Few-Shot Training and Transfer in NLP

Anonymous EMNLP submission

Abstract

Natural Language Processing (NLP) tasks are data-hungry and when the situation arises, where data is scarce, NLP models often fail to carry out reliable generalizations. Humans can, however, generalize only by seeing a few labeled examples on a specific task. Motivated by this, the rise in popularity in techniques that can generalize to new tasks containing only a few samples, called Few-Shot Learning, was inevitable. This survey discusses pre-trained Language Models and Meta-Learning for Few-Shot Training and Transfer in NLP, critically assess their application and identifies future work. Furthermore, we study the application of Few-Shot approaches in a cross-lingual setting.

1 Introduction

Deep learning methods defined NLP in recent years, achieving impressive performance when sufficient amounts of labeled data are available. However, from a practical view, in many tasks, a large scale dataset is not available, e.g. low-resource languages, and annotating new labeled data labels is expensive and time-consuming (Fort, 2016), leaving us with only building more efficient algorithms as conventional deep learning methods fail in this low data regime (Yogatama et al., 2019). Humans, on the other hand, only need a few demonstrations to learn new language tasks. Motivated by this, Few-Shot Learning tries to solve all those issues by learning just from a few labeled samples.

1.1 Few-Shot Scenario

Few-Shot Learning (FSL) is the ability to learn tasks with limited examples. Most existing FSL problems are supervised learning problems, which is our focus in this survey. In an (N-way-)K-shot classification problem, we are only given K labeled examples per class, where the number of classes is N. K-shot regression estimates a regression function given only K input-output example pairs.

To understand the challenges and approaches to Few-Shot Learning, we first analyze existing State-of-the-Art (SOTA) supervised approaches for NLP tasks.

1.2 Inducing Prior Knowledge

In a normal supervised setting, we would train our model on hundreds of thousands to millions of input–output pairs, which found success in numerous fields. However studies show that in NLP tasks, this paradigm of supervised learning does not generalize well outside the training data characteristics (Jia and Liang, 2017; Belinkov and Bisk, 2018), even when provided with enormous training data. The models are sensitive to noise, adversarial examples and are prone to overfitting. The reason is that language is complex and diverse and when conditions change, e.g. a new domain, the model is not able to adapt. Without any modification to the supervised approach, our Few-Shot Learning scenario will even amplify the poor generalization.

The most prominent way to help generalization is to induce an inductive bias by using transfer learning (Ruder, 2019), especially using pre-trained representations. In the last years, NLP saw the rise of pre-trained language representations for downstream tasks, achieving new SOTA on many NLP tasks. First, single-layer representations using word embedding vectors (Mikolov et al., 2013a) followed by contextualized word embeddings (Dai and Le, 2015; McCann et al., 2018a; Peters et al., 2018) were proposed, which were both simply fed into a task-specific architecture. With the rise of transformer language models (Vaswani et al., 2017), which enable direct finetuning of the whole architecture, there was no need for task-specific architectures anymore (Devlin et al., 2019). This was a breakthrough for NLP as many SOTA on NLP tasks were achieved by finetuning on task-specific samples using transformers, that are simply pre-trained on a language modeling objective

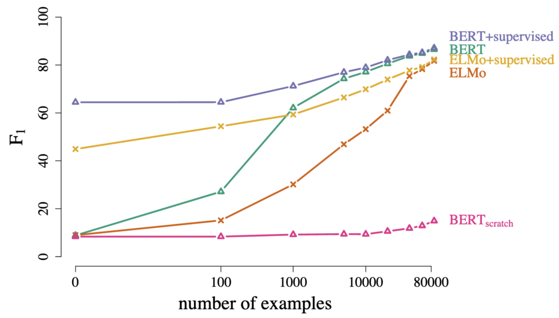

Figure 1: F_1 scores on SQuAD as a function of the number of training examples (log scale) (Yogatama et al., 2019). BERT$_{\text{+supervised}}$ and ELMo$_{\text{+supervised}}$ denote BERT and ELMo models that are pre-trained on other similar tasks, BERT$_{\text{scratch}}$ denotes a Transformer with a similar architecture to BERT that is trained from scratch.

in a semi-supervised fashion, inducing contextualized word embeddings. Clark et al. (2019) show that a pre-trained transformer model, like BERT, obtain knowledge about characteristics of the language, e.g. syntax and semantics as well as certain facts about the world, in short, have some general-purpose language understanding capacity, which can explain the generalization ability on a finetuned task.

