On the Effectiveness of Sentence Encoding for Intent Detection Meta-Learning

Tingting Ma[1]*, **Qianhui Wu**[2], **Zhiwei Yu**[2], **Tiejun Zhao**[1], **Chin-Yew Lin**[2]
[1]Harbin Institute of Technology, Harbin, China
[2]Microsoft Research Asia
hittingtingma@gmail.com
{qianhuiwu, zhiwyu, cyl}@microsoft.com, tjzhao@hit.edu.cn

Abstract

Recent studies on few-shot intent detection have attempted to formulate the task as a meta-learning problem, where a meta-learning model is trained with a certain capability to quickly adapt to newly specified few-shot tasks with potentially unseen intent categories. Prototypical networks have been commonly used in this setting, with the hope that good prototypical representations could be learned to capture the semantic similarity between the query and a few labeled instances. This intuition naturally leaves a question of whether or not a good sentence representation scheme could suffice for the task without further domain-specific adaptation. In this paper, we conduct empirical studies on a number of general-purpose sentence embedding schemes, showing that good sentence embeddings without any fine-tuning on intent detection data could produce a non-trivially strong performance. Inspired by the results from our qualitative analysis, we propose a frustratingly easy modification, which leads to consistent improvements over all sentence encoding schemes, including those from the state-of-the-art prototypical network variants with task-specific fine-tuning.[1]

1 Introduction

The task of *intent detection* aims at classifying user queries, typically in the form of short sentences, into intent categories. It has been widely adopted as one crucial component inside various applications such as dialogue systems, virtual assistants, and search engines. The domain-specific nature of those applications makes intent detection rather challenging because of the difficulty to acquire high-quality labeled data at scale. In particular, the sets of intents could vary a lot in different real-world scenarios. Such scenarios motivate the research of few-shot intent detection, which aims to classify utterances with new intent labels given very few labeled examples.

Recently, there exists a popular stream of research efforts (Yu et al., 2018; Geng et al., 2019; Nguyen et al., 2020; Dopierre et al., 2021a,b) that models the few-shot intent detection task as a *meta-learning* problem - a general machine learning paradigm which has already been successfully applied in other tasks of natural language processing (Han et al., 2018; Gu et al., 2018; Chen et al., 2019; Hou et al., 2020, *inter alia*). Under this formulation, the target is to train a good meta-learner that could be used to quickly adapt to any few-shot intent classification task with very few labeled examples. One of the most popular methods of meta-learning is the *prototypical network* (Snell et al., 2017), which learns an embedding of the input data, and then constructs a prototypical representation for every class via averaging over the input embeddings. Each query will be classified as the class with the minimum distance between the query embedding and the class prototype. Earlier empirical findings (Dopierre et al., 2021a) suggest that the prototypical networks could reach the state-of-the-art performance on most intent detection datasets when a text encoder based on fine-tuned BERT (Devlin et al., 2019) is used for sentence representation.

However, one notable challenge for prototypical networks, or basically most of the current meta-learning approaches is that the models could easily overfit the sparse and biased data distribution from merely a few training instances (Yang et al., 2021). Given that the goal of prototypical networks is essentially to learn proper encoding functions for nearest-neighbor classification, one natural question arises: *is it possible to utilize other general-purpose sentence representation schemes without fine-tuning on intent detection data*? In this way, the cost of collecting and fine-tuning on domain-specific labeled data might be mitigated,

*Work during internship at Microsoft Research Asia.
[1]Our code is available at https://github.com/microsoft/KC/tree/main/papers/IDML.

while reducing the risk of domain-specific overfitting.

In this paper, we conduct an empirical study to verify the utility and the effectiveness of recent popular sentence encoding schemes for intent detection meta-learning. Specifically, we make the following contributions:

- We empirically compare a number of popular sentence embedding methods on various intent detection benchmarks and observe non-trivial strong performance in the meta-learning setup for few-shot intent detection.
- We quantitatively verify the better capability for cross-dataset generalization from general-purpose sentence encoders, and conduct qualitative analysis on the behaviors of different sentence encoding schemes.
- Based on our analysis, we propose a frustratingly simple modification to utilize the label name information, with the hope to yield sentence representations more targeted at the intent detection task. Follow-up experiments show consistent and substantial improvements over all sentence encoding methods, making them stronger baselines for the task, while the modification could also improve the state-of-the-art system performance.

2 Preliminary

2.1 The meta-learning setup

The meta-learning setting aims at learning a good "meta-learner" that could quickly adapt to newly specified classification tasks with few labeled examples available. A few-shot task, called an *episode*, is denoted as $\mathcal{T} = (\mathcal{S}, \mathcal{Q}, \mathcal{Y})$, usually following an N-way K-shot setting: given a *support set* $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N \times K}$ containing a small number of K labeled examples for each class $c \in \mathcal{Y}$ ($|\mathcal{Y}| = N$), the model is expected to assign a label from $\mathcal{Y}$ to each instance in the *query set* $\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{N \times K}$. At the **meta-training** phase, the meta-learner is trained from a series of episodes sampled from training data with classes $\mathcal{Y}_{train}$, via updating based on the prediction loss on $\mathcal{Q}$. At the **meta-testing** phase, the meta-learner is evaluated on many episodes $\mathcal{T}' = (\mathcal{S}', \mathcal{Q}', \mathcal{Y}')$ constructed from test data, with each $\mathcal{Y}' \subset \mathcal{Y}_{test}$ from a non-overlapping label space to verify the meta-learning capability, i.e., $\mathcal{Y}_{test} \cap \mathcal{Y}_{train} = \emptyset$.

