happy, neutral, surprise, sad, and fear
TREC	It is <EDef>. It was <EDef>news. Why <EDef>? Answer: <EDef>.	location, numeric, description, entity, human, and abbreviation

Table 1: The hypotheses used in each dataset. <EDef>is where the extensional definition is placed.

4.2 Experiment Settings

We conduct both zero-shot learning and few-shot learning experiments under each hypothesis. The experiments are under the N-way-K-shot training setting (Wang et al., 2022a) while the size of validation is the same as the size of the training set (Wang et al., 2021). In few-shot experiments, we designated N=5 and K=[1, 16, 32]. This means that for both the training and validation datasets, we randomly selected K samples for each label, repeating this process five times within five different training sets and corresponding validation sets. The EDef length n, namely the number of extensional descriptive words in EDef, is grid-searched over $\{1, \cdots, 10\}$ under few-shot settings. In few-shot or zero-shot learning experiments, the n for each label in one dataset is the same. The reported results are the average of five repeated experiments. The robustness is evaluated based on the standard deviation of the five results.

The compared baseline models are two supervised models: Finetune (FT) and DualCL (Kumar and Raman, 2022), three entailment-based models: EFL (Wang et al., 2021), Label-Entail (Plaza-del Arco et al., 2022) and IDef-Entail, and four prompt-based models: PET (Schick and Schütze, 2021), WARP (Hambardzumyan et al., 2021), LM-BFF (Gao et al., 2021), and KPT (Hu et al., 2022).

Detailed information on the baseline models and the EDEntail implementation is provided in Appendix C.1 and Appendix C.2, respectively.

4.3 Overall Results

Table 2 summarized the results of the baseline models and our approach under Frequency knowledge

	Model	SST2	CR	AMAN	MELD	TREC
K=1	Finetune	49.7(0.5)	52.6(2.7)	27.5(27.3)	21.1(13.2)	21.6(3.9)
	DualCL	53.1(0.7)	51.4(3.2)	23.4(2.3)	19.6(0.9)	19.9(0.8)
	EFL	69.3(15.2)	57.5(8.2)	33.8(23.3)	28.2(7.5)	22.2(8.8)
	Label-Entail	88.2(2.1)	90.4(0.5)	35.2(9.2)	29.0(4.6)	42.8(11.6)
	IDef-Entail	70.9(4.3)	79.7(7.5)	35.7(13.3)	17.5(5.1)	29.0(5.9)
	PET	74.2(3.6)	64.7(12.5)	28.6(0.7)	22.1(1.7)	29.4(2.3)
	WRAP	85.6(4.6)	68.4(6.5)	24.3(3.6)	21.0(8.2)	36.7(4.8)
	LM-BFF	83.8(1.2)	80.3(2.3)	27.3(8.5)	16.8(3.7)	40.2(6.8)
	KPT	66.9(13.9)	75.7(17.1)	19.8(2.0)	16.6(2.2)	**56.1(6.4)**
	EDEntail-EK	87.5(2.8)	89.9(2.5)	43.3(9.3)	32.2(4.3)	43.4(7.6)
	- Ensemble	**89.2(2.5)^**	**90.4(2.1)^**	**43.6(1.3)^**	**33.1(4.1)^**	53.2(9.3)^
	EDEntail-FK	87.6(3.1)	88.5(1.5)	38.9(1.9)	25.2(4.2)	46.8(3.8)
	- Ensemble	89.0(3.4)^	89.6(1.9)^	43.3(1.7)^	28.1(3.2)^	51.4(8.6)^
K=16	Finetune	59.2(1.3)	62.9(1.4)	34.1(10.9)	27.0(11.4)	47.4(5.9)
	DualCL	65.9(2.7)	76.3(5.3)	25.8(3.2)	21.6(1.3)	20.1(0.9)
	EFL	55.0(3.5)	64.9(1.6)	45.0(13.3)	20.8(12.2)	62.2(1.6)
	Label-Entail	91.6(1.5)	90.7(0.7)	52.9(14.2)	32.0(8.5)	79.3(2.7)
	IDef-Entail	81.2(10.2)	90.1(0.9)	47.3(16.6)	41.1(9.5)	56.4(11.3)
	PET	91.9(0.8)	89.5(2.2)	54.2(2.8)	33.5(2.0)	80.3(2.8)
	WRAP	83.9(2.7)	88.8(3.2)	62.8(5.3)	35.9(6.5)	87.4(2.2)
	LM-BFF	**93.0(0.8)**	90.6(2.2)	64.0(4.6)	36.9(3.7)	89.0(3.5)
	KPT	87.6(6.9)	90.4(1.4)	56.5(4.0)	37.5(5.8)	88.6(2.9)
	EDEntail-EK	91.6(1.9)	90.9(0.6)	67.3(4.1)	43.1(2.4)	86.4(2.6)
	- Ensemble	92.5(2.0)^	**91.8(0.7)^**	**68.5(5.2)^**	**44.5(3.5)^**	89.8(2.2)^
	EDEntail-FK	92.0(0.9)	91.1(0.5)	60.6(3.7)	40.4(3.8)	89.1(2.1)
	- Ensemble	92.1(1.7)^	91.3(1.1)^	67.1(6.2)^	43.0(5.5)^	**89.9(2.5)^**
K=32	Finetune	88.0(1.6)	85.8(1.9)	60.0(8.9)	38.1(11.1)	81.0(6.8)
	DualCL	80.8(8.6)	88.9(1.1)	24.2(1.6)	20.9(1.8)	20.2(1.0)
	EFL	91.1(0.2)	91.7(0.4)	69.4(5.5)	41.3(3.3)	75.0(4.8)
	Label-Entail	92.2(0.2)	90.8(0.4)	64.4(3.8)	38.2(3.3)	84.0(5.6)
	IDef-Entail	89.7(1.7)	90.1(1.5)	41.2(24.1)	42.0(12.9)	70.5(11.2)
	PET	92.7(2.0)	90.7(2.2)	63.0(2.7)	42.2(2.2)	86.5(5.2)
	WRAP	92.2(0.9)	90.2(1.7)	66.6(5.0)	37.9(2.2)	86.8(5.3)
	LM-BFF	92.9(1.1)	92.0(0.7)	66.8(3.3)	42.3(4.1)	89.1(6.1)
	KPT	92.7(1.3)	91.5(1.0)	71.1(3.7)	39.9(2.7)	89.4(4.0)
	EDEntail-EK	93.4(0.3)	**93.5(0.4)**	72.3(1.5)	45.7(1.8)	90.9(1.5)
	- Ensemble	**94.7(0.6)^**	93.4(0.2)	73.9(4.3)^	51.7(3.4)^	**93.8(1.1)^**
	EDEntail-FK	94.2(0.1)	92.8(0.5)	72.3(1.2)	46.0(1.3)	90.7(1.9)
	- Ensemble	94.5(0.7)^	93.1(0.7)^	**75.3(3.8)^**	**53.5(3.1)^**	91.2(2.7)^