1.3 Few-Shot Learning Challenges

One could naively apply the same strategy as in a normal supervised setting for our Few-Shot scenario: Finetune a pre-trained transformer model on the few labeled examples. Even though the pre-training helps the model to generalize on a new task, a sufficient amount of labeled data (Yogatama et al., 2019) is still needed in order to get reasonable results. The Figure 1 shows that in Few-Shot scenarios (i.e. < 1000 examples), all models lack far behind the fully trained variants, indicating *sample inefficiency* of the transformer model BERT. Additionally, the BERT$_{\text{scratch}}$ does not learn much without inducing an inductive bias via pre-training, showing the importance of the procedure. The Figure 1 also shows that, pre-training (sequentially) on similar tasks can help in a Few-Shot scenario, however, the sample inefficiency remains. Similar to this, Multi-Task Learning leverages information contained in multiple related tasks to help improve the generalization performance on all tasks (Zhang and Yang, 2021). Nonetheless, the method favors tasks with significantly more data, making it unsuitable for Few-Shot tasks. Another problem of big transformers is that they suffer from high variance (Phang et al., 2019; Dodge et al., 2020). This is amplified in a Few-Shot scenario, where Language Models only finetune on a few samples (Zhang et al., 2021; Zhao et al., 2020). Changing the set of training examples can result in significant performance differences. Therefore, it is essential to use the same set or average between multiple equal sets when comparing Few-Shot approaches, making it hard to compare different approaches. (Zhang et al., 2021) provide alternative practices to reduce instability.

The question remains, how a transformer model can effectively leverage the few given examples without suffering from high variance. Section 2 describes a method that tries to exploit the induced bias of pre-trained language models explicitly through using the model directly by reformulating the task as a language model problem. Driven by the results of pre-training sequentially on similar tasks, see Figure 1, Section 3 will analyze Meta-Learning Approaches, which also "pre-train" on similar tasks to induce an inductive bias with the goal to use the model in a Few-Shot scenario. Section 4 will cover the use-case of K-Shot Cross-Lingual Transfer. Finally, Section 5 will conclude this survey.

2 Reformulating Tasks as Language Modelling Problems

As pre-trained language models possess some general language purpose understanding, the idea is to solve Few-Shot Learning tasks through directly using the obtained linguistic knowledge by reformulating tasks as language modeling problems and then predicting labels as "fill-in-the-blank" tasks, sharing the same format as pre-training LMs.

2.1 Approaches

Brown et al. (2020) introduces **GPT-3**, which essentially uses the same model as GTP-2 (Radford et al., 2019), but scales the data, training time, and model to 175 billion parameters. On the contrary to standard finetuning that condition on the task on the algorithmic level $p(output|input)$, the idea of GPT series is to condition the model on the selected task $p(output|input, task)$, by inducing the task into the text sequence with a task description. A reading comprehension training example could be formulated as (`answer the question`, `document question`, `answer`).

To create a training set, they scraped web pages, but with a focus on document quality. The hope is that task formulations occur naturally in the dataset. Brown et al. (2020) explore different settings for *learning within context*, which means that during inference, the model is given a *prompt*, which consists of a task description and K examples of context and completion, which they call model *priming*. Then to make predictions, one final context is given, but the model has to fill in the completion. One important note is, that the model does not do any weight updates during inference, even after seeing the K examples, leaving room for more optimization. Additionally, K is upper bounded by the context windows size ($n_{ctx} = 2048$), meaning that typically the window fits around 10 to 100 examples. With this strategy of using the pre-trained language model directly, GTP-3 shows impressive Few-Shot capabilities across diverse tasks, surpassing some strong finetuned models baselines, such as tasks in the SuperGLUE benchmarks (Wang et al., 2020) by only giving 32 labeled examples. However, for finding the right prompt, a hold out set is necessary, which then in return needs more examples. As we are in a Few-Shot scenario, this makes it difficult to obtain a sufficiently large hold out set. As GTP-3 naively concatenates the K randomly selected examples (as the model's input size is bounded) with the input to create their in-context learning, the model does not make sure that the most informative demonstration are prioritized. However, prioritizing is important, since the number of usable demonstrations is bounded by the model's input size. We will call this problem *in-context selection problem*. Furthermore, as GPT-3 uses an autoregressive language model, experiments do not include any bidirectional architectures, even though Raffel et al. (2020) indicate that (finetuned) models benefit from such bidirectionality to solve NLP tasks. Finally, as GPT-3 has 175 billion parameters, performing inference is expensive and makes it impracticable for many applications.