2.2 Prototypical networks

Formally, a prototypical network (Snell et al., 2017, ProtoNet) learns an encoding function E_ϕ (parameterized by ϕ) to embed a sentence x_i into a vector $E_\phi(x_i)$. The class prototype e_c for each class c is obtained by taking the average embedding of sentences with the label c in the support set $\mathcal{S}$:

$$e_c = \frac{1}{K} \sum_{(x_i, y_i) \in \mathcal{S}:\, y_i = c} E_\phi(x_i).$$

With these prototype vectors, the predicted class distribution in the label space $\mathcal{Y}$ for a query x is calculated by

$$p(y = c|x) = \frac{\exp(-d(E_\phi(x), e_c))}{\sum_{c \in \mathcal{Y}} \exp(-d(E_\phi(x), e_c))},$$

where d is a distance metric, usually set to be the Euclidean distance. The encoder is trained by minimizing the cross-entropy loss on the query set of the episodes from $\mathcal{Y}_{train}$.

3 Our Empirical Study

3.1 Adapting generic sentence embedding

The main goal of this work is to explore the effectiveness of general-purpose sentence embedding methods without fine-tuning on intent data. A high-quality sentence embedding could be used to identify which instance in the few labeled examples is semantically close to an input query and henceforth expressing the same intent. This intuition makes it natural to directly adapt sentence encoding to ProtoNet. Specifically, for whatever pre-trained encoder E_s to produce sentence embedding, we replace the encoder E_ϕ with E_s in the ProtoNet. Note that we take a pre-trained E_s as-is, in other words, there is **no meta-training phase**.

We experiment with a number of modern popular sentence embedding methods, covering sentence embeddings pre-trained from either large-scale unlabeled text data or with supervision from additional sentence pairs. Specifically, the following four typical methods yielding five specific model instances in total are used in our experiments:

Sentence-BERT *Sentence-BERT* (Reimers and Gurevych, 2019) takes BERT / RoBERTa (Liu et al., 2019b) as the basic encoder and uses Siamese and triplet network (Schroff et al., 2015) structure to derive sentence embeddings by comparing similarities between sentence pairs. Here we consider

two model instances pre-trained on different data: i) *SBERT-paraphrase*, a DistillRoBERTa (Sanh et al., 2019) based model trained on a broad range of paraphrase corpora;[2] ii) *SBERT-NLI*, a RoBERTa based model trained on SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018), using a three-way classification objective to predict the relationship of a pair of sentences, *i.e.*, `entailment`, `neural`, or `contradiction`. Both model instances utilize mean pooling over token representations in a sentence for sentence representation.

SimCSE *SimCSE* (Gao et al., 2021) learns sentence embeddings by contrastive learning. Specifically, it encodes a sentence with the RoBERTa model and takes the representation of the `[CLS]` token as the sentence representation. Given an anchor sentence, the model is trained to predict the "positive" example, *i.e.*, the most semantic similar example, among the "negatives". Here we consider the situation that all anchors and their positive as well as negative examples are constructed from SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018), denoted as *SimCSE-NLI*.

DeCLUTR *DeCLUTR* (Giorgi et al., 2021) is an unsupervised sentence embedding method trained on documents from OpenWebText (Gokaslan and Cohen, 2019) and a subset of WebText (Radford et al., 2019). The mean pooling of contextual word representations obtained from RoBERTa is used as the sentence embedding. The sentence encoder is trained with a self-supervised contrastive loss to minimize the distance between the embeddings of textual segments randomly sampled from nearby in the same document.

SP-paraphrase Different from afore-mentioned models that are built upon large pre-trained language models, Wieting et al. (2019, 2021) propose a lightweight paraphrastic sentence embedding method, denoted as *SP-paraphrase*. *SP-paraphrase* first tokenizes a sentence into subwords with SentencePiece (Kudo and Richardson, 2018), then learns context-independent embeddings for sub-word tokens within a pre-defined vocabulary, and finally averages over all sub-word embeddings of a sentence for the sentence embedding. This model is trained on ParaNMT (Wieting and Gimpel, 2018) with a margin loss and carefully selected negative examples.

[2] `https://www.sbert.net/examples/training/paraphrases/README.html`

3.2 Meta-learning methods for reference

To study the different behaviors between general sentence encoding and meta-learning algorithms trained on intent datasets, we also compare with the representative and the start-of-art meta-learning algorithms as following:

ProtoNet As described before.

ProtoNet+MLM The unlabeled data of the target dataset is used for finetuning the pretrained language model with the masked language modeling (MLM) objective. This step is intentionally serving as a kind of domain adaptation, leading to a finetuned encoder to be used as the initial base encoder of ProtoNet (Dopierre et al., 2021a).

ProtAugment The current start-of-the-art framework in intent detection meta-learning proposed by Dopierre et al. (2021b). A paraphrasing model is trained to produce multiple diverse paraphrases for an unlabeled sentence from the training, development, or testing instance. Based on ProtoNet+MLM, the prototypical network is trained with an additional consistency loss to make the embedding of a sentence to be closer to the unlabeled prototype embedding of its paraphrases, and more distant away from other unlabeled prototypes.

3.3 Datasets

We evaluate these methods on four datasets for a comprehensive analysis and fair comparison with Dopierre et al. (2021b).

Banking77 Banking77 (Casanueva et al., 2020) is a single-domain dataset that contains 77 fine-grained intents about banking. It consists of 13,083 customer service queries and many of the intents are partially overlapped (e.g. verify top up, top up limits, pending top up). 25 intent classes are used for training, 25 for development and the remaining classes are for testing.