Table 2: Fewshot experimental results: We report the average accuracy of 5 runs under the best format with standard deviation in parentheses. EDEntail-EK and EDEntail-FK represent our approach under Entailment knowledge and Frequency knowledge, respectively. The best results are marked in 'bold'. Marker ^ signifies ensemble results enhanced our approach.

and Entailment knowledge, respectively under the setting of few-shot learning.

Our method outperforms fine-tuning and supervised deep learning model (DualCL) by a dramatic margin under all situations. Surprisingly, the deep learning network breaks down under the few-shot setting with multi-class classification. Compared with other classification methods, EDEntail performs well and achieves further improvement under the ensemble approach. It works well in multi-class datasets with a maximum of 7.9% (AMAN), 4.1% (MELD) accuracy improvement in 1-shot settings and 4.5% (AMAN), 3.4% (MELD) improvement in 16-shot settings. In 32-shot setting, EDEntail wins all models with improvements of 1.3%, 1.5%, 1.2%, 3.7% and 1.5% on SST-2, CR, AMAN, MELD, and TREC respectively. By inspecting the standard deviation, we can see our approach achieves improved robustness even in 1-shot settings, which is contrary to other compared entailment-based methods. Under ensemble settings, our approach achieves a stronger performance, especially in 32-shot settings, with 1.8%, 1.4%, 4.2%, 11.2% and 4.4% performance improvement on SST-2, CR, AMAN, MELD, and TREC respectively while the robustness is sacrificed a bit. From the table, it is evident that while state-of-the-art entailment-based models perform less effective than prompt-based models in most experimental settings, our novel entailment-based approach

shows a different story.

Significance Testing Regarding concerns about high standard deviation in the compared methods, Table 3 displays the p-values derived from comparing five accuracy results from our method and the SOTA model under the 1-shot, 16-shot, and 32-shot settings. Values highlighted in bold represent a statistically significant difference between the outcomes of the two models, with significance defined by a threshold of 0.1. This demonstrates the genuine effectiveness of our approach.

	SST2	CR	AMAN	MELD	TREC
1-shot	**0.01**	0.13	**3.66e-08**	**9.42e-05**	0.44
16-shot	0.22	**0.06**	**6.52e-03**	**0.067**	0.51
32-shot	**0.02**	**0.02**	**0.03**	**0.03**	**0.06**

Table 3: Significance testing between our method (EDEntail-EK-Ensemble) and the KPT model.

