Distinct Label Representations for Few-Shot Text Classification

Sora Ohashi[†], **Junya Takayama**[†], **Tomoyuki Kajiwara**[‡], **Yuki Arase**[†]
[†] Graduate School of Information Science and Technology, Osaka University
[‡] Graduate School of Science and Engineering, Ehime University
[†] {ohashi.sora, takayama.junya, arase}@ist.osaka-u.ac.jp
[‡] kajiwara@cs.ehime-u.ac.jp

Abstract

Few-shot text classification aims to classify inputs whose label has only a few examples. Previous studies overlooked the semantic relevance between label representations. Therefore, they are easily confused by labels that are semantically relevant. To address this problem, we propose a method that generates distinct label representations that embed information specific to each label. Our method is widely applicable to conventional few-shot classification models. Experimental results show that our method significantly improved the performance of few-shot text classification across models and datasets.

1 Introduction

Few-shot text classification (Ye and Ling, 2019; Sun et al., 2019; Gao et al., 2019; Bao et al., 2020) has been actively studied aiming to classify texts whose labels have only a few examples. Such infrequent labels are pervasive in datasets in practice, which are headaches for text classifiers because of the lack of training examples. Snell et al. (2017) showed that the conventional text classifiers are annoyed by the over-fitting problem when the distribution of labels is skewed in a dataset.

Few-shot classification has two approaches: metric-based and meta-learning based methods. The metric-based methods conduct classification based on distances estimated by a certain metric, *e.g.*, cosine similarity (Vinyals et al., 2016), euclidean distance (Snell et al., 2017), convolutional neural networks (Sung et al., 2018), and graph neural networks (Satorras and Estrach, 2018). Metric-based methods in natural language processing focus on representation generation that are suitable for few-shot classification using the attention mechanism with various granularity (Sun et al., 2019), local and global matching of representations (Ye and Ling, 2019), and word co-occurrence patterns in attention mechanisms (Bao et al., 2020). In contrast, meta-larning based methods *learn to learn* for achieving higher accuracy by learning parameter generation (Finn et al., 2017), learning rates and parameter updates (Li et al., 2017; Antoniou et al., 2019), and parameter updates using gradients (Andrychowicz et al., 2016; Ravi and Larochelle, 2017; Li and Malik, 2017).

TECH	Apple confirms it slows down old iPhones as their batteries age Self-driving cars may be coming sooner than you thought
BIZ	Apple apologizes for slowed iPhones, drops price of battery replacements Wall Street isn't too worried about first self-driving Tesla death

Table 1: Examples from Huffpost (BIZ: BUSINESS)

All of these previous studies overlooked the effects of the semantic relevance between label representations, which confuses few-shot classifiers. As a result, the classifiers tend to fail distinguishing examples with semantically relevant labels. Table 1 shows examples with labels sampled from Huffpost (Misra, 2018). The label pair of TECH and BUSINESS is semantically relevant, for which the classifiers are easily confused.

To address this problem, we propose a mechanism that compares label representations to derive distinctive representations. It learns semantic differences between labels and generates representations that embed information specific to each label. Our method can be easily applied to existing few-shot classification models.

We evaluated our method using the standard benchmarks of Huffpost and FewRel (Han et al., 2018). Experimental results showed that our method significantly improved the performance of previous few-shot classifiers across models and datasets, and achieved the state-of-the-art accuracy.

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing (Short Papers)*, pages 831–836
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Few-Shot Text Classification

This section describes the problem definition and a general form of conventional few-shot classifiers.

2.1 Problem Definition

In few-shot text classification, sets of supports and queries are given as input. A support set S consists of pairs of text x and corresponding label y: $S = \{(x_i, y_i) | i \in \{1, 2, \cdots, NK\}\}$. N is the number of label types in the support set and K is the number of samples per label type. A query set Q consists of M texts to be classified: $Q = \{q_j | j \in \{1, 2, \cdots, M\}\}$. Note that S and Q have the same set of label types. A few-shot text classifier aims to predict a label for each q_j.

In few-shot classification, training and evaluation are performed on a subset of a dataset called as *episode* (Vinyals et al., 2016). A setting of $N = n$ and $K = k$ is called as n-way k-shot classification. A training episode is created by sampling $k + m$ examples with n types of labels from a training set, and then by dividing them into support and query sets, where $m = \frac{M}{n}$. An evaluation episode is created in the same manner using an evaluation set. Note that labels in the training and evaluation episodes are exclusive, *i.e.*, the classifier is required to predict labels that it has not been exposed during training. The performance of a model is measured using the macro-averaged accuracy of all episodes.

