Few-Shot Learning with Siamese Networks and Label Tuning

Thomas Müller and **Guillermo Pérez-Torró** and **Marc Franco-Salvador**
Symanto Research, Valencia, Spain
https://www.symanto.com
{thomas.mueller,guillermo.perez,marc.franco}@symanto.com

Abstract

We study the problem of building text classifiers with little or no training data, commonly known as zero and few-shot text classification. In recent years, an approach based on neural textual entailment models has been found to give strong results on a diverse range of tasks. In this work, we show that with proper pre-training, Siamese Networks that embed texts and labels offer a competitive alternative. These models allow for a large reduction in inference cost: constant in the number of labels rather than linear. Furthermore, we introduce label tuning, a simple and computationally efficient approach that allows to adapt the models in a few-shot setup by only changing the label embeddings. While giving lower performance than model fine-tuning, this approach has the architectural advantage that a single encoder can be shared by many different tasks.

1 Introduction

Few-shot learning is the problem of learning classifiers with only a few training examples. Zero-shot learning (Larochelle et al., 2008), also known as dataless classification (Chang et al., 2008), is the extreme case, in which no labeled data is used. For text data, this is usually accomplished by representing the labels of the task in a textual form, which can either be the name of the label or a concise textual description.

In recent years, there has been a surge in zero-shot and few-shot approaches to text classification. One approach (Yin et al., 2019, 2020; Halder et al., 2020; Wang et al., 2021) makes use of entailment models. Textual entailment (Dagan et al., 2006), also known as natural language inference (NLI) (Bowman et al., 2015), is the problem of predicting whether a textual premise implies a textual hypothesis in a logical sense. For example, *Emma loves apples* implies that *Emma likes apples*.

The entailment approach for text classification sets the input text as the premise and the text representing the label as the hypothesis. A NLI model is applied to each input pair and the entailment probability is used to identify the best matching label.

In this paper, we investigate an alternative based on Siamese Networks (SN) (Bromley et al., 1993), also known as dual encoders. These models embed both input and label texts into a common vector space. The similarity of the two items can then be computed using a similarity function such as the dot product. The advantage is that input and label text are encoded independently, which means that the label embeddings can be pre-computed. Therefore, at inference time, only a single call to the model per input is needed. In contrast, the models typically applied in the entailment approach are Cross Attention (CA) models which need to be executed for every combination of text and label. On the other hand, they allow for interaction between the tokens of label and input, so that in theory they should be superior in classification accuracy. However, in this work we show that in practice, the difference in quality is small.

Both CA and SNs also support the few-shot learning setup by fine-tuning the models on a small number of labeled examples. This is usually done by updating all parameters of the model, which in turn makes it impossible to share the models between different tasks. In this work, we show that when using a SN, one can decide to only fine-tune the label embeddings. We call this Label Tuning (LT). With LT the encoder can be shared between different tasks, which greatly eases the deployment of this approach in a production setup. LT comes with a certain drop in quality, but this drop can be compensated by using a variant of knowledge distillation (Hinton et al., 2014).

Our contributions are as follows: We perform a large study on a diverse set of tasks showing that CA models and SN yield similar performance for both zero-shot and few-shot text classification.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8532 - 8545