4.4 Few-shot Entailment Learning

The first task in our model is to entail each text with the hypothesis encoded with EDef. The performance at this level significantly influences the final classification results. Table 4 summarizes the performance of our model and the other three methods, EFL, Label-Entail and IDef-Entail. The results are obtained from the 32-shot experiments with the same experimental settings.

Model	SST2	CR	AMAN	MELD	TREC
EFL	92.2*	92.1*	80.4	51.4	80.8
IDef-Entail	89.0*	89.6*	69.0	66.0	87.1
Label-Entail	91.5*	90.3*	85.3	69.4	86.0
EDEntail-EK	**93.1***	**92.3***	86.2	**69.6**	**91.3**
EDEntail-FK	92.9*	91.2*	**87.8**	69.1	90.8

Table 4: Entailment experimental results: The marker * indicates the classification is a binary-class task as the entailment task.

When comparing the results in Table 2 and Table 4, we observe that for binary classification like SST-2 and CR, the results are closely aligned between the two tables. The classification performance is even better than the entailment performance, demonstrating that our classification loss function (see Eqn 3) effectively addresses the limitations with one entailment-based label removed. However, in multi-class classifications such as AMAN, MELD and TREC, the entailment performance significantly outperforms the classification performance due to the contradiction bias in the entailment task, wherein the number of contradiction samples in the text sets is $(|Label|-1)$ times higher than the number of entailment samples. While the bias is eliminated in binary-entailment tasks, it poses challenges in fine-grained tasks. Our method enhances feature learning by incorporating more label information into hypothesis. Compared with other entailment-based methods, it improves the alignment between entailment performance and classification performance.

4.5 Efficiency of Utilizing Limited Training Data

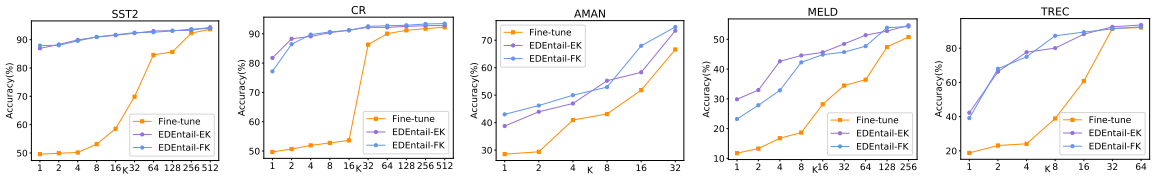
In few-shot learning, the model is expected to produce the best performance under limited training data. We conducted a comparison between the standard fine-tuning and our entailment-based approach under different numbers of samples selected as training data, ranging from 1 to the maximum number of samples that can be obtained for all classes in the available training set. The experimental setting aligns with the few-shot setting experiments. As shown in Figure 3, EDEntail consistently maintains a performance advantage over standard fine-tuning, particularly when the number of training samples is very small. In simple tasks like SST2 and CR, the performance saturates only with 32 examples, indicating the high efficiency of our proposed EDEntail in utilizing limited training data to achieve good performance.

4.6 Zero-shot Learning

The zero-shot setting experiments are conducted on the same test datasets as in experiments of the few-shot setting. We compare our method with fine-tuning, Label-Entail and IDef-Entail. We drop the Entailment knowledge since it originates from training sets, utilizing of which is unfair for zero-shot learning. Table 5 summarizes the experimental results. Obviously, zero-shot learning produces inferior performance to few-shot learning. Under the zero-shot setting, however, our method still surpasses the best performing label entailment method by 2.4%, 0.8%, 9.8%, 7.2%, and 10.9% in classification accuracy on the 5 datasets, respectively.

Model	SST2	CR	AMAN	MELD	TREC
Finetune	49.9	36.2	13.7	14.1	18.8
IDef-Entail	64.3	67.8	10.4	15.4	11.6
Label-Entail	86.3	89.2	33.0	28.3	24.3
EDEntail-FK	**88.7**	**90.0**	**42.8**	**35.5**	**35.2**

Table 5: Zeroshot experimental results.



(a) SST-2 (b) CR (c) AMAN (d) MELD (e) TREC

Figure 3: The efficiency of training sample scale comparison between standard fine-tuning and EDEntail in five evaluated datasets.

4.7 Analysis on Word Embedding Consistency

In the entailment-based model, learning a directional relation between premise and hypothesis is important (Mayer et al., 2023; Yang et al., 2023). When the training and testing samples exhibit higher feature similarity in keywords, there is a greater likelihood that the model will grasp a consistent relation during few-shot learning. Therefore, for the same label premises, high consistency of the embedding of keywords across different samples is beneficial for entailment-based feature learning.