HWU64 HWU64 (Liu et al., 2019a) contains 11,036 user-generated utterances about home robots covering 64 intents from 21 different domains such as alarm and calendar. This dataset is class-balanced and each intent has 190 instances. Intents are split into train, dev, and test by domains to minimize the label semantic sharing amongst splits.

Liu57 This is also a multi-domain intent dataset from Liu et al. (2019a) which contains 25,478 user

Method	Base Model	Banking77		HWU64		Liu57		Clinic150	
		K=1	K=5	K=1	K=5	K=1	K=5	K=1	K=5
Meta-learning models trained on intent dataset:									
ProtoNet	RoBERTa-base	86.98±1.09	94.37±0.39	79.23±2.35	91.77±1.12	82.51±1.99	93.16±0.97	94.61±0.74	98.55±0.26
ProtoNet+MLM	RoBERTa-base	89.38±1.42	95.84±0.52	85.11±1.42	93.47±0.83	87.33±2.61	95.16±0.80	96.89±0.28	98.89±0.26
ProtAugment	RoBERTa-base	90.34±0.99	96.28±0.49	85.61±1.37	93.88±0.76	88.08±1.70	95.33±0.64	97.26±0.28	99.10±0.26
ProtAugment*	BERT-base-cased	89.56	94.71	84.34	92.55	86.11	93.70	96.49	98.74
Pre-trained general sentence encodings:									
SBERT-paraphrase	DistilRoBERTa	**81.32**±1.43	**94.35**±0.55	77.01±1.24	91.73±1.22	77.90±1.46	**93.84**±0.97	90.87±0.81	**98.55**±0.18
SBERT-NLI	RoBERTa-base	78.97±1.33	93.69±0.58	**78.18**±1.41	**92.31**±1.01	**79.45**±2.49	93.81±1.07	**91.24**±0.65	**98.55**±0.29
SimCSE-NLI	RoBERTa-base	78.62±0.91	93.44±0.68	77.37±1.86	92.00±0.95	78.65±2.43	93.73±1.22	90.95±0.70	**98.54**±0.24
DeCLUTR	RoBERTa-base	71.75±1.29	91.26±0.90	71.13±2.85	90.33±1.02	71.07±2.63	91.93±1.17	85.71±1.13	98.32±0.30
SP-paraphrase	Sentence-Piece	78.44±1.47	92.81±0.53	73.45±2.03	89.00±1.19	74.80±1.87	89.50±1.49	86.11±2.26	96.68±0.68

Table 1: Performance comparison under the 5-way K-shot settings. ProtAugment* denotes results reported in Dopierre et al. (2021b) with BERT-base-cased model. The highest accuracies of meta-learning models are underlined while those of general sentence embedding methods are **bolded**.

utterances about home robots with a total number of 54 intent classes. The class distribution is highly imbalanced, with the most frequent intent (`query`) accounts for 23% of all utterances while the least frequent intent (`volume_other`) only occupies 0.09%. Each of the train, dev, and test splits has 18 classes.

Clinc150 Larson et al. (2019) introduce a crowd-sourced dataset containing 22,500 user queries covering 150 different intents in ten general domains. We randomly select 50 classes for training, development, and testing, respectively.

3.4 Evaluation

We evaluate all methods on the standard 5-way 1-shot and 5-way 5-shot settings as in Dopierre et al. (2021b). We compute the averaged accuracy on 600 episodes and there are five query examples in each episode. To reduce the performance variation, we run all experiments five times with five different class splits and report the averaged accuracy.

4 Results and Analysis

Table 1 shows the performance of all approaches on four benchmarks. We can see that **general sentence embeddings without any task-specific fine-tuning achieve non-trivial performance**. Compared with meta-learning models elaborately designed for few-shot learning, general sentence embedding can reach a rather strong performance on all datasets in the 5-way 5-shot setting, even outperform or on par with the representative ProtoNet on HWU64, Liu57, and Clinic150. For the 5-way 1-shot setting, general sentence embeddings yield less satisfactory results than meta-learning. We suspect that the finetuned meta-learning models could be less sensitive to the distribution of support examples, especially in the 1-shot setting.