Schick and Schütze (2021b) introduce **iPET**, a task-agnostic method for Few-Shot Learning that can perform on par with the GTP-3 model on the SuperGLUE dataset using a 785 times smaller Language Model, making the approach more "greener" and practical. Instead of providing prompts, as in the GPT models, iPET uses pattern-exploiting training (PET) (Schick and Schütze, 2021a), which reformulates tasks as cloze questions (no additional context samples provided) with regular gradient-based finetuning. Addititionally, the model utilizes gradient steps after seeing the K examples. For that, PET requires a pattern-verbalizer pairs (PVPs) $\boldsymbol{p} = (P, v)$, which maps the input x of a task to a cloze question formulation. They call this a pattern P. Then for each possible output y of the task, PET maps it to a single token, representing its task-specific meaning in the pattern, called verbalizer v. Now, given a pre-trained masked language model, we only have to check the probabilities of the mapped output $v(y)$ being the correct token at the masked position. To generate good PVPs on a small development set of held out tasks, PET uses a combination of 3 PVPs per pattern for which a separate pre-trained MLM is first finetuned on the given (small) training set and then used to annotate unlabeled examples. Finally, the soft-labeled dataset is used to finetune a single sequence classifier, which is closely related to knowledge distillation (Hinton et al., 2015). However, PET only works when the answer is a single token. Schick and Schütze (2021b) proposes iPET, which modifies PET to handle more than just one token during predictions and refines the generation of PVPs by enabling them to learn from each other. Schick and Schütze (2021b) shows that iPET with ALBERT (Lan et al., 2020) as the underlying LM achieves similar results on the SuperGLUE dataset as GTP-3, given 32 examples. Additionally, iPET with ALBERT only uses 223 million parameters, which is a magnitude smaller than GTP-3. Even though iPET mitigates the problems of choosing a single cloze question formulation (pattern) by combining multiple formulations, it still requires engineering a set of suitable patterns. Furthermore, iPET requires additional unlabeled data, which it uses in the knowledge distillation stage. This can be hard to acquire, where samples are pairs of text with a label, constructed to test a model's natural language understanding abilities (e.g. SuperGLUE). (Tam et al., 2021) propose **ADAPET**, which uses no unlabeled data by providing more supervision by modifying PET's objective. ADAPET outperforms iPET on SuperGLUE without any unlabeled data.

GTP and PET models use prompt-based (pattern-based) prediction, but finding the right prompt/pattern is an art. Gao et al. (2020) proposes **LM-BFF**, which alleviates this problem by generating the prompt automatically given a few examples, outperforming or matching manually selected prompts. Gao et al. (2020) first finds a label word mapping given a template (pattern) and then generates a diverse set of templates from the fixed set of label words by using the T5 model (Raffel et al., 2020). Even though Gao et al. (2020) propose a way to automatically find prompts, it still needs an "initial" template (pattern) or label words, inducing a bias that could potentially restrict the search space to a suboptimal one. Contrary to iPET, LM-BFF uses demonstrations for each input by concatenating them for additional context. However, to combat the *in-context selection problem* (see GTP-3), LM-BFF randomly selects a single example from *each* class for each input iterative at a time to create multiple, minimal demonstration sets, making it more efficient for Few-Shot tasks than GTP-3. As the underlying Language model, they use RoBERTa large model, which is again a magnitude smaller than GTP-3, with $K = 16$ examples and then use prompt-based finetuning with demonstrations. Notice that the finetuning process is different than iPET, which does not use any demonstrations, and GPT-3's in-context learning, which simply concatenates the input with demonstrations randomly drawn from the training set with no finetuning. Gao et al. (2020) evaluates on 8 tasks from the GLUE benchmark (Wang et al., 2019), SNLI (Bowman et al., 2015), and sentence classification tasks. Gao et al. (2020) show that their method of prompt-based finetuning outperforms standard finetuning (on $K = 16$ examples), except for the CoLA task and outperforms the GPT-3-style in-context learning. They also show that using demonstrations in context performs better than without any demonstrations in the context.