To explore the effect of a single word versus extensional descriptive words used in the hypothesis on word embedding consistency, we conduct experiments on Bert[6] and Roberta[7]. The experiments evaluate 8 emotional classes that are usually studied in emotion recognition tasks. For each emotional class, we use ChatGPT to generate 30 sentences consisting of the class label itself and 3 non-emotional words ('I', 'the', and 'of'). The experiment of using a sequence of emotional words comprises two parts: 1). 6 synonyms, and 2). 6 antonyms.

We define inter-sentence same-word cosine similarity as the cosine similarity of the same word's embedding vector in different contexts. Our investigation focused on comparing inter-sentence same-word cosine similarity when a **single** extensional descriptive word is used in a hypothesis versus when a **sequence** of extensional descriptive words is used in constructing a hypothesis.

The results in Table 6 reveal that, in both language models, a higher inter-sentence same-word cosine similarity is achieved when a sequence of emotional words is used than when a single word or a sequence of opposite emotional words is used. Beyond the tabulated outcomes, added with a sequence of emotional words, the three non-emotional words in the same 30 sentences are found to have their inter-sentence same-word cosine similarity decrease or a comparatively lesser increase compared to the emotional words.

Addition Method	sadness	joy	anger	disgust	fear	surprised	shame	guilt
Bert								
Without	0.8829	0.8286	0.8127	0.7169	0.8507	0.8045	0.7873	0.8459
Single	0.8857	0.8407	0.8322	0.7966	0.8676	0.8169	0.7876	0.8449
Seqence (same emotion)	**0.9079**	**0.8752**	**0.8800**	**0.8376**	**0.8969**	**0.8526**	**0.8149**	**0.8554**
Seqence (opposite emotion)	0.8887	0.8640	0.8434	0.8191	0.8795	0.8392	0.8004	0.8471
Roberta								
Without	0.9933	0.9933	0.9876	0.9886	0.9924	0.9847	0.9858	0.9899
Single	0.9933	0.9933	0.9860	0.9887	0.9896	0.9855	0.9879	0.9918
Seqence (same emotion)	**0.9942**	**0.9941**	**0.9881**	**0.9922**	**0.9934**	**0.9912**	**0.9914**	**0.9954**
Seqence (oppsite emotion)	0.9919	0.9923	0.9869	0.9871	0.9822	0.9858	0.9835	0.9883

Table 6: Descriptive words addition experiments. The results are the cosine similarity between the same evaluated emotional words in 30 sentences under different addition methods.

From the above results, we draw the following conclusions:

1. The use of a sequence of extensional descriptive words results in improved inter-sentence same-word cosine similarity compared with the use of a single word such as a class label or its synonyms.
2. The use of a sequence of relevant descriptive words can generate higher inter-sentence same-word cosine similarity among other extensional descriptive words than when a sequence of irrelevant descriptive words.

Consequently, our suggested approach EDEntail, by incorporating an extensional definition, which comprises a sequence of extensional descriptive words, into the hypothesis, holds the potential to enhance the performance of few-shot text classification by improving keyword embedding consistency.

4.8 Comparison with Large Language Models

In light of the impressive performance and efficiency exhibited by large language models (LLMs), we conduct a comparative analysis of our method against three LLMs: GPT-3.5 (175B parameters) (Ouyang et al., 2022), Llama2 (Touvron et al.,

[6] https://huggingface.co/bert-base-uncased
[7] https://huggingface.co/roberta-large

2023) 7B and 13B, within a few-shot learning framework. Listed in Table 7, our model, EDEntail, with only 335 million parameters, outperforms larger models GPT-3.5, Llama2-13b, Llama2-7b. The results are the average of five tests using the same dataset as Table 2 but without entailment reformulations. The detailed model settings and used prompts for task classification are summarized in Appendix C.1.

	SST2	CR	AMAN	MELD	TREC
GPT-3.5	94.3	88.3	62.4	53.0	55.8
Llama2-13b	87.7	84.3	38.1	46.1	48.2
Llama2-7b	82.2	77.4	34.2	41.8	45.2
EDEntail	**94.7**	**93.5**	**75.3**	**53.5**	**93.8**

Table 7: Fewshot experimental results comparison with large language models. Our results are the best results reported from Table 2.

5 Conclusion

In this paper, we propose EDEntail, a novel entailment-based method with an extensional definition (EDef) for few-shot text classification. We leverage a number of extensional descriptive words encoded in the hypothesis to offer diverse label definitions, enhancing the feature similarity between train and test samples in few-shot entailment relation learning. A structured method is provided for the instruction of EDef generation and hypothesis construction. As a new method in providing label semantic information in hypothesis, extensive experiments show that EDEntail can achieve competitive classification performance with stronger robustness and sample efficiency.

6 Limitations

As discussed in Section 4.4, an entailment-based approach is not immune to potential biases arising from the uneven distribution of entailment and contradiction samples in multi-class test sets. To address this issue, we have implemented careful measures in our few-shot training method to achieve a balanced representation of both entailment and contradiction samples by ensuring that each label's extensional definition is represented as a contradiction sample at least once in train and valid set. However, further research is required to address the issue of sample number bias between entailment and contradiction in multi-class tasks.