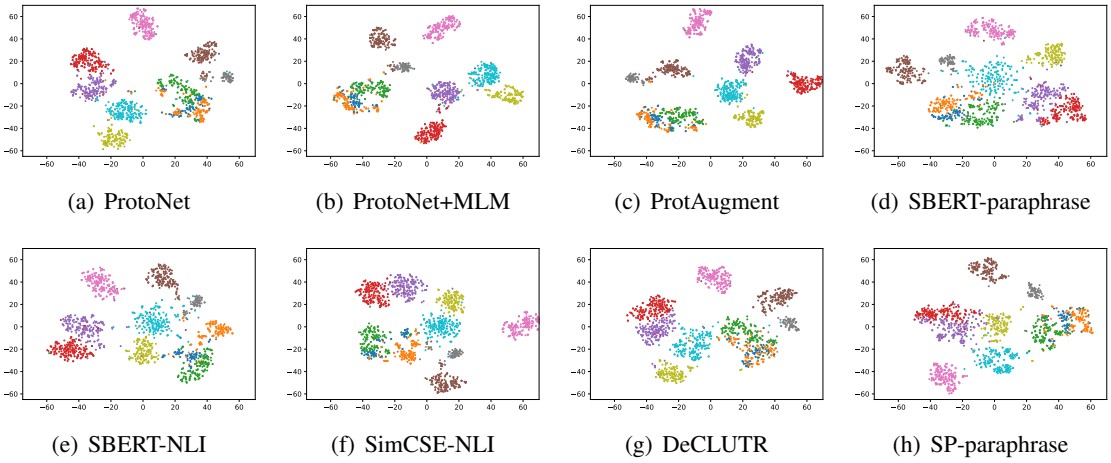
4.1 Sentence embedding visualization

Here we try to obtain a more intuitive understanding of the sentence embeddings obtained from different methods via low-dimensional projection using t-SNE (van der Maaten and Hinton, 2008). For meta-learning based models, we use the sentence encoders trained on the HUW64 training split[3] in the 5-way 1-shot setting. Figure 1 shows the projection results of sentences from 10 randomly selected classes in the HWU64 test split.

1) Meta-learning based methods generally produce sentence embeddings with larger between-class distances and smaller within-class distances on most classes. Compared with general pre-trained sentence embeddings, embeddings of different classes derived from meta-learning methods are more compact and distinguishable, which enables these meta-learning based models to achieve more robust test performance when given different support examples for similarity comparison, especially in the 1-shot setting.

2) Meta-learning based methods are good at handling test classes that share similar patterns with training classes. As shown in Table 2, when only testing on the class `remove_calendar` and class `query_calendar`, all meta-learning models significantly outperform sentence embeddings since two similar classes `query_alarm` and `remove_alarm` exist in the training data.

[3]**Training intent classes**: query_alarm, remove_alarm, set_alarm, email_addcontact, music_likeness, query_music, music_settings, query_news, query_transport, transport_taxi, transport_ticket, transport_traffic, query_weather, query_email, query_emailcontact, send_email.



(a) ProtoNet (b) ProtoNet+MLM (c) ProtAugment (d) SBERT-paraphrase

(e) SBERT-NLI (f) SimCSE-NLI (g) DeCLUTR (h) SP-paraphrase

Figure 1: T-SNE visualization of different sentence representations from 10 randomly selected classes in the HWU64 test split. Intent classes: query_calendar, remove_calendar, volume_mute, volume_up, volume_down, QA_definition, QA_math, iot_wemo_on, iot_coffee, iot_cleaning.

Type Set	Meta-Learning Models trained on HWU64			General Sentence Embeddings					ProtAugment+SentEmb	
	ProtoNet	ProtoNet+MLM	ProtAugment	SBERT-para.	SBERT-NLI	SimCSE-NLI	DeCLUTR	SP-para.	PA-SBERT-NLI	PA-SimCSE-NLI
`Calendar/*`	83.77	**95.90**	95.18	76.55	78.77	77.88	72.12	68.68	-	-
`Volume/*`	47.43	50.73	48.46	51.73	**62.19**	60.32	45.42	57.60	55.09↓	53.09↓

Table 2: Accuracy on particular type sets from HWU64 test classes under 1-shot setting. `Volume/*` denotes {`volume_up`, `volume_down`, `volume_mute`}. while `Calendar/*` denotes {`query_calendar`, `remove_calendar`}. PA- denotes applying the meta-learning procedure as in *ProtAugment*.

3) General sentence embeddings may achieve superior performance than meta-learning based models on target tasks that require fine-grained information but share little knowledge with the training tasks. In Figure 1, all methods seem to struggle in differentiating the fine-grained intent classes `volume_up`, `volume_down`, and `volume_mute`, which are very dissimilar to classes in training tasks. To quantify the ability of different methods to discriminate them , we evaluate all methods only on these three classes in the 1-shot setting. Surprisingly, as shown in Table 2, meta-learning models lag behind almost all general sentence embedding methods. To investigate whether this inferiority is caused by overfitting, we take *SBERT-NLI* and *SimCSE-NLI*[4] as base encoders of *ProtAugment*, denoted as *PA-SBERT-NLI* and *PA-SimCSE-NLI*. Table 2 indicates that compared *SBERT-NLI* and *SimCSE-NLI*, finetuning will lead to a performance drop of 7.10 points and 7.23 points respectively, which verifies that meta-learning based models may have a bias towards categorizing intent classes similar to training tasks, which, on the debit side, restraints their capabilities on distinguishing fine-grained test classes that share little knowledge with training classes.

Model	Source dataset	5-way 1-shot	5-way 5-shot
ProtAugment	HWU64	73.20±2.40	90.81±1.02
ProtAugment	Liu57	71.70±3.92	90.16±1.35
ProtAugment	Clinic150	79.35±1.49	92.86±0.57
SBERT-para.	paraphrase data	**81.32**±1.43	**94.35**±0.55
SBERT-NLI	NLI data	78.97±1.33	93.69±0.58
SimCSE-NLI	NLI data	78.62±0.91	93.44±0.68
DeCLUTR	unlabeled data	71.75±1.29	91.26±0.90
SP-para.	ParaNMT	78.44±1.47	92.81±0.53
PA-SBERT-para.	HWU64	80.97±1.67↓	93.72±0.68↓
PA-SBERT-para.	Liu57	79.56±2.52↓	93.96±0.87↓
PA-SBERT-para.	Clinic150	82.77±1.39↑	94.75±0.54↑

Table 3: Accuracy on Banking77 dataset while the models/sentence encodings are trained on source datasets. **PA-SBERT-para.** denotes training ProtAugment on intent data by taking SBERT-para. as initial encoder.