2.2 Discussion: Reformulating Tasks for NLP Few-Shot Tasks

Even though the models presented here, achieve impressive results with only a small amount of examples, it is still lacking quite far behind SOTA models that finetune on big datasets with thousands of examples. These approaches also favor tasks, that can naturally be reformulated as "fill-in-the-blank" problems, such as sentiment classification (e.g. positive class: "A fun ride. All in all **great**."), leaving room for future work. Additionally, methods require manual work to find a good reformulation for tasks. This problem is amplified in practical situations, where we want to deploy such systems since we need domain and model expertise to find an optimal reformulation by hand for unseen tasks. Even though Gao et al. (2020) try to mitigate this problem by automatically find reformulations, LM-BFF still needs an initial reformulation. Additionally, Language Models have a restricted input size. Tasks that have too long input sequences can not be properly solved. Future work could investigate using Language Models that allow such long input sequences, e.g. Longformer (Beltagy et al., 2020). Furthermore, these approaches finetune the downstream tasks in isolation, not utilizing any information from similar tasks.

3 Meta-Learning Algorithms

Additionally to the general-purpose language understanding properties of pre-trained language models, Meta-Learning algorithms try to induce another inductive bias, which allows the model to quickly adapt after only seeing a few examples. In comparison to the methods described in Section 1.2, Meta-Learning explicitly take the Few-Shot scenario into account and utilize information from similar tasks. This is achieved by collecting many training tasks, where each training task consists of a training dataset $\mathcal{D}_i^{tr} = \{(\boldsymbol{x}_{tr}, \boldsymbol{y}_{tr})\}$, called support set S, and a test set $\mathcal{D}_i^{val} = \{(\boldsymbol{x}_{val}, \boldsymbol{y}_{val})\}$. The idea is to then pre-train on them such that the final model can generalize to new tasks rapidly, which allows us to perform Few-Shot tasks.

We will discuss 2 popular forms of Meta-learning for NLP tasks (Yin et al., 2020a): Metric-based and Optimisation-based learning.

3.1 Metric-based Meta-Learning

The idea in metric-based Meta-Learning is to learn a representation space through the training tasks, which enables us to classify test instances correctly by just comparing them to the K labeled examples in this representation space.

Vinyals et al. (2017) proposes **Matching Networks**, which use two different embedding functions, one for the training examples and one for the test examples. The representation of one example can change, depending on the given support set $\mathcal{D}_i^{tr}$ for the task $\mathcal{T}_i$. For a test example $\hat{x}$, given its support set S, we choose the class with the highest aggregated similarity between class examples in the support set and the test instance by calculating the cosine similarity in the embedding space. Vinyals et al. (2017) evaluated Matching Networks on Few-Shot language modeling. Even though the approach found many applications in image classification, it has not yet found any impressive results in NLP tasks. One of the reasons is that matching networks do not finetune on the support set during inference, making it hard to find a good general embedding space that would work for many NLP tasks since text is quite diverse. If Matching Networks choose to finetune, it suffers from overfitting issues (Vinyals et al., 2017), not gaining much in performance.

To enable finetuning during inference and not suffer from overfitting, Snell et al. (2017) propose **Prototypical Networks**, which induce a simple bias: There exists an embedding space where points that belong to one class, cluster around a single prototype representation. For that, they learn a non-linear mapping of the input into an embedding space using a neural network and calculate the class' prototype as the mean of its support set in the embedding space. Finally, we can classify a new instance by finding the nearest class prototype. In comparison to Matching Networks, they do not compare instances to each other but use the prototypes (class representation) calculated from the support set. Therefore, only in a Few-Shot scenario, the approaches differ. Additionally, Prototypical Networks use Euclidean distance which outperforms the proposed cosine similarity of Matching Networks (Snell et al., 2017). Prototypical networks were first originally suggested for images in computer vision problems, however, the method was also applied to NLP tasks. Most applications use pre-trained word embeddings and instead of averaging to calculate the prototype class, they use more sophisticated methods, such as attention-based prototypes, reaching new SOTA on some benchmarks (Han et al., 2018; Gao et al., 2019; Hui et al., 2020; Deng et al., 2020) and also finding applications in domain transfer (Bansal et al., 2019). However, as the metric plays an important role in gaining performance (Snell et al., 2017), Sung et al. (2018) introduces a learnable metric instead of a fixed metric, calling it **Relation Networks**. Yu et al. (2018) try to solve diverse Few-Shot text classification by extending Prototype Networks with clustering similar training tasks, learning one metric for each, and then automatically determining the best weighted combination of those metrics for a newly seen Few-Shot task.