Ethics Statement

This work introduces EDEntail, an entailment-based method for few-shot text classification. All experiments conducted in this study utilize publicly available datasets and codes. To facilitate future reproduction without unnecessary energy consumption, we will make our codes openly accessible.

Acknowledgements

The research was conducted at the Future Resilient Systems at the Singapore-ETH Centre, and is supported by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise programme.

References

Saima Aman and Stan Szpakowicz. 2008. Using roget's thesaurus for fine-grained emotion recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Angelo Basile, Guillermo Pérez-Torró, and Marc Franco-Salvador. 2021. Probabilistic ensembles of zero-and few-shot learning models for emotion classification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 128–137.

William F Battig and William E Montague. 1969. Category norms of verbal items in 56 categories a replication and extension of the connecticut category norms. *Journal of experimental Psychology*, 80(3p2):1.

Roy T Cook. 2009. Intensional definition. *A dictionary of philosophical logic*, page 155.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240.

Jesse Egbert and Brent Burch. 2023. Which words matter most? operationalizing lexical prevalence for rank-ordered word lists. *Applied Linguistics*, 44(1):103–126.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*.

Jiaxin Ge, Hongyin Luo, Yoon Kim, and James Glass. 2023. Entailment as robust self-learner. *arXiv preprint arXiv:2305.17197*.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1119.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. 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 4921–4933.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240.

Puneet Kumar and Balasubramanian Raman. 2022. A bert based dual-channel explainable text emotion recognition system. *Neural Networks*.

Dmitry Lamanov, Pavel Burnyshev, Katya Artemova, Valentin Malykh, Andrey Bout, and Irina Piontkovskaya. 2022. Template-based approach to zero-shot intent recognition. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 15–28.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tingting Ma, Jin-Ge Yao, Chin-Yew Lin, and Tiejun Zhao. 2021. Issues with entailment-based zero-shot text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 786–796.

Christian WF Mayer, Sabrina Ludwig, and Steffen Brandt. 2023. Prompt text classifications with transformer models! an exemplary introduction to prompt-based learning with large language models. *Journal of Research on Technology in Education*, 55(1):125–141.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Marc Pàmies, Joan Llop, Francesco Multari, Nicolau Duran-Silva, César Parra-Rojas, Aitor González-Agirre, Francesco Alessandro Massucci, and Marta Villegas. 2023. A weakly supervised textual entailment approach to zero-shot text classification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–296.

Flor Miriam Plaza-del Arco, María-Teresa Martín-Valdivia, and Roman Klinger. 2022. Natural language inference prompts for zero-shot emotion classification in text across corpora. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6805–6817.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Tim Schopf, Daniel Braun, and Florian Matthes. 2020. Semantic label representations with lbl2vec: A similarity-based approach for unsupervised text classification. In *International Conference on Web Information Systems and Technologies*, pages 59–73. Springer.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jianing Wang, Chengyu Wang, Fuli Luo, Chuanqi Tan, Minghui Qiu, Fei Yang, Qiuhui Shi, Songfang Huang, and Ming Gao. 2022a. Towards unified prompt tuning for few-shot text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 524–536.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*.

Yuqi Wang, Wei Wang, Qi Chen, Kaizhu Huang, Anh Nguyen, and Suparna De. 2022b. Generalised zero-shot learning for entailment-based text classification with external knowledge. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 19–25. IEEE.

Brent Woo. 2019. Innovation in functional categories: slash, a new coordinator in english. *English Language & Linguistics*, 23(3):621–644.

Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. 2022. Fastclass: A time-efficient approach to weakly-supervised text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4746–4758.

Zongbao Yang, Yinxin Xu, Jinlong Hu, and Shoubin Dong. 2023. Generating knowledge aware explanation for natural language inference. *Information Processing & Management*, 60(2):103245.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923.

Haoran Zhang, Aysa Xuemo Fan, and Rui Zhang. 2023a. Conentail: An entailment-based framework for universal zero and few shot classification with supervised contrastive pretraining. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1933–1945.

Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. Multi-task label embedding for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4545–4553.

Pengfei Zhang, Tingting Chai, and Yongdong Xu. 2023b. Adaptive prompt learning-based few-shot sentiment analysis. *Neural Processing Letters*, pages 1–14.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Zixiao Zhu and Kezhi Mao. 2023. Knowledge-based bert word embedding fine-tuning for emotion recognition. *Neurocomputing*, page 126488.