4.2 Cross-dataset generalization

To further verify our hypothesis that meta-learning methods could fail when the target tasks require fine-grained information but share little knowledge with the training tasks, we test models on a more challenging setting: directly transfer ProtAugment trained on other datasets to the single-domain Bank-

[4]We choose these two models because they achieve best performance in Table 2.

Query:	[create_or_add_lists] add new item to list
Support #1:	[create_or_add_lists] add to my groceries
Support #2:	[remove_lists] delete list
Predict:	ProtAugment: [create_or_add_lists] GeneralSentEmb: [remove_lists]
Query:	[receiving_money] How can my friend transfer money to me?
Support #1:	[receiving_money] How can my friend pay me?
Support #2:	[beneficiary_not_allowed] I tried to transfer cryptocurrency into my account but was denied
Predict:	ProtAugment: [beneficiary_not_allowed] GeneralSentEmb: [receiving_money]
Query:	[meaning_of_life] would you let me know what the meaning of life is
Support #1:	[meaning_of_life] is there a reason people exist
Support #2:	[are_you_a_robot] can you tell me know what kind of life form you are
Predict:	ProtAugment: [are_you_a_robot] GeneralSentEmb: [are_you_a_robot]

Table 4: Case study under 1-shot setting. Green color means the method predicts correctly while red color means wrong prediction. **GeneralSentEmb** means all general sentence embedding methods.

ing77 dataset.[5] From Table 3, we find:

1) General sentence embeddings have better cross-dataset generalization performance. ProtAugment, which shows better in-domain accuracy, lags behind most general sentence embeddings under the challenging cross-dataset generalization test. ProtAugment trained on HWU64 and Liu57 performs the worst, even underperforms the sentence embedding baselines since their intents are about home robot, which is obviously different from Banking77. ProtAugment trained on Clinic150 achieves better performance since Clinic150 has several intents[6] from bank domain, but still inferior to SBERT-paraphrase on 5-way 1-shot setting, SBERT-paraphrase, SBERT-NLI, SimCSE-NLI on 5-way 5-shot setting. This indicates the meta-learning methods could overfit the task distribution from training datasets. Benefiting from the supervision signal and diverse data distribution coverage provided by labeled sentence pairs (paraphrase, or NLI), such tasks guided the general sentence encoding to encode more fine-grained information which might relevant to target intent labels in any granularity, and moreover, reducing the risk of overfitting to a small intent dataset.

2) Finetuning the strongest general sentence embedding by meta-learning algorithm on intent datasets struggles to bring significant improvement compared with raw sentence embedding. *PA-SBERT-para.* even drops performance compared to *SBERT-para.* when finetuning on HWU64 and Liu57 datasets. This indicates that finetuning sentence embedding on a small intent corpus dissimilar to test data may cause overfitting and harm the cross-dataset generalization. When finetuning on a more similar corpus Clinic 150, the limited improvement questions the necessity of current meta-learning algorithms in practical large-domain gap scenarios when a high-quality general sentence embedding is available.

4.3 Case study

We conduct qualitative analysis to get a better understanding of the behaviors from different methods. A few examples have been listed in Table 4.

1) General sentence encoding captures semantic relatedness instead of pure intent similarity between query and support, which may sometimes be misleading. For the first example, all general sentence embedding methods fail due to the sharing part "list" between the query example and the support example of the wrong label. This indicates that the sentence embeddings actually capture the relatedness between two sentences instead of intents. So the general sentence embeddings tend to be misled by irrelevant parts in the sentences which do not convey the real intent. However, ProtAugment can focus on the key parts such as verb phrases for identifying intents by domain specific finetuning.

2) Patterns learned by meta-learning models could overfit and fail in cross-dataset settings. The second example from Banking77 shows an interesting case where ProtAugment trained on HWU64 fails in the cross-dataset scenario. The support example of the category `receiving_money` and the query are semantically close, therefore no surprise for our sentence embedding baselines to get it correct. However, ProtAugment makes the wrong prediction by incorrectly focusing on the same verb "transfer" between

[5] We choose Banking77 as test dataset since it contains the most fine-grained intents (bank domain) and is significantly different from other datasets. HWU64, Clinic150, and Liu57 share some intents, like "recipe", "query weather", "traffic".

[6] such as "transaction", "exchange_rate", "credit_limit"

Method	Banking77		HWU64		Liu57		Clinic150	
	K=1	K=5	K=1	K=5	K=1	K=5	K=1	K=5
L-ProtoNet+MLM	94.00 (+4.62)	95.97 (+0.13)	91.30 (+6.19)	94.09 (+0.62)	92.00 (+4.67)	95.26 (+0.10)	98.36 (+1.47)	99.08 (+0.19)
L-ProtAugment	93.42 (+3.08)	96.11 (-0.17)	91.73 (+6.12)	94.15 (+0.27)	92.79 (+4.71)	95.34 (+0.01)	98.43 (+1.17)	99.19 (+0.09)
L-SBERT-paraphrase	88.94 (+7.62)	**94.53** (+0.18)	87.09 (+10.08)	92.43 (+0.70)	88.03 (+10.40)	94.11 (+0.27)	**96.49** (+5.62)	98.67 (+0.12)
L-SBERT-NLI	88.30 (+9.33)	94.06 (+0.37)	**87.28** (+9.10)	**92.90** (+0.59)	88.31 (+8.86)	94.06 (+0.25)	96.11 (+4.87)	98.70 (+0.15)
L-SimCSE-NLI	88.19 (+9.57)	93.44 (0.00)	87.14 (+9.77)	92.70 (+0.70)	**89.09** (+10.44)	**94.29** (+0.56)	96.35 (+5.40)	**98.74** (+0.20)
L-DeCLUTR	82.33 (+10.58)	91.91 (+0.65)	80.95 (+9.82)	91.22 (+0.89)	80.61 (+9.54)	92.40 (+0.47)	93.05 (+7.34)	98.50 (+0.18)
L-SP-paraphrase	**89.00** (+10.56)	93.29 (+0.48)	83.20 (+9.75)	90.19 (+1.19)	82.09 (+7.28)	89.90 (+0.40)	91.81 (+5.70)	96.49 (-0.19)