Matching, Prototypical and Relation Networks in NLP are mostly restricted to test tasks that are very similar to the training tasks, e.g. doing domain transfer. When we have diverse NLP tasks, finding an appropriate metric space becomes much harder. Yu et al. (2018) try to solve diverse Few-Shot text classification by extending Prototype Networks with clustering similar training tasks, learning one metric for each, and then automatically determining the best weighted combination of those metrics for a newly seen Few-Shot task. They show significant gains on Few-Shot sentiment classification and dialog intent classification tasks, indicating that clustering related tasks to handle diverse Few-Shot NLP tasks, might be a good research direction to improve metric-based or even optimization-based Meta-learning approaches for Few-Shot NLP tasks. A closely related method to metric-based approaches is supervised contrastive learning (Gunel et al., 2021) as both rely on capturing the similarity between examples in one class and contrasting them with examples in other classes. Instead of the usual Cross-Entropy Loss of Language Models, which is prone to high variance, Gunel et al. (2021) propose a loss function, consisting of cross-entropy and their supervised contrastive learning (SCL) term that pushes examples from the same class closer and the examples from different classes further apart. Gunel et al. (2021) obtain significant improvements over a strong RoBERTa-Large baseline on multiple datasets of the GLUE benchmark in few-shot learning settings. This method is closely related to metric-based approaches as both rely on capturing the similarity between examples in one class and contrasting them with examples in other classes.

3.2 Optimization-based Meta-Learning

In contrast to metric-based Meta-Learning, optimization-based Meta-Learning approaches try to learn a *good set of parameter initialization*, such that the model can quickly converge to a minimum in just a few gradient descent steps.

Finn et al. (2017) proposed the first optimization-based model, called **Model-Agnostic Meta-Learning (MAML)**. Let θ denote the parameter initialization of the model, ϕ_i the finetuned model parameters and $\mathcal{L}_i$ the loss function of each task $\mathcal{T}_i$. The idea is to first sample a (or batch of) task $\mathcal{T}_i$ with the corresponding (disjoint) datasets $\mathcal{D}_i^{tr}, \mathcal{D}_i^{val}$. To train the model on $\mathcal{D}_i^{tr}$, gradient-finetune with respect to the loss function $\mathcal{L}_i$ to obtain ϕ_i. We then can update the original initial model parameters θ using the "test" loss $\mathcal{L}_i(\phi_i, \mathcal{D}_i^{val})$ across sampled tasks

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}_i(\phi_i, \mathcal{D}_i^{val}). \tag{1}$$

This enables MAML to find a good parameter initialization that can quickly converge to a minimum, making it suitable for Few-Shot Learning. The model was used for Few-Shot text classification (Han et al., 2018; Obamuyide and Vlachos, 2019; Jiang et al., 2018; Bao et al., 2020), where each class is considered a task. Additionally, Jiang et al. (2018) introduces task-agnostic parameters and task-specific parameters to MAML, which they call **ATAML**, outperforming vanilla MAML on Few-Shot topic classification. MAML has also seen applications in Few-Shot domain adaption, e.g. Few-Shot dialogue system (Lin et al., 2019; Mi et al., 2019; Qian and Yu, 2019), where each domain dialog is treated as a task. One problem that MAML has, is that it is both computationally and memory intensive since it needs to calculate second derivatives in equation (1) as we get a nested back-propagation, where second derivates may come up. **First-Order MAML (FOMAML)** and **REPTILE** (Finn et al., 2017; Nichol et al., 2018) are methods which approximate the second derivative. One of the biggest challenges is to apply MAML to diverse tasks, as most applications are limited to similar train and test tasks, e.g. domain adaption tasks or to simulated classification datasets where each label is considered a task. Furthermore, even though the approach itself is model agnostic, meaning we can combine any model representation and any differentiable objective, the approach is restricted to tasks that have the same label space since to learn a good initialization, MAML requires sharing model parameters, including softmax classification layers across tasks.