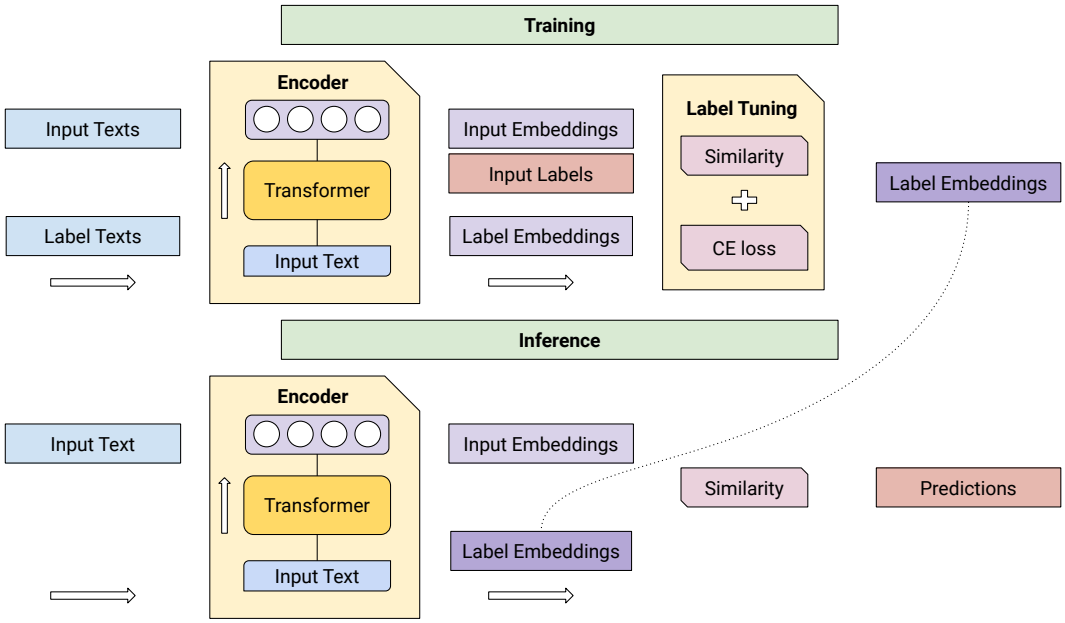

Figure 1: Overview of training and inference with Label Tuning (LT). At training time, input and label texts (hypotheses) are processed by the encoder. LT then tunes the labels using a cross entropy (CE) loss. At inference time, the input text is passed through the same encoder. The tuned label embeddings and a similarity function are then used to score each label. The encoder remains unchanged and can be shared between multiple tasks.

In contrast to most prior work, we also show that these results can also be achieved for languages other than English. We compare the hypothesis patterns commonly used in the literature and using the plain label name (identity hypothesis) and find that on average there is no significant difference in performance. Finally, we present LT as an alternative to full fine-tuning that allows using the same model for many tasks and thus greatly increases the scalability of the method. We will release the code[1] and trained models used in our experiments.

2 Methodology

Figure 1 explains the overall system. We follow Reimers and Gurevych (2019) and apply symmetric Siamese Networks that embed both input texts using a single encoder. The encoder consists of a transformer (Vaswani et al., 2017) that produces contextual token embeddings and a mean pooler that combines the token embeddings into a single text embedding. We use the dot product as the similarity function. We experimented with cosine similarity but did not find it to yield significantly better results.

As discussed, we can directly apply this model to zero-shot text classification by embedding the input text and a textual representation of the label. For the label representation we experiment with a plain verbalization of the label, or identity hypothesis, as well as the hypotheses or prompts used in the related work.

2.1 Fine-Tuning

In the case of few-shot learning, we need to adapt the model based on a small set of examples. In gradient-based few-shot learning we attempt to improve the similarity scores for a small set of labeled examples. Conceptually, we want to increase the similarity between every text and its correct label and decrease the similarity for every other label. As the objective we use the so called *batch softmax* (Henderson et al., 2017):

$$\mathcal{J} = -\frac{1}{B}\sum_{i=1}^{B}\left[S(x_i, y_i) - \log\sum_{j=1}^{B} e^{S(x_i, y_j)}\right]$$

Where B is the batch size and $S(x, y) = f(x) \cdot f(y)$ the similarity between input x and label text y under the current model f. All other elements of the batch are used as *in-batch negatives*. To this end, we construct the batches so that every batch contains exactly one example of each label. Note that this is similar to a typical softmax classification objective. The only difference is that $f(y_i)$ is computed during the forward pass and not as a simple parameter look-up.

[1] https://tinyurl.com/label-tuning

2.2 Label Tuning

Regular fine-tuning has the drawback of requiring to update the weights of the complete network. This results in slow training and large memory requirements for every new task, which in turn makes it challenging to deploy new models at scale. As an alternative, we introduce **label tuning**, which does not change the weights of the encoder. The main idea is to first pre-compute label embeddings for each class and later tune them using a small set of labeled examples. Formally, we have a training set containing N pairs of an input text x_i and its reference label index z_i. We pre-compute a matrix of the embedded input texts and embedded labels, $X \in \mathcal{R}^{N \times d}$ and $Y \in \mathcal{R}^{K \times d}$, respectively. d is the embedding dimension and K the size of the label set. We now define the score for every input and label combination as $S = X \times Y^T$ ($S \in \mathcal{R}^{N \times K}$) and tune it using cross entropy:

$$\mathcal{J}' = -\frac{1}{N} \sum_{i=1}^{N} \left[S_{i,z_i} - \log \sum_{j=1}^{K} e^{S_{i,j}} \right]$$