Table 5: Accuracy with improvement value obtained after adding label name under 5-way 1-shot setting and 5-way 5-shot setting. We denote all methods adding the label names in the support set with a "**L-**" prefix. The absolute improvement accuracy is shown in (parentheses) compared the baselines without label names.

the query and support example for the wrong label, while ignoring the overall semantic meaning in the sentence. Such shortcut leads to better accuracy on the HWU64 dataset since most of its intents contain only verbs and object nouns, however it could lead to failure in Banking77 since intents in Banking77 need capture more fine-grained semantic information.

3) Uninformative support examples make all methods fail. For the last instance, the support example for `meaning_of_life` (asking what is the meaning of life) is not a usual expression for that intent, and all methods fail by predicting a label with more content similarity.

5 An Easy Modification

5.1 Label names as support

Inspired by previous analysis revealing the need for sentence encoders to capture more of the key phrases, we propose the following frustratingly simple modification: **adding the label names as instances into the support set**. Denoting the label name of an intent category c as l_c, then the prototype of class c after adding the label name becomes

$$e_c = \frac{1}{K+1}(E_s(l_c) + \sum_{(x_i,y_i)\in\mathcal{S}:y_i=c} E_s(x_i)).$$

Note that the label names are *always available for free* at both meta-training and meta-testing phases. The label name can be seen as a **discriminative** and **representative** example for the corresponding intent. It is **discriminative** in the sense that adding label names to the support set is equivalent to putting more weights on words similar to the intent phrase when calculating the prototypes since words in the intent label usually are key words. For the first example in Table 4, adding the label name `create_or_add_lists` and `remove_lists` could make the model paying more attention to discriminate the act of "add" and "delete". The label name is also **representative** in that it sometimes could directly convey a relatively abstract concept. For the third example in Table 4, the label name `meaning_of_life` is more informative than the support set examples.

5.2 Results and discussion

From Table 5, we can observe that the label name support has consistently and substantially improved all methods. Specifically, in the 5-way 1-shot setting, the results improve about 3% to 11% on Banking77, 6% to 10% on HWU64, 1% to 8% on Clinc150. Moreover, adding the label name as support could also improve the state-of-the-art ProtAugment framework, as shown in the results of L-ProtoNet+MLM and L-ProtAugment.

Meaningful labels improve baselines. As shown in Table 6, for the first example, adding label names correct the prediction of sentence embedding baselines. Adding `share_location` as a support example, the prototype for this class may contain more information about "know" and "location" compared to the irrelevant word "miranda". This illustrates the discriminative effectiveness of label names. For the second example, all methods incorrectly predict `remove_lists` because the bad example given in the support set for the label `create_or_add_lists`. Adding the label names corrects all methods with better representation for the class `create_or_add_lists`.

Limitation: negative effect from misleading or vague labels. We also observe slightly negative effects in some cases. For example, when adding the label name `general_negate` (this intent means a person does not agree with something) for the third example in Table 6, "negate" is not a usual expression for the intent `general_negate`, and the association between "not" in the query and

Episode		ProtAugment		SBERT-Para.	
		w/o L	*w/* L	*w/o* L	*w/* L
Query:	`[spelling]` i need to know how to spell "miranda"				
Support #1:	`[spelling]` i can't figure out how to spell superficial	✓	✓	✗	✓
Support #2:	`[share_location]` have miranda know about my current location				
Query:	`[create_or_add_lists]` i need to create a new to do list				
Support #1:	`[create_or_add_lists]` we need milk	✗	✓	✗	✓
Support #2:	`[remove_lists]` delete butterfly clips from my wish list				
Query:	`[general_negate]` sorry but that is not the right answer.				
Support #1:	`[general_negate]` please rectify the command.	✓	✗	✓	✗
Support #2:	`[general_don't_care]` any one would be okay with me, olly.				

Table 6: Case study on HWU64, Clinic150 datasets after adding label names. `[Intent]`: the true intent label; ✓: correct predictions; ✗: wrong predictions; *w/o* L: without label name; *w/* L: after adding label name.

the word "don't" in the label name `general don't_care` makes the model prefer this wrong intent. We also find some label names in the Liu57 dataset to be rather vague or ambiguous, making them difficult to bring any useful information. For example, the label name `likeness` actually corresponds to utterances expressing the likeness to music, while the label name `music` means listening to or playing music. Adding these label names usually leads to confusion between these two intents.