2.2 General Form of Few-shot Text Classification Models

A classification model first converts texts in the support and query sets into vector representations. We denote a subset $S_l \subset S$ as $S_l = \{(x_l^p, y_l^p) | y_l^p = l, p \in \{1, 2, \cdots, K\}\}$ in which all texts have the same label l. An encoder $E(\cdot)$ converts x_l^p and a query q_j to vectors, $\boldsymbol{x}_i^p \in \mathbb{R}^d$ and $\boldsymbol{q}_j \in \mathbb{R}^d$ (d is the dimension of representations), respectively:

$$\boldsymbol{x}_l^p = E\left(x_l^p\right), \quad \boldsymbol{q}_j = E(q_j). \tag{1}$$

$E(\cdot)$ can be any text encoder, such as recurrent neural networks (Yang et al., 2016), convolutional neural networks (Kim, 2014), and pre-trained language models like BERT (Devlin et al., 2019).

Second, the classification model generates a label representation for l. Let $C(\cdot)$ be the function that generates the label representation $\boldsymbol{l} \in \mathbb{R}^d$:

$$\boldsymbol{l} = C\left(\boldsymbol{x}_l^1, \boldsymbol{x}_l^2, \cdots, \boldsymbol{x}_l^K\right). \tag{2}$$

$C(\cdot)$ is typically a pooling function, such as average pooling and max pooling.

Finally, the model calculates the similarity between $\boldsymbol{q}_j$ and each label representation $\boldsymbol{l}_i$ ($i \in \{1, 2, \cdots, N\}$) using a function $R(\cdot)$, and predicts a label whose representation is most similar to that of the query. The probability distribution of the i-th label is computed as:

$$p(i | \boldsymbol{l}_1, \cdots, \boldsymbol{l}_N, \boldsymbol{q}_j) = \frac{e^{R(\boldsymbol{l}_i, \boldsymbol{q}_j)}}{\sum_i e^{R(\boldsymbol{l}_i, \boldsymbol{q}_j)}}. \tag{3}$$

$R(\cdot)$ can be any metrics for estimating similarity. In natural language processing, cosine similarity is a standard choice.

As a loss function L_c, negative log-likelihood is commonly used:

$$L_c = -\frac{1}{M} \sum_{i=1}^{M} \log p(y_j), \tag{4}$$

where y_j is the gold-standard label of q_j.

3 Proposed Method

Figure 1 shows the overview of our method. It adds a mechanism for learning to generate distinctive label representations into conventional few-shot classification models by converting its training into multi-task learning. Our method adds a difference extractor (Section 3.1) and a loss function based on mutual information (Section 3.2) to an arbitrary few-shot classification model.

3.1 Difference Extractor

The difference extractor compares a set of N label representations $\boldsymbol{l}_i$ obtained by Equation (2) with each other and generates representations that retains only the information specific to each label. For doing so, a label representation should depend on a query q_j as classification is conducted based on similarity between the query and labels as shown in Equation (3) (Ye and Ling, 2019). Hence, we model both the label and query representations simultaneously. Specifically, the label representations $\boldsymbol{l}_1, \cdots, \boldsymbol{l}_N$ and the query representation $\boldsymbol{q}_j$ are transformed as:

$$\boldsymbol{H} = \text{MultiHeadAttention}(\boldsymbol{l}_1, \cdots, \boldsymbol{l}_N, \boldsymbol{q}_j), \tag{5}$$

$$\hat{\boldsymbol{l}}_i = \text{GELU}(\boldsymbol{W}_1 \boldsymbol{H}_{l_i} + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \tag{6}$$

$$\hat{\boldsymbol{q}}_j = \text{GELU}(\boldsymbol{W}_1 \boldsymbol{H}_{q_j} + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \tag{7}$$

where MultiHeadAttention$(\cdot)$ is a self-attention mechanism (Vaswani et al., 2017) that outputs