To avoid overfitting, we add a regularizer that penalizes moving too far from the initial label embeddings Y_0 as $\|Y_0 - Y\|_F$, where $\|.\|_F$ is the Frobenius norm.[2] Additionally, we also implement a version of dropout by masking some of the entries in the label embedding matrix at each gradient step. To this end, we sample a random vector $\vec{r}$ of dimension d whose components are 0 with probability *dropout* and 1 otherwise. We then multiply this vector component-wise with each row in the label embedding matrix Y. The dropout rate and the strength of the regularizer are two hyper-parameters of the method. The other hyper-parameters are the learning rate for the stochastic gradient descent as well as the number of steps. Following Logan IV et al. (2021), we tune them using 4-fold cross-validation on the few-shot training set. Note that the only information to be stored for each tuned model are the d-dimensional label embeddings.

2.3 Knowledge Distillation

As mentioned, label tuning produces less accurate models than real fine-tuning. We find that this can be compensated by a form of knowledge distillation (Hinton et al., 2014). We first train a normal fine-tuned model and use that to produce label distributions for a set of unlabeled examples. Later, this silver set is used to train the new label embeddings for the untuned model. This increases the training cost of the approach and adds an additional requirement of unlabeled data but keeps the advantages that at inference time we can share one model across multiple tasks.

[2] https://en.wikipedia.org/wiki/Matrix_norm#Frobenius_norm

3 Related Work

Pre-trained Language Models (LMs) have been proved to encode knowledge that, with task-specific guidance, can solve natural language understanding tasks (Petroni et al., 2019). Leveraging that, Le Scao and Rush (2021) quantified a reduction in the need of labeled data of hundreds of instances with respect to traditional fine-tuning approaches (Devlin et al., 2019; Liu et al., 2019). This has led to quality improvements in zero and few-shot learning.

Semantic Similarity methods Gabrilovich and Markovitch (2007) and Chang et al. (2008) use the explicit meaning of the label names to compute the similarity with the input text. Prototypical Networks (Snell et al., 2017) create class prototypes by averaging embedded support examples and minimizing a distance metric to them for classification of input examples. The class prototypes are similar to our label embeddings but we initialize them from the hypothesis and we only tune the embeddings instead of the entire encoder. Recent advances in pre-trained LMs and their application to semantic textual similarity tasks, such as Sentence-BERT (Reimers and Gurevych, 2019), have shown a new opportunity to increase the quality of these methods and set the stage for this work. Baldini Soares et al. (2019) use Siamese Networks apply to a few-shot relation extraction (RelEx) task. Their architecture and similarity loss is similar to ours, but they update all encoder parameters when performing fine-tuning. Chu et al. (2021) employ a technique called *unsupervised label-refinement* (LR). They incorporated a modified *k-means* clustering algorithm for refining the outputs of cross attention and Siamese Networks. We incorporate LR into our experiments and extend the analysis of their work. We evaluate it against more extensive and diverse benchmarks. In addition, we show that pre-training few-shot learners on their proposed textual similarity task NatCat underperforms pre-training on NLI datsets.

Prompt-based methods GPT-3 (Brown et al., 2020), a 175 billion parameter LM, has been shown to give good quality on few-shot learning tasks. Pattern-Exploiting Training (PET) (Schick and Schütze, 2021) is a more computational and memory efficient alternative. It is based on ensembles of smaller masked language models (MLMs) and was found to give few-shot results similar to GPT-3. Logan IV et al. (2021) reduced the complexity of finding optimal templates in PET by using *null-prompts* and achieved competitive performance. They incorporated BitFit (Ben-Zaken et al., 2021) and thus reached comparable accuracy fine-tuning only 0.1% of the parameters of the LMs. Hambardzumyan et al. (2021) present a contemporary approach with a similar idea to **label tuning**. As in our work, they use label embeddings initialized as the verbalization of the label names. These task-specific embeddings, along with additional ones that are inserted into the input sequence, are the only learnable parameters during model training. They optimize a cross entropy loss between the label embeddings and the output head of a MLM. The major difference is that they employ a prompt-based approach while our method relies on embedding models.