A Appendix A

The general prompts input for ChatGPT for vocabulary construction.

```
`Please generate <M> single words according to:
Express <`Label_1'> meanings, like <some related
words of `Label_1'>
do not have the meanings of <`Label_2'>, like <some
related words of `Label_2'>
. . .
do not have the meanings of <`Label_|Label|'>, like
<some related words of `Label_|Label|'>
<`Label_1' definition 1> given from Oxford English
dictionary
<`Label_1' definition 2> given from Oxford English
dictionary
. . .
<`Label_1' definition D> given from Oxford English
dictionary'
```

B Appendix B

Dataset Settings All datasets are split based on the N-way-K-shot training setting. Especially, as AMAN has no provided testing set, its testing sets are randomly selected with the size of 20% of the whole dataset and no overlapping of the corresponding training and validation sets five times for repeated experiments. The zero-shot learning experiments are implemented under the same dataset settings as the few-shot learning experiments.

In the construction of the training and validation sets, we maintain an equal balance between entailment and contradiction samples, ensuring $|E| : |C| = 1 : 1$. This approach aims for equitable representation in both categories. Furthermore, during the development of these sets, each label's extensional definition is represented at least once as a contradiction sample. This strategy is designed to enhance the learning process, facilitating a comprehensive understanding of every feature associated with the extensional definitions.

C Appendix C

C.1 Baseline Model Experimental Settings

Detailed information on baseline models and the corresponding re-run experimental settings for few-shot (1, 16, and 32) and zero-shot learning experiments.

Fine-tuning (FT) The traditional fine-tuning method inputs the hidden embedding of [CLS] into a pre-trained language model (PLM) to make predictions. In our re-run experiments, the PLM is

'Roberta-large'. The learning rate for all datasets is 1e-5.

DualCL A traditional supervised deep-learning model under BiLSTM-CNN dual-channel structure for text classification. In our re-run experiments, the PLM model is 'Roberta-large' with a learning rate of 1e-5. The other setting defaulted in [8].

EFL A few-shot learning method by reformulating all classification tasks as an entailment task. In the re-run experiment, the hypotheses for datasets "It was <LabelDef>" as mentioned in its paper. The extensional descriptive words in the hypothesis for SST-2 and CR are 'positive' and 'negative', for AMAN and MELD are 'joy', 'anger', 'sadness', 'surprise', 'disgust', 'others', and 'fear' and for TREC are 'expression', 'entity', 'description', 'human', 'location' and 'number'. The PLM model is 'Roberta-large'. To fair comparison, there is no data augmentation in EFL implementation.

Label-Entail The entailment-based method with label word in the hypothesis. Based on the suggested methodology, we refine the model using our few-shot training datasets and evaluate our test sets. In the re-run experiment, the hypotheses for datasets are the same as ours. The label words are the same as EFL. The PLM model is 'roberta-large-mnli'.

IDef-Entail The entailment-based method with intensional definition sourced from WordNet (Miller, 1995) in the hypothesis. Same as Label-Entail, we refine the model using our few-shot training datasets and evaluate our test sets. In the re-run experiment, the intensional definition is label WordNet definition. The PLM model is 'roberta-large-mnli'.

PET The basic prompt-tuning method uses the class name as the prompt word for each class. The prompt words in our re-run for SST-2 and CR are 'great' and 'terrible', for AMAN and MELD are 'joy', 'anger', 'sadness', 'surprise', 'disgust', 'others', and 'fear' and for TREC are ' Expression', ' Entity', ' Description', ' Human', ' Location' and ' Number'. In re-run experiments, the results are obtained from the prompts reported in the paper and other experiment settings are defaulted in [9].

WARP A prompt-based method by selecting the best prompt with training data in the continuous embedding space. The prompt tokens are trainable by the classification result. For re-run experiments, the manual verbalizer for SST-2 and CR is the same as IMDB given in [10]. The manual verbalizers for AMAN, MELD, and TREC are the same as ours. The initialization is the word embedding of the name of the class. The other experimental settings default in [10].

LM-BFF A prompt-based fine-tuning method with automatically generated prompts. This method follows in-context learning with training examples as demonstrations in the input context. For re-run experiments, the number of demonstration samples is one. The prompt is "It was [MASK]." The mask token mapping is the same setting as PET. Other experiment settings are defaulted in [11].

KPT A knowledgeable prompt-tuning method. KPT expands the label verbalizer with external knowledge bases to make the prediction mapping covers various perspectives of the label words. For re-run experiments, the prompts used in KPT are the same as our experiments. The knowledge verbalizer and prompts for SST-2 and CR are the same as IMDB given in [10]. The knowledge verbalizer and prompts for AMAN, MELD, and TREC are the same as our verbalizer and prompts. The other experimental settings default in [10].