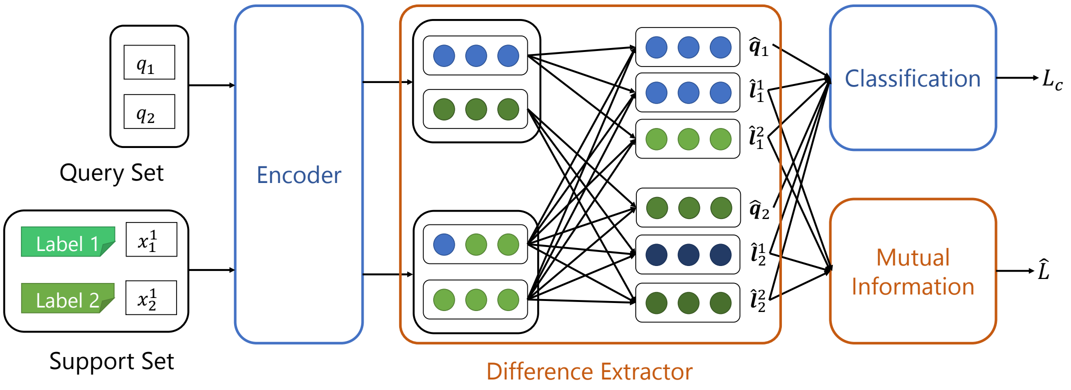

Figure 1: Outline of our method: Components of red boxes are applied to conventional few-shot classifiers.

$\boldsymbol{H} \in \mathbb{R}^{d\times(N+1)}$ hidden representations. $\boldsymbol{H}_{l_i} \in \mathbb{R}^d$ is an output of the self-attention corresponds to l_i, and similarly, $\boldsymbol{H}_{q_j} \in \mathbb{R}^d$ is that of q_j. These hidden representations are further transformed by fully-connected layers with the activation function of GELU$(\cdot)$ (Hendrycks et al., 2020).

3.2 Design of Loss Function

We assume that an ideal representation $\hat{\boldsymbol{l}}_i$ retaining only information specific to an i-th label satisfies that $I(\hat{\boldsymbol{l}}_i;\hat{\boldsymbol{l}}_r) = 0$ for all $\hat{\boldsymbol{l}}_r$ $(i \neq r)$, where $I(\cdot)$ computes mutual information (MI). That is, each label representation is independent. Hence, we propose an MI-based loss function $\hat{L}$, which constrains such that a label representation $\hat{\boldsymbol{l}}_i$ contains only information specific to the i-th label by minimizing:

$$\hat{L} = \sum_{1\leq i,r\leq N, i\neq r} I(\hat{\boldsymbol{l}}_i, \hat{\boldsymbol{l}}_r). \tag{8}$$

Because the exact value of Equation (8) is difficult to calculate in practice, we minimize its upper-bound following Cheng et al. (2020):

$$\hat{I}(\hat{\boldsymbol{l}}_i;\hat{\boldsymbol{l}}_r) = \sum_{j=1}^{|Q|} R_j, \tag{9}$$

$$R_j = \left[\log p_\theta(\hat{\boldsymbol{l}}_i^j|\hat{\boldsymbol{l}}_r^j) - \frac{1}{|Q|}\sum_{j'=1}^{|Q|} \log p_\theta(\hat{\boldsymbol{l}}_i^j|\hat{\boldsymbol{l}}_r^{j'})\right],$$

where $p_\theta(\cdot)$ is a neural network which approximates the conditional probability $p(\hat{\boldsymbol{l}}_i^j|\hat{\boldsymbol{l}}_r^j)$.

Finally, the overall loss function is:

$$L = L_c + \alpha\hat{L}, \tag{10}$$

where $\alpha(> 0)$ balances the effect of $\hat{L}$.

4 Experiment

We evaluated our method on different few-shot classification models using the standard benchmarks.

4.1 Benchmark Datasets

Following previous studies (Bao et al., 2020; Gao et al., 2019; Ye and Ling, 2019; Sun et al., 2019), we use Huffpost and FewRel as benchmarks.[1] Following these previous studies, we evaluated the performance of each model using $1,000$ episodes. Because episode generation involves random sampling from a dataset, we repeated this process for 10 times and computed the macro-averaged accuracy as the final score. The statistic significance was measured using a bootstrap significance test.

Huffpost This dataset consists of titles extracted from HuffPost[2] articles. The task is a prediction of a category of an article from its title. The training, validation, and test sets contain 20, 5, and 16 types of labels, respectively. The number of examples per label is 900.