Entailment methods The entailment approach (Yin et al., 2019; Halder et al., 2020) uses the label description to reformulate text classification as textual entailment. The model predicts the entailment probability of every label description . Wang et al. (2021) report results outperforming LM-BFF (Gao et al., 2021), an approach similar to PET.

True Few-Shot Learning Setting Perez et al. (2021) argue that for *true few-shot learning*, one should not tune parameters on large validation sets or use parameters or prompts that might have been tuned by others. We follow their recommendation and rely on default parameters and some hyper-parameters and prompts recommended by Wang et al. (2021), which according to the authors, were not tuned on the few-shot datasets. For label tuning, we follow Logan IV et al. (2021) and tune parameters with cross-validation on the few-shot training set.

4 Experimental Setup

In this section we introduce the baselines and datasets used throughout experiments.

4.1 Models

Random The theoretical performance of a random model that uniformly samples labels from the label set.

Word embeddings For the English experiments, we use Word2Vec (Mikolov et al., 2013) embeddings[3]. For the multi-lingual experiments, we use FastText (Grave et al., 2018). In all cases we preprocess using the NLTK tokenizer (Bird et al., 2009) and stop-words list and by filtering non-alphabetic tokens. Sentence embeddings are computed by averaging the token embeddings.

Char-SVM For the few-shot experiments we implemented a Support Vector Machines (SVM) (Hearst et al., 1998) based on character n-grams. The model was implemented using the text vectorizer of scikit-learn (Pedregosa et al., 2011) and uses bigrams to fivegrams.

Cross Attention For our experiments we use pretrained models from HuggingFace (Wolf et al., 2020). As the cross attention baseline, we trained a version of MPNET (Song et al., 2020) on Multi-Genre (MNLI, Williams et al. (2018)) and Stanford NLI (SNLI, Bowman et al. (2015)) using the parameters and code of Nie et al. (2020). This model has approx. 110M parameters. For the multilingual experiments, we trained – the cross-lingual language model – XLM roberta-base (Liu et al., 2019) on SNLI, MNLI, adversarial NLI (ANLI, Nie et al. (2020)) and cross-lingual NLI (XNLI, Conneau et al. (2018)), using the same code and parameters as above. The model has approx. 280M parameters. We give more details on the NLI datasets in Appendix G.

Siamese Network We also use models based on MPNET for the experiments with the Siamese Networks. *paraphrase-mpnet-base-v2*[4] is a sentence transformer model (Reimers and Gurevych, 2019) trained on a variety of paraphrasing datasets as well as SNLI and MNLI using a batch softmax loss (Henderson et al., 2017). *nli-mpnet-base-v2*[5] is identical to the previous model but trained exclusively on MNLI and SNLI and thus comparable to the cross attention model. For the multilingual experiments, we trained a model using the code

[3] https://code.google.com/archive/p/word2vec

[4] https://tinyurl.com/pp-mpnet

[5] https://tinyurl.com/nli-mpnet

name	task	lang.	train	test	labels	token length
GNAD (Block, 2019)	topic	de	9,245	1,028	9	279
AG News (Gulli, 2005)		en	120,000	7,600	4	37
HeadQA (Vilares and Gómez-Rodríguez, 2019)		es	4,023	2,742	6	15
Yahoo (Zhang et al., 2015)		en	1,360,000	100,000	10	71
Amazon Reviews (Keung et al., 2020)	reviews	de, en, es	205,000	5,000	5	25-29
IMDB (Maas et al., 2011)		en	25,000	25,000	2	173
Yelp full (Zhang et al., 2015)		en	650,000	50,000	5	99
Yelp polarity (Zhang et al., 2015)		en	560,000	38,000	2	97
SAB (Navas-Loro et al., 2017)	sentiment	es	3,979	459	3	13
SemEval (Nakov et al., 2016)		en	9,834	20,632	3	20
sb10k (Cieliebak et al., 2017)		de	8,955	994	3	11
Unified (Bostan and Klinger, 2018)	emotions	en	42,145	15,689	10	15
deISEAR (Troiano et al., 2019)		de	643	340	7	9
COLA (Warstadt et al., 2019)	acceptability	en	8,551	1,043	2	7
SUBJ (Pang and Lee, 2004)	subjectivity	en	8,019	1,981	2	22
TREC (Li and Roth, 2002)	entity type	en	5,452	500	6	10

Table 1: Overview of the evaluated datasets. Token length is the median value.

of the sentence transformers with the same batch softmax objective used for fine-tuning the few-shot models and on the same data we used for training the cross attention model.