GPT-3.5 We use 'gpt-3.5-turbo-16k' with 16,385 tokens available. Temperature sets to 0. The prompt for SST-2 and CR is 'Review:<example>', 'Sentiment Type:<label>', AMAN and MELD is 'Review:<example>', 'Emotion Type:<label>', and TREC is 'Question:<example>', 'Answer Type:<label>'. In 32-shot settings, all samples can be inputted in the prompt as <example> and <label> is the task-specific classification label.

Llama2 We use Llama2 7b[12] and 13b[13] with 4096 tokens available. In 32-shot settings, prompts exceeding model token limits were truncated. The prompt for task classification is the same as GPT-3.5.

C.2 EDEntail Experimental Settings

The reported results are the average five times randomly repeated performance. The non-ensembled result is the optimal accuracy among the evaluated

[8] https://github.com/hiyouga/Dual-Contrastive-Learning

[9] https://github.com/timoschick/pet

[10] https://github.com/thunlp/KnowledgeablePromptTuning

[11] https://github.com/princeton-nlp/LM-BFF

[12] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[13] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

hypotheses. The ensembled result uses all hypotheses in both training and testing as we introduced in section 3.3.

For baseline models' settings, all the compared models are under the same training and testing datasets as our model and reported the average accuracy under five repeated. For models with more than one prompt or hypothesis, the compared results are reported under the average accuracy across all prompts or hypotheses for a fair comparison.

For EDef length n (extensional definition boundary), if the size of the vocabulary for one label is smaller than 10, n is grid-searched from 2 to the largest number of the smallest vocabulary size. The experiments are run under a learning rate of 1e-5 with a batch size equal to 10 under the 1-shot setting, which appears to be effectively transferable to other few-shot scenarios, albeit with a minor decrease in performance. Regarding the time efficiency, the processing times for SST2, CR, AMAN, MELD, and TREC datasets were 44, 29, 46.5, 56.7, and 13.7 minutes respectively, which we believe is within an acceptable range.

The detailed information for the length and format of the EDef that we used in the few-shot (1, 16, and 32) learning experiment, few-shot learning (ensemble) experiment, and zero-shot learning experiment are summarized in Table 8 with descriptive words usage in Table 9. The pre-trained language model is 'roberta-large-mnli'[14]. All experiments are implemented under Python 3.7 environment and PyTorch 1.12.1. with Cuda version 11.3, GPU NVIDIA RTX A5000.

	Fewshot		Few-shot (ensemble)		Zeroshot
	Entailment Knoledge	Frequency Knowledge	Entailment Knoledge	Frequency Knowledge	Frequency Knowledge
SST-2	n=9,EDef-CL	n=8,EDef-CL	n=4,EDef-CA	n=7,EDef-CC	n=2,EDef-CC
CR	n=10,EDef-CL	n=9,EDef-CL	n=6,EDef-CA	n=7,EDef-CA	n=4,EDef-CC
AMAN	n=9,EDef-CC	n=7,EDef-CL	n=6,EDef-CC	n=6,EDef-CC	n=2,EDef-CA
MELD	n=3,EDef-CS	n=3,EDef-CC	n=5,EDef-CS	n=6,EDef-CC	n=2,EDef-CA
TREC	n=3,EDef-CA	n=4,EDef-CS	n=3,EDef-CC	n=5,EDef-CA	n=4,EDef-CS

Table 8: Experimental setting information on EDEntail, where n is the length of the EDef, and CC, CS, CL, and CA are the EDef formats we designed in the EDef Generation section.