FewRel The task is a prediction of a relation between entities. The training, validation, and test sets contain 65, 5, and 10 types of labels, respectively. The number of examples per label is 700.

4.2 Compared Models and Training Settings

We applied our method on three few-shot classifiers to investigate its effects on different models. As the de-facto standard of metric-based and

[1] Downloaded from https://github.com/YujiaBao/Distributional-Signatures

[2] https://www.huffpost.com/

	Huffpost				FewRel			
	5-Way		10-Way		5-Way		10-Way	
	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
ProtoNet	51.03	68.36	37.42	55.81	78.61	88.92	65.97	80.38
ProtoNet + DE	**51.76**	**69.07**	**38.08**	**56.85**	77.35	88.85	64.96	80.44
ProtoNet + DE + $\hat{L}$	**52.34****	**69.66***	**38.83***	**57.26***	**79.52***	**89.28***	**68.08***	**82.51***
MAML	51.10	65.23	37.37	51.74	68.94	76.49	58.07	65.01
MAML + DE	**51.80**	**67.28**	**38.36**	**53.54**	**75.45**	**85.06**	**62.33**	**72.31**
MAML + DE + $\hat{L}$	**51.71**	**67.38**	**38.11**	**53.75***	**75.99***	**84.07**	**63.13***	**70.99**
MLMAN	47.07	57.80	33.86	43.79	73.61	82.75	60.28	71.48
MLMAN + DE	**49.73**	**60.94**	**36.37**	**47.25**	**74.38**	**83.67**	**61.14**	**72.70**
MLMAN + DE + $\hat{L}$	**48.98**	**60.75**	**35.60**	**46.73**	**78.21***	**86.43***	**65.44***	**76.43***
Bao et al. (2020)	42.12	62.97	-	-	70.08	88.07	-	-

Table 2: Experimental results (**bold** and * indicate significantly higher accuracies compared to each baseline model and baseline + DE, respectively.)

meta-learning based models, we employed ProtoNet (Snell et al., 2017) and MAML (Finn et al., 2017), respectively. Besides, we employed MLMAN (Ye and Ling, 2019), which is the state-of-the-art few-shot classification model on FewRel. We also compared to Bao et al. (2020), which achieved the sate-of-the-art on HuffPost.

As the Encoder $E(\cdot)$ and pooling function $C(\cdot)$ for each model, we used the BERT-base, uncased[3] and average pooling, respectively, which showed strong performance in various text classification tasks (Devlin et al., 2019). We used PyTorch and Huggingface Transformers (Wolf et al., 2020) for implementation.[4]

We applied our difference extractor and MI-loss function (denoted as "+ DE + $\hat{L}$") to ProtoNet, MAML, and MLMAN. For the difference extractor, we used 1-layer self-attention mechanism with 8-heads. As an ablation study, we also compared our method that only applies the difference extractor (denoted as "+ DE"), which is trained only with the classification loss (Equation (4)).

We trained all models with 5-way 1-shot setting. We then tested the models on different ways and shots. As an optimizer, we used Adam (Kingma and Ba, 2015). A learning rate and α in Equation (10) were searched in ranges of $[1e-5, 3e-5, 5e-5]$ and $[1e-6, 1e-4, 1e-2, 1]$, respectively, to maximise accuracy on the validation set.

4.3 Overall Results

As Table 2 shows, our method significantly improved all of the baseline models across datasets.[5] For MAML and MLMAN, our difference extractor always improved the performance of the original models. By combination with the MI-loss, the performance improved by from 0.61 up to 7.68 points. In contrast, applying only the difference extractor to ProtoNet, *i.e.*, ProtoNet + DE, deteriorated its original performance on FewRel dataset. These results confirm that both the difference extractor and MI-loss are crucial for ProtoNet. By using both, ProtoNet + DE + $\hat{L}$ consistently improved the baseline by from 0.39 up to 2.13 points.

4.4 Impact of DE and MI-loss on Baselines

The experimental results confirmed that the combination of our difference extractor and MI-loss function consistently improved the few-shot classification models. In particular, MI loss is more effective for a simpler model, *i.e.*, ProtoNet. MLMAN has an internal mechanism for comparing supports and queries, and MAML has a mechanism for updating the model parameters to accurately classify supports. These internal mechanisms allow to learn label representations that boost classification accuracy. Hence, the functionality of MI loss is partly achieved by these internal mechanisms. On the other hand, ProtoNet has the simplest architec-

[3] https://github.com/google-research/bert

[4] Our code is available at https://github.com/21335732529sky/difference_extractor

[5] Note that the performance of ProtoNet was higher than that in (Ye and Ling, 2019) and (Bao et al., 2020). This is because we tuned the learning rate using the development set.