Roberta-NatCat For comparison with the related work, we also trained a model based on `RoBERTa` (Liu et al., 2019) and fine-tuned on the NatCat dataset as discussed in Chu et al. (2021) using the code[6] and parameters of the authors.

4.2 Datasets

We use a number of English text classification datasets used in the zero-shot and the few-shot literature (Yin et al., 2019; Gao et al., 2021; Wang et al., 2021). In addition, we use several German and Spanish datasets for the multilingual experiments. Table 1 provides more details.

These datasets are of a number of common text classification tasks such as topic classification, sentiment and emotion detection, and review rating. However, we also included some less well-known tasks such as acceptability, whether an English sentence is deemed acceptable by a native speaker, and subjectivity, whether a statement is subjective or objective. As some datasets do not have a standard split we split them randomly using a 9/1 ratio.

4.3 Hypotheses

We use the same hypotheses for the cross attention model and for the Siamese network. For Yahoo and Unified we use the hypotheses from Yin et al. (2019). For SUBJ, COLA, TREC, Yelp, AG News and IMDB we use the same hypotheses as Wang et al. (2021). For the remaining datasets we designed our own hypotheses. These were written in an attempt to mirror what has been done for other datasets and they have not been tuned in any way. Appendix B shows the patterns used. We also explored using an identity hypothesis, that is the raw label names as the label representation and found this to give similar results.

4.4 Fine-Tuning

Inspired by Wang et al. (2021), we investigate fine-tuning the models with 8, 64 and 512 examples per label. For fine-tuning the cross attention models we follow the literature (Wang et al., 2021) and create examples of every possible combination of input text and label. The example corresponding to the correct label is labeled as entailed while all other examples are labeled as refuted. We then fine-tune the model using stochastic gradient descent and a cross-entropy loss. We use a learning rate of 1e-5, a batch size of 8 and run the training for 10 epochs. As discussed in the methodology Section 2.1, for the Siamese Networks every batch contains exactly one example of every label and therefore the batch size equals the number of labels of the task. We use a learning rate of 2e-5 and of 2e-4 for the BitFit experiments. Appendix D contains additional information on the hyper-parameters used.

We use macro F1-score as the evaluation metric. We run all experiments with 5 different training sets and report the mean and standard deviation. For