[14] https://huggingface.co/roberta-large-mnli

		Entailment Knoledge		Frequency Knowledge
SST2	positive	n=4:proactive, constructive, relief, encouraging n=9: skilled, renaissance, gold, ecstatic, grateful, protection, freedom, pride, defeat	positive	n=2: encouraging, redemption n=7: gold, relief, pride, defeat, paradise, personalized, truthful n=8: gold, relief, pride, grateful, defeat, skilled, ecstatic, renaissance
	negative	n=4:attack, disappointed, needy, sneaky n=9: complain, impose, damage, opponent, attack, naughty, devastated, haunting, rage	negative	n=2: worst, sneaky n=7: attack, damage, racism, fearful, lied, sneaky, bashing n=8: attack, damage, racism, impose, opponent, fearful, lied, bashing
CR	positive	n=6:truthful, personalized, relief, pride, paradise, defeat n=10: clearly, vigilance, dazzling, protection, personalized, freedom, ecstatic, gold, pride, defeat	positive	n=4: relief, encouraging, constructive, proactive n=7: gold, relief, pride, defeat, paradise, personalized, truthful n=9: freedom, protection, gold, pride, grateful, defeat, skilled, ecstatic, renaissance
	negative	n=6:damage, broken, waning, attack, divisive, sneaky n=10: complain, impose, opponent, damage, fuck, disappointed, waning, attack, rage, hateful	negative	n=4: attack, disappointed, needy, sneaky n=7: attack, damage, racism, fearful, lied, sneaky, bashing n=9: attack, damage, rage, impose, opponent, complain, devastated, naughty, haunting
AMAN	joy	n=6:pleasure, cheer, triumphant, gratitude, enchantment, blessed n=9:pleasure, cheer, triumphant, joy, achievement, bliss, enchantment, ecstasy, blessed	joy	n=2:gratitude, triumphant n=6:pleasure, blessed, gratitude, cheer, triumphant, enchantment n=7:pleasure, achievement, blessed, gratitude, cheer, triumphant, enchantment
	anger	n=6:indignation, hostility, fury, outrage, aversion, provocation n=9:indignation, hostility, fury, animosity, outrage, provocation, resentment, hatred, malice	anger	n=2:hostility, ire n=6:fury, hostility, indignation, outrage, aversion, provocation n=7:hatred, hostility, indignation, annoyance, aversion, provocation, ire
	sadness	n=6:sadness, sadly, melancholy, dismal, grief, pathetic n=9:sadness, sadly, melancholy, dismal, grief, despair, pathetic, blues, depression	sadness	n=2:sadness, dismal n=6:grief, sadness, melancholy, sadly, pathetic, dismal n=7:grief, sadness, melancholy, sadly, pathetic, blues, dismal
	surprise	n=6:shocked, shock, awe, unbelievable, amazing, sudden n=9:astonishing, shock, curious, awe, unexpected, amazing, sudden, abrupt, breathtaking	surprise	n=2:unforeseen, breathtaking n=6:sudden, shock, amazing, shocked, awe, unbelievable n=7:sudden, shock, amazing, shocked, awe, abrupt, startling
	disgust	n=6:dislike, aversion, contempt, ugly, hateful, nausea n=9:dislike, disdain, aversion, contempt, ugly, hateful, disgusting, offensive, vomiting	disgust	n=2:offensive, intolerable n=6:ugly, contempt, dislike, nausea, aversion, hateful n=7:ugly, contempt, dislike, nausea, vomiting, aversion, hateful
	fear	n=6:fright, worry, panic, insecurity, terrifying, horror n=9:apprehension, scare, fright, worry, panic, anxiety, insecurity, terrifying, horror	fear	n=2:horror, shudder n=6:worry, horror, panic, terrifying, insecurity, fright n=7:worry, anxiety, horror, panic, terrifying, insecurity, fright
	neutral	others, no emotion	neutral	others, no emotion
MELD	joy	n=3:cheer, triumphant, bliss n=5:cheer, triumphant, achievement, pleasure, gratitude	joy	n=2:gratitude, triumphant n=3:cheer, bliss, triumphant n=6:pleasure, blessed, gratitude, cheer, triumphant, enchantment
	anger	n=3:discontent, outrage, hostility n=5:indignation, aversion, hostility, provocation, anger	anger	n=2:hostility, ire n=3:hostility, outrage, discontent n=6:fury, hostility, indignation, outrage, aversion, provocation
	sadness	n=3:grief, dismal, depression n=5:sadness, grief, blues, tragic, depression	sadness	n=2:sadness, dismal n=3:depression, grief, dismal n=6:grief, sadness, melancholy, sadly, pathetic, dismal
	surprise	n=3:shocked, unbelievable, unforeseen n=5:shocked, shock, awe, unbelievable, amazing	surprise	n=2:unforeseen, breathtaking n=3:shocked, unbelievable, unforeseen n=6:sudden, shock, amazing, shocked, awe, unbelievable
	disgust	n=3:displeasure, ugly, offensive n=5:aversion, ugly, bitter, nausea, hateful	disgust	n=2:offensive, intolerable n=3:ugly, offensive, displeasure n=6:ugly, contempt, dislike, nausea, aversion, hateful
	fear	n=3:suspense, horror, terrifying n=5:fright, panic, insecurity, horror, chilling	fear	n=2:horror, shudder n=3:horror, terrifying, suspense n=6:worry, horror, panic, terrifying, insecurity, fright
	neutral	others, no emotion	neutral	others, no emotion
TREC	entity	n=3:substance, event, body	entity	n=4:body, event, color, substance n=5:body, method, event, color, substance
	number	n=3:number, date, distance	number	n=4:number, percent, distance, date n=5:number, percent, distance, date, code
	description	n=3:reason, manner, description	description	n=4:reason, manner, definition, description n=5:reason, action, manner, definition, description
	human	n=3:title, group, organization	human	n=4:group, organization, persons, title n=5:individual, organization, persons, description, title
	location	n=3:location, city, state	location	n=4:state, country, location, mountain n=5:state, country, city, location, mountain
	abbreviation	abbreviation, expression abbreviated	abbreviation	abbreviation, expression abbreviated

Table 9: The extensional descriptive words in EDef that are used in the reported experimental results.