	Huffpost	FewRel
ProtoNet + DE + $\hat{L}$	$1e-4$	$1e-4$
MLMAN + DE + $\hat{L}$	$1e-4$	$1e-2$
MAML + DE + $\hat{L}$	$1e-6$	$1e-4$

Table 3: Weight of MI loss determined to maximise the performance on the development set

ture as described in Section 2.2 without additional mechanisms. Hence, both of the difference extractor and our loss function are crucial for ProtoNet.

Another factor affecting the performance of MI loss is the number of labels in a datset. When the number of labels is large, semantically relevant labels more likely exist, where MI loss plays a role. This assumption was empirically confirmed by the fact that FewRel, where MI loss (DE + $\hat{L}$) outperformed DE for most cases, has 80 labels. On the other hand, Huffpost has about half the number of labels (i.e., 41 labels).

4.5 Impact of Hyperparameters

Table 3 shows the settings of α tuned on the development set. Overall, the values of α on FewRel are larger than those on Huffpost. Larger α values increase the influence of MI loss on models, which is effective on datasets with a large number of labels like FewRel.

Figure 2 shows the accuracy measured on the development set when varying α. The performance tends to decrease when α is set too large. We suspect that too large α forces models to extract differences irrelevant to the classification task. For example, the second examples in Table 1 are about self-driving cars, where only the `BIZ` example contains named entities of `Wall Street` and `Tesla`. It is a noticeable difference; however, unlikely be useful for the classification task. Label representations of such spurious distinctiveness may degrade the classification performance.

5 Conclusion and Future Work

In this paper, we introduced a novel method shedding light on semantic relations between labels. Our method improved the classification accuracy of representative few-shot classifiers on both Huffpost and FewRel datasets, confirming the reasonable applicability of the proposed method.

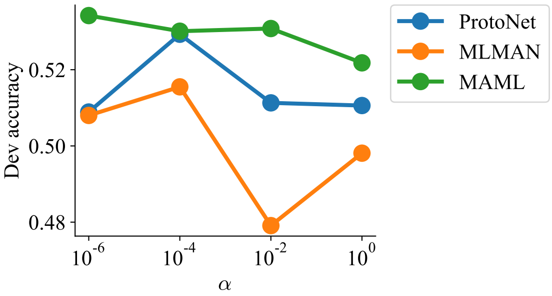

Figure 2: Accuracy on the Huffpost development set when varying α values

Technically, our method can be applied to other classification problems that handle semantic labels, such as image and entity classifications. We will conduct evaluation to see its effects on various types of classifcations.

Acknowledgments

This work was supported by JST AIP-PRISM Grant Number JPMJCR18Y1, Japan.

References

Marcin Andrychowicz, Misha Denil, Sergio Gómez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. 2016. Learning to Learn by Gradient Descent by Gradient Descent. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, pages 3981–3989.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2019. How to Train Your MAML. In *Proceedings of the 7th International Conference on Learning Representations*, pages 1–11.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot Text Classification with Distributional Signatures. In *Proceedings of the 8th International Conference on Learning Representations*, pages 1–24.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving Disentangled Text Representation Learning with Information-Theoretic Guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7530–7541.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 6407–6414.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained Transformers Improve Out-of-Distribution Robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–15.

Ke Li and Jitendra Malik. 2017. Learning to Optimize. In *Proceedings of the 5th International Conference on Learning Representations*, pages 1–13.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. 2017. Meta-SGD: Learning to Learn Quickly for Few Shot Learning. *arXiv:1707.09835*, pages 1–11.

Rishabh Misra. 2018. News Category Dataset.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a Model for Few-Shot Learning. In *Proceedings of the 5th International Conference on Learning Representations*, pages 1–11.

Victor Garcia Satorras and Joan Bruna Estrach. 2018. Few-Shot Learning with Graph Neural Networks. In *Proceedings of the 6th International Conference on Learning Representations*, pages 1–13.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pages 4077–4087.

Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical Attention Prototypical Networks for Few-Shot Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 476–485.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pages 5998–6008.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, pages 3630–3638.

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.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2872–2881.