[6] https://github.com/ZeweiChu/ULR

name	n	Yahoo	AG News	Unified	COLA	SUBJ	TREC	IMDB	SemEval	Yelp pol	Yelp full	Amazon	Mean
random	0	10.0	25.0	10.0	**50.0**	50.0	16.7	50.0	33.3	50.0	20.0	20.0	30.5
W2V (IH)	0	$44.8_{0.2}$	$59.1_{0.5}$	$10.1_{0.3}$	$46.9_{1.7}$	$37.1_{0.7}$	$17.6_{1.4}$	$71.0_{0.3}$	**$46.8_{0.3}$**	$65.9_{0.2}$	$14.8_{0.1}$	$17.8_{0.4}$	$39.3_{0.7}$
RoBERTa-NatCat	0	$50.0_{0.2}$	$49.8_{0.6}$	$7.9_{0.3}$	$35.5_{1.5}$	$44.3_{0.9}$	$18.6_{1.1}$	$45.6_{0.3}$	$36.6_{0.3}$	$49.8_{0.2}$	$11.1_{0.1}$	$11.2_{0.4}$	$32.8_{0.7}$
RoBERTa-NatCat (IH)	0	$37.3_{0.2}$	$62.6_{0.5}$	$15.2_{0.3}$	$42.3_{1.4}$	$40.4_{1.0}$	$22.2_{1.2}$	$39.9_{0.2}$	$30.9_{0.3}$	$47.7_{0.2}$	$17.5_{0.1}$	$17.5_{0.5}$	$33.9_{0.7}$
mpnet (CA)	0	$51.8_{0.1}$	$60.5_{0.6}$	**$23.3_{0.4}$**	$47.0_{1.4}$	$41.0_{0.9}$	$19.8_{1.6}$	**$87.5_{0.2}$**	$37.4_{0.3}$	**$88.4_{0.2}$**	**$36.7_{0.2}$**	$25.6_{0.6}$	$47.2_{0.8}$
mpnet (CA-IH)	0	$46.3_{0.2}$	$56.3_{0.5}$	$22.2_{0.4}$	$47.7_{1.5}$	**$55.7_{1.1}$**	$20.2_{1.5}$	$83.5_{0.2}$	$38.8_{0.2}$	$83.4_{0.2}$	$36.1_{0.2}$	$33.4_{0.6}$	$47.6_{0.8}$
mpnet (SN)	0	**$53.9_{0.1}$**	$62.5_{0.5}$	$21.6_{0.3}$	$46.0_{1.5}$	$42.0_{0.8}$	$31.5_{1.4}$	$73.8_{0.2}$	$46.7_{0.3}$	$78.6_{0.2}$	$26.1_{0.2}$	**$40.6_{0.6}$**	$47.6_{0.7}$
mpnet (SN-IH)	0	$51.4_{0.1}$	**$64.2_{0.6}$**	$21.2_{0.3}$	$46.0_{1.6}$	$54.0_{1.0}$	**$32.1_{1.7}$**	$69.6_{0.3}$	$41.5_{0.3}$	$83.6_{0.2}$	$34.3_{0.2}$	$37.4_{0.7}$	$48.7_{0.8}$
Char-SVM	8	$29.3_{1.6}$	$54.3_{2.5}$	$12.2_{1.1}$	$45.6_{1.8}$	$64.9_{3.9}$	$39.5_{3.9}$	$57.1_{3.5}$	$33.6_{1.1}$	$56.7_{5.4}$	$29.2_{1.8}$	$30.0_{1.6}$	$41.1_{2.9}$
mpnet (CA)	8	$58.3_{2.8}$	$80.6_{2.9}$	$23.6_{1.1}$	**$50.4_{2.1}$**	$75.2_{5.0}$	$66.4_{6.0}$	**$88.4_{0.9}$**	**$59.5_{1.3}$**	**$90.3_{1.9}$**	**$50.9_{2.1}$**	**$47.7_{1.3}$**	$62.8_{2.9}$
mpnet (CA-IH)	8	$59.2_{2.6}$	$83.1_{1.7}$	$23.0_{2.2}$	$48.4_{2.2}$	$74.6_{5.3}$	**$68.7_{7.7}$**	$87.2_{0.8}$	$58.2_{1.0}$	$88.9_{3.8}$	$49.3_{2.4}$	$47.3_{1.7}$	$62.5_{3.5}$
mpnet (SN)	8	**$62.0_{0.4}$**	$84.2_{1.5}$	**$24.8_{1.3}$**	$49.6_{1.8}$	$79.6_{5.4}$	$62.8_{6.4}$	$76.4_{1.6}$	$58.7_{2.4}$	$84.8_{1.8}$	$44.7_{2.0}$	$46.9_{1.7}$	$61.3_{3.0}$
mpnet (SN-IH)	8	$61.0_{0.9}$	**$84.4_{1.2}$**	$24.6_{1.1}$	$46.3_{2.7}$	**$80.5_{5.0}$**	$58.5_{2.4}$	$76.1_{1.9}$	$57.0_{3.2}$	$86.2_{0.4}$	$43.5_{1.8}$	$46.0_{1.8}$	$60.4_{2.4}$
Char-SVM	64	$49.0_{0.5}$	$76.6_{0.6}$	$17.3_{0.4}$	$48.5_{1.6}$	$79.6_{1.2}$	$60.4_{2.2}$	$70.9_{1.5}$	$39.0_{0.8}$	$77.3_{2.5}$	$41.8_{0.4}$	$43.5_{0.8}$	$54.9_{1.3}$
mpnet (CA)	64	$66.5_{0.9}$	**$87.9_{0.9}$**	$28.1_{1.3}$	$54.2_{0.8}$	$91.6_{1.4}$	$87.0_{1.9}$	**$90.7_{1.0}$**	$62.0_{2.4}$	**$93.5_{0.4}$**	**$57.0_{0.4}$**	**$54.1_{1.5}$**	$70.2_{1.3}$