To enable MAML to learn across diverse tasks with disjoint label spaces, Bansal et al. (2019) proposes **LEOPARD**, which uses a parameter generator, which learns on $\mathcal{D}_i^{tr}$ to generate *task-dependent initial softmax classification parameters* for any specific task. Furthermore, the approach transforms the text input into a feature representation by using a (shared) BERT model across tasks. To find a good parameter initialization, LEOPARD uses a modified MAML-based adaptation method by distinction between *task-specific parameters*, which are adapted per task, and *task-agnostic parameters*, which are shared across tasks. This is similar to Jiang et al. (2018). This allows for more efficient adaptation. Since BERT has a high number of parameters, LEOPARD uses lower-layers of BERT as task-agnostic parameters and higher-level layer and the softmax generating function as task-specific parameters. Since we already used $\mathcal{D}_i^{tr}$ to generate task-depended initial softmax classification parameters, we use subsequent batches for adaption. On the contrary to vanilla MAML, LEOPARD can handle test tasks that are notably different from the training tasks. They evaluate LEOPARD using target tasks that were *not seen during training* and evaluate on their *entire test* set after finetuning on K examples per label from the corresponding training set. The target tasks were selected such that they differ significantly from the training task and have a varying number of labels. They show that on average LEOPARD performs significantly better than the chosen baselines, BERT-base model (Devlin et al., 2019), Multi-task BERT (comparable to Liu et al. (2019)) and a Prototypical Network (Snell et al., 2017) that uses BERT-base as the underlying neural model. With that experiment, they show that LEOPARD can leverage Meta-learning to learn a more general-purpose parameter initialization that can then be used to solve completely unseen new tasks with just a few examples. Furthermore, Bansal et al. (2019) evaluate Few-Shot Domain-transfer, showing that LEOPARD performs on par or better than the baselines. They also show that prototypical networks give competitive results on domain-transfer tasks. One disadvantage of LEOPARD is that it requires labeled data from many different tasks, for training and also hyperparameter tuning. Additionally, it suffers from overfitting to the training task-distribution (Bansal et al., 2019, 2020) (*Meta-overfitting*), leaving room for a more efficient adaption to diverse tasks.

3.3 Discussion: Meta-Learning for NLP Few-Shot Tasks

One of the main challenges in Meta-Learning (in general) is to create training tasks that enable Meta-learning algorithms to find a good initialization set to solve the target task (Vinyals et al., 2017). As previously mentioned, many applications create training tasks from a fixed task dataset, where we have many labels, by subsampling from the set of labels. While it enables to generalize to unseen labels, this can also lead to overfitting to the training task distribution, making it hard to generalize to unseen tasks (Yin et al., 2020a). Furthermore, one of the reasons why most Meta-learning algorithms were first proposed in image classification problems is because they have big labeled sets with a large number of labels. In NLP tasks, however, they are often restricted to a small number of labels, e.g. sentiment classification has only a few discrete labels. To remedy this, Bansal et al. (2020) propose a self-supervised approach to generate a Meta-learning task distribution from an unlabeled text by masking words from a specified vocabulary (or subsets of it) and posing it as a multi-class classification. Combining the generated tasks with the available supervised tasks can improve Meta-learning algorithms, such as LEOPARD (Bansal et al., 2020). However, as these generated tasks are only (masked language) classification tasks, this can lead to a narrow training-task distribution. Additionally, most of the research only explores classification problems, leaving room for future work to expand into more diverse problem structures and to find more suitable ways to generate diverse Meta-learning tasks.