6 Related Work

6.1 Few-Shot Intent Detection

Studies on few-shot intent detection usually focus on two settings: (1) only a handful of annotated examples for each intent are available during training (Casanueva et al., 2020; Mehri and Eric, 2021; Zhang et al., 2020, 2021b; Qu et al., 2021); (2) in addition to the few-shot examples of target intents, rich labeled examples of other intents are available for training (Xia et al., 2021, 2020; Nguyen et al., 2020; Li and Zhang, 2021; Dopierre et al., 2021b; Yu et al., 2021; Zhang et al., 2021a). In this paper, we focus on the second setting, where the problem is typically formulated as the *meta-learning* problem and various approaches have been proposed (Yu et al., 2018; Geng et al., 2019; Nguyen et al., 2020; Li and Zhang, 2021; Dopierre et al., 2021b). Dopierre et al. (2021b) propose to use diverse paraphrases to improve the ProtoNet and achieve the SoTA performance in the semi-supervised intent detection meta-learning setting. Zhang et al. (2021a) study the effectiveness of pre-training with labeled intent data for this problem. Instead of developing a new framework or a new algorithm, this work conducts an empirical study on the effectiveness of using general pre-trained sentence encodings for this task.

6.2 Label Semantics for Low-Resource Text Classification

There also exists more complex usage of label names in the related literature on zero-shot or few-shot text classification tasks (Yazdani and Henderson, 2015; Chen et al., 2016; Wang et al., 2018; Yan et al., 2020; Luo et al., 2021; Hou et al., 2021), typically involving learnable label embeddings with crafted encoder modules or a more clever usage of pre-trained language models.

In zero-shot text classification, Chang et al. (2008) embed label descriptions and texts into a shared Wikipedia concept space and then measure their similarities for classification. Similarly, Chen et al. (2016) jointly embed label names and utterances into one distributed embedding space. Yazdani and Henderson (2015) leverage the structure of intent labels to produce a classification hyperplane for zero-shot intent classification.

For few-shot text classification, prompt-based methods (Schick and Schütze, 2021a,b) use label names to construct verbalizers to map each label into cloze-style phrases, which enables the utilization of powerful pretrained language models. More recently, Müller et al. (2022) propose label tuning which only finetunes the label embeddings but freezes the sentence encoder for few-shot text classification. Mueller et al. (2022) explore label semantics by performing pre-training on a mix of gold and weakly annotated sentence-label pairs. Another line of works (Luo et al., 2021; Hou et al., 2021) incorporate label names into the meta-learning models. Luo et al. (2021) append label names to the utterances to help the model extract discriminative features. Hou et al. (2021) take a linear combination of the prototype calculated

by support examples and the label embedding as the anchored label representation to separate different labels in a multi-label intent classification scenario. Different from previous works, our simple modification aims to enhance general sentence embeddings towards key parts that express an intent. Besides, the proposed modification doesn't introduce any parameters, so it can be adapted to any pretrained sentence encoders with ease.

7 Conclusion and Future Work

Motivated by the nature of prototypical networks for intent detection meta-learning, in this paper, we empirically compare some modern popular sentence encoders on multiple intent detection benchmarks, observing non-trivially strong performance with better cross-dataset generalization capability than the fine-tuned sentence encoders. Inspired by our follow-up analysis, we propose a simple modification that has consistently and substantially brought performance gain over all systems: adding the intent label names into the support set. This strategy not only improves over the performance from general-purpose sentence encoding, but also the state-of-the-art results from the fine-tuned ProtAugment framework.

One limitation of our study for now is that the sentence representation in use is mostly based on BERT variants. It could be technically interesting to experiment with models pre-trained in a sequence-to-sequence fashion (Lewis et al., 2020; Raffel et al., 2020), which might have better captured semantic paraphrase representations via denoising or other training objectives.

Acknowledgments

We would like to thank Jin-Ge Yao for helpful discussion and valuable feedback on this study. We thank all the anonymous reviewers for their suggestions to improve the initial draft.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI'08*.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 4217–4226. Association for Computational Linguistics.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 6045–6049. IEEE.

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

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021a. A neural few-shot text classification reality check. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 935–943, Online. Association for Computational Linguistics.

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021b. ProtAugment: Intent detection meta-learning through unsupervised diverse paraphrasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2454–2466, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 3904–3913, Hong Kong, China. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings*

of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 879–895, Online. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

Yutai Hou, Yongkui Lai, Yushan Wu, Wanxiang Che, and Ting Liu. 2021. Few-shot learning for multi-label intent detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 13036–13044. AAAI Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1311–1316. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yue Li and Jiong Zhang. 2021. Semi-supervised meta-learning for cross-domain few-shot intent classification. In *Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing*, pages 67–75, Online. Association for Computational Linguistics.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019a. Benchmarking natural language understanding services for building conversational agents. In *IWSDS*.

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

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2773–2782, Online. Association for Computational Linguistics.

Shikib Mehri and Mihail Eric. 2021. Example-driven intent prediction with observers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2979–2992, Online. Association for Computational Linguistics.

Aaron Mueller, Jason Krone, Salvatore Romeo, Saab Mansour, Elman Mansimov, Yi Zhang, and Dan Roth. 2022. Label semantic aware pre-training for few-shot text classification. In *ACL 2022*, Online. Association for Computational Linguistics.

Thomas Müller, Guillermo Pérez-Torró, and Marc Franco-Salvador. 2022. Few-shot learning with siamese networks and label tuning. In *ACL 2022*, Online. Association for Computational Linguistics.

Hoang Nguyen, Chenwei Zhang, Congying Xia, and Philip Yu. 2020. Dynamic semantic matching and aggregation network for few-shot intent detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1209–1218, Online. Association for Computational Linguistics.