One major obstacle for Meta-learning approaches is to solve diverse NLP Few-Shot tasks. Meta-Learning approaches may work well for simulated datasets, where we just subsample labels from one single task dataset and define them as training tasks because the underlying task does not change in this situation, e.g. the model was "pre-trained" to solve translation tasks. However, if you want to test on a truly unseen task, the model has to first learn the underlying task from a few given examples. Jiang et al. (2018); Bansal et al. (2019) mitigate this problem by introducing task-specific parameters and task-agnostic parameters for more efficient adaption. Another interesting approach for future work could be to combine Meta learning with additional task information, e.g. task descriptions, to solve new diverse tasks (approaches in Section 2 do this).

4 K-Shot Cross-Lingual Transfer with Multilingual Language Models

This section will deal with K-Shot Cross-Lingual Transfer as a use-case of Few-Shot Learning. Achieving SOTA on (monolingual) NLP tasks is usually done by using transformers, pre-trained on language modeling objectives in a semi-supervised fashion, and then finetuning on a specific NLP task, which in return need a lot of labeled training data. Those are available in common languages, such as the English language, however, in low resource languages models fail to generalize well. The idea is to transfer the knowledge about a task from a high resource language to another low resource language, called cross-lingual transfer (CLT).

To achieve CLT between tasks from different languages, one has to induce a shared representation space between the source and target language. Previous SOTA methods used to induce continuous cross-lingual representation spaces by using cross-lingual word embeddings (Mikolov et al., 2013b; Glavaš et al., 2019) and sentence embeddings (Artetxe and Schwenk, 2019). However, with transformers getting popular, this survey will focus on inducing multilingual word embeddings with transformers.

4.1 Zero-Shot Cross-Lingual Transfer

One way to try to combat sparsely labeled training data in one language is by pretraining transformer models on multiple languages and automatically induce a multilingual word embedding. This idea gave rise to powerful massively multilingual transformers, such as mBert, XLM-R, and the recently introduced mT5 (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021). These architectures can encode text from any of the languages seen in pretraining and allows for a very straightforward approach to Zero-Shot cross-lingual model transfer: Finetune the model using task-specific supervised training data from one high resource language (*source-training*) and predict on other languages by feeding the target language text into the finetuned model. Pires et al. (2019) show effective results of Zero-Shot cross-lingual transfer with mBERT on POS tagging and NER for related languages. Furthermore, Wu et al. (2020); K et al. (2020) show the cross-lingual potential of mBERT by extending the analysis. Nevertheless, the literature mostly showed good results in languages that were from the same language family or that had a large corpus in pretraining, languages such as German, Spanish or French. This concern is raised by multiple sources (Lauscher et al., 2020; Wu and Dredze, 2020), which show that the performance drops huge for distant target languages and target languages that have small pre-training corpus. Furthermore, Lauscher et al. (2020) empirically show that for massively multilingual transformers, pre-training corpora sizes affect the Zero-Shot performance in higher-level language understanding tasks (e.g. NLI and QA), whereas the results in lower-level language understanding tasks are more impacted by typological language proximity. To summarize, Zero-Shot cross-lingual transfer with source training is effective for languages that are linguistically similar and languages that have a great amount of data for pre-training. However, this scenario is almost always never the case for low resource languages, where cross-lingual transfer is needed. The next section will investigate Few-Shot transfer to mitigate the transfer gap.

4.2 Few-Shot Cross-Lingual Transfer

To improve upon the results of Zero-Shot CLT, which only uses source training, we now additionally exploit the K task-specific examples in the target language (Few-Shot cross-lingual scenario) by further finetuning on those K examples (*target-adapting*). Lauscher et al. (2020) experiment with Few-Shot CLT on lower-level structured prediction tasks (POS tagging, dependency parsing, and NER) and higher-level language understanding tasks (NLI and QA) with varying numbers of K examples. They show that distant languages gain much more in performance from Few-Shot data than closely related languages. Hedderich et al. (2020) use Few-Shot CLT on NER task on genuine low-resource languages like Hausa and isiXhosa, also showing significant improvements by finetuning on the few examples. Zhao et al. (2020) applied Few-Shot CLT with mBERT on POS, NER, and sequence classification, observing the same phenomenon. In summary, additional finetuning on the given few examples from the target language can significantly improve performances on distant languages - Exactly where Zero-Shot CLT fails. Since we only have to finetune on a small set of examples, this additional finetuning is not computationally expensive but shows promising results.

As we only discussed "naively" finetuning for target adaption, one could further investigate how to exploit the given examples efficiently. Zhao et al. (2020) investigated freezing parameters during finetuning to mitigate the overfitting problem, however, experiments show no significant improvements in performance. To use the few given examples more efficiently, Nooralahzadeh et al. (2020) use MAML to further find optimal initialization parameters (after source training), which then can be used for either Zero-Shot or again finetuning in a Few-Shot setup. However, the method requires many training tasks in low-resource languages. Future work could focus on using Meta-Learning further.

One downside of all Few-Shot CLT approaches is that you need labeled data in the low resource target language, which is typically hard to acquire. It may become costly to annotate data for minor languages, however as Lauscher et al. (2020) show, even 10 annotated instances can give substantial performance improvement. This begs the question if annotating data is more cost-efficient in the long run than using GPU hours.

5 Conclusion and Discussion

We studied two methods to tackle Few-Shot tasks in NLP: Using pre-trained Language Models and Meta-learning. Even though Meta-Learning provides diverse applications as most methods are task and model agnostic, they struggle to solve unseen diverse NLP tasks. Future work should investigate how to improve generalization to new tasks. Pre-trained language models can be effective by reformulating NLP tasks as language model problems, enabling Few-Shot abilities. However these methods require manual work to find a good reformulation and they favor tasks, that can be naturally reformulated as a "fill-in-the-blank" task. We then discussed a use-case of Few-Shot Learning: Few-Shot CLT. In CLT, we have the chance to first finetune in a rich-resource language, and then transfer the knowledge to a low-resource language. Using more sophisticated methods to train on high resource languages, e.g. Meta-Learning (Nooralahzadeh et al., 2020), can improve performance and is a promising research direction. Nevertheless, most methods need labeled examples in low resource languages, making them expensive to obtain. As previously discussed in Section 1.3, almost all Few-Shot techniques have high variance. Therefore, we identify the necessity of standardization of Few-Shot datasets. As a final word, there are other approaches to Few-Shot Learning in NLP that was not discussed in this survey, e.g. unifying NLP tasks formats (McCann et al., 2018b; Yin et al., 2020b; Raffel et al., 2020; Khashabi et al., 2020).

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2019. Learning to few-shot learn across diverse natural language classification tasks. *CoRR*, abs/1911.03863.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and A. McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. *ArXiv*, abs/2009.08445.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

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*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 151–159, New York, NY, USA. Association for Computing Machinery.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400.

Karën Fort. 2016. *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*, 1st edition. Wiley-IEEE Press.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6407–6414.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy. Association for Computational Linguistics.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning.

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, Brussels, Belgium. Association for Computational Linguistics.

Michael A. Hedderich, David Adelani, Dawei Zhu, Jesujoba Alabi, Udia Markus, and Dietrich Klakow. 2020. Transfer learning and distant supervision for multilingual transformer models: A study on african languages.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Bei Hui, Liang Liu, J. Chen, X. Zhou, and Yuhui Nian. 2020. Few-shot relation classification by context attention-based prototypical networks with bert. *EURASIP Journal on Wireless Communications and Networking*, 2020:1–17.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems.

Xiang Jiang, Mohammad Havaei, G. Chartrand, H. Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and S. Matwin. 2018. Attentive task-agnostic meta-learning for few-shot text classification.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers.

Zhaojiang Lin, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2018a. Learned in translation: Contextualized word vectors.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018b. The natural language decathlon: Multitask learning as question answering.

Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems.

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

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning.

Abiola Obamuyide and Andreas Vlachos. 2019. Model-agnostic meta-learning for relation classification with limited supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5873–5879, Florence, Italy. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2019. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert?

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Sebastian Ruder. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. thesis, National University of Ireland, Galway.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze questions for few shot text classification and natural language inference.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners.

J. Snell, Kevin Swersky, and R. Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS*.

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.

Derek Tam, R. R. Menon, M. Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. *ArXiv*, abs/2103.11955.

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

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2017. Matching networks for one shot learning.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? In *RepL4NLP@ACL*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. 2020a. Meta-learning without memorization.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020b. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence.

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.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2020. A closer look at few-shot crosslingual transfer: Variance, benchmarks and baselines.