Jin Qu, Kazuma Hashimoto, Wenhao Liu, Caiming Xiong, and Yingbo Zhou. 2021. Few-shot intent classification by gauging entailment relationship between utterance and semantic label. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 8–15, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Timo Schick and Hinrich Schütze. 2021a. 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, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 4080–4090. Curran Associates, Inc.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2321–2331, Melbourne, Australia. Association for Computational Linguistics.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 451–462. Association for Computational Linguistics.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Simple and effective paraphrastic similarity from parallel translations. In *Proceedings of the Association for Computational Linguistics*.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Paraphrastic representations at scale. *arXiv preprint arXiv:2104.15114*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Congying Xia, Caiming Xiong, and Philip Yu. 2021. *Pseudo Siamese Network for Few-Shot Intent Generation*, page 2005–2009. Association for Computing Machinery, New York, NY, USA.

Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. 2020. Composed variational natural language generation for few-shot intents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3379–3388, Online. Association for Computational Linguistics.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y.S. Lam. 2020. Unknown intent detection using Gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1050–1060, Online. Association for Computational Linguistics.

Shuo Yang, Lu Liu, and Min Xu. 2021. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations (ICLR)*.

Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 244–249, Lisbon, Portugal. Association for Computational Linguistics.

Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. Few-shot intent classification and slot filling with retrieved examples. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 734–749, Online. Association for Computational Linguistics.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y.S. Lam. 2021a. Effectiveness of pre-training for few-shot intent classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1114–1120, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021b. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1906–1912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5064–5082, Online. Association for Computational Linguistics.

A Appendix

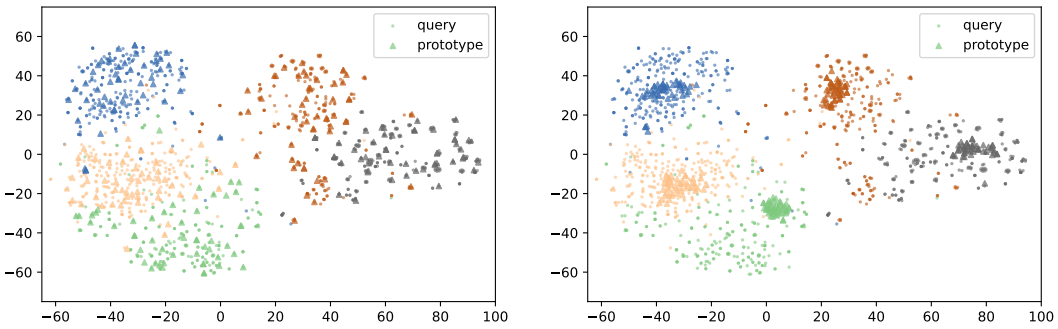
Extended Analysis for Label Semantics Since we add label names as additional support examples, the embedding space of sentences does not change in general sentence embedding based methods. For meta-learning based methods, we can take label names as support examples only in the meta-testing phase, or leverage them in both meta-training and meta-testing stages. The latter will change the embedding space. As shown in Table A.1, we find that adding label names at both stages generally yields slightly better performance than only using it during meta-testing.

To further validate the effectiveness of using label names, we sample fifty 5-way 1-shot episodes from five classes in Liu57 dataset. We visualize the query instances and prototype representations in the same embedding space. As shown in Figure A.1, in the 1-shot setting, the prototypes of the same class spread out and it is difficult to distinguish them from other classes. This indicates that the model struggles to extract the crucial information relevant to the intent class due to the limited information contained in one support example. However, adding label names helps centralize the prototypes of the same class and separate them away from those of other classes. This suggests that label names may be regarded as high-quality free examples which could help distinguish different classes.

Implementation Details We use Euclidean distance as the distance metric in ProtoNet for all sentence encoders, except for SP-paraphrase, in which cosine distance leads to much better performance. For meta-learning based methods, we use `RoBERTa-base` from Huggingface (Wolf et al., 2020) as the encoder. For the optimizer, we use Adam (Kingma and Ba, 2015) with a learning rate of 2e-5. We set the batch size to 32 and train the model for 10,000 episodes. We use the validation split to evaluate the model every 600 episodes and select the checkpoint with the best performance. The ProtAugment model has 123M parameters and the training lasts around one hour on four Tesla V100 GPUs.

Method	Banking77		HWU64		Liu57		Clinic150	
	K=1	K=5	K=1	K=5	K=1	K=5	K=1	K=5
L†-ProtoNet+MLM	93.86±0.44	<u>96.21</u>±0.48	89.81±0.96	93.99±0.96	91.09±2.15	<u>95.28</u>±0.74	97.64±0.27	98.93±0.19
L†-ProtAugment	<u>94.05</u>±0.54	<u>96.45</u>±0.44	90.79±0.88	94.11±0.72	92.21±1.32	<u>95.49</u>±0.55	98.15 ±0.24	99.17±0.22
L-ProtoNet+MLM	<u>94.00</u>±0.66	95.97±0.58	<u>91.30</u>±1.72	<u>94.09</u>±0.43	<u>92.00</u>±1.42	95.26±0.67	<u>98.36</u>±0.22	<u>99.08</u>±0.21
L-ProtAugment	93.42±1.42	96.11±0.75	<u>91.73</u>±1.23	<u>94.15</u>±0.58	<u>92.79</u>±1.28	95.34±0.88	<u>98.43</u>±0.17	<u>99.19</u>±0.22

Table A.1: Performance comparison under the 5-way K-shot settings. L represents leveraging label names in both meta-training and meta-testing stages. L† represents using label names during meta-testing only. The better performance between L† and L is <u>underlined</u>.



(a) SBERT-paraphrase (b) L-SBERT-paraphrase

Figure A.1: T-SNE visualization of prototypes and query instances for SBERT-paraphrase and L-SBERT-paraphrase. We randomly select five classes from the Liu57 dataset for exemplification.