mpnet (CA-IH)	64	$65.8_{0.4}$	$87.4_{1.0}$	$26.4_{0.6}$	$51.3_{2.2}$	$92.5_{0.5}$	$85.0_{2.1}$	$89.3_{0.5}$	**$62.6_{1.5}$**	$92.7_{0.4}$	$56.1_{0.6}$	**$54.1_{1.3}$**	$69.4_{1.2}$
mpnet (SN)	64	**$66.6_{0.4}$**	$87.7_{1.0}$	**$29.3_{0.3}$**	**$56.6_{1.8}$**	$92.0_{1.0}$	$87.7_{1.9}$	$79.7_{1.4}$	$61.9_{1.2}$	$88.7_{0.4}$	$50.8_{0.9}$	**$54.1_{1.4}$**	$68.6_{1.2}$
mpnet (SN-IH)	64	$66.5_{0.4}$	$87.3_{1.2}$	**$29.3_{0.5}$**	$46.5_{11.0}$	**$92.7_{0.3}$**	**$87.5_{3.1}$**	$79.7_{1.6}$	$61.5_{1.7}$	$88.1_{0.2}$	$50.7_{0.8}$	$54.0_{1.7}$	$67.6_{3.6}$
Char-SVM	512	$59.6_{0.2}$	$85.8_{0.3}$	$23.0_{0.4}$	$51.2_{1.1}$	$87.0_{0.6}$	$87.5_{0.7}$	$82.8_{0.5}$	$46.0_{0.5}$	$87.1_{0.2}$	$49.3_{0.3}$	$50.4_{0.4}$	$64.5_{0.5}$
mpnet (CA)	512	$67.1_{0.7}$	$90.2_{0.4}$	$32.4_{1.2}$	$68.5_{2.0}$	$94.6_{1.1}$	$95.2_{0.6}$	**$92.5_{0.2}$**	$63.6_{1.2}$	**$95.2_{0.3}$**	**$60.8_{0.4}$**	$60.1_{0.5}$	$74.6_{0.9}$
mpnet (CA-IH)	512	$67.7_{0.2}$	**$90.4_{0.3}$**	$32.8_{0.6}$	$68.0_{1.6}$	$94.9_{0.6}$	$94.4_{1.5}$	$90.1_{1.1}$	$63.7_{1.4}$	$94.6_{0.2}$	$59.5_{0.7}$	$59.7_{0.9}$	$74.2_{0.9}$
mpnet (SN)	512	**$68.9_{0.2}$**	$90.3_{0.3}$	$33.2_{0.3}$	**$74.3_{0.9}$**	**$96.1_{0.3}$**	**$95.3_{0.6}$**	$84.0_{0.3}$	**$64.6_{0.7}$**	$90.0_{0.3}$	$55.3_{0.3}$	**$60.4_{0.5}$**	$73.9_{0.5}$
mpnet (SN-IH)	512	**$68.9_{0.2}$**	$90.2_{0.2}$	**$33.5_{0.5}$**	$62.8_{19.6}$	$95.9_{0.4}$	$95.0_{0.6}$	$83.7_{0.3}$	$64.1_{0.8}$	$90.1_{0.2}$	$55.1_{0.3}$	$60.3_{0.6}$	$72.7_{5.9}$

Table 2: English results for models based on MPNET and trained on SNLI and MNLI, comparing Siamese architecture (SN) and cross attention (CA) and also models with a identity hypothesis (IH). Results are grouped by the number of training examples (n). Underlined results are significant. **Bold** font indicates maxima.

the zero-shot experiments, we estimate the standard deviation using bootstrapping (Koehn, 2004). In all cases, we use Welch's t-test[7] with a p-value of 0.05 to establish significance (following Logan IV et al. (2021)). For the experiments with label refinement (Chu et al., 2021) and distillation, we use up to 10,000 unlabeled examples from the training set.

5 Results

Here we present the results of our experiments. The two main questions we want to answer are whether Siamese Networks (SN) give comparable results as Cross Attention models (CA) and how well Label Tuning (LT) compares to regular fine-tuning.

5.1 Siamese Network and Cross Attention

Table 2 shows results comparing SN with CA and various baselines. As discussed above, SN and CA models are based on the MPNET architecture and trained on SNLI and MNLI.

For the zero-shot setup (n=0) we see that all models out-perform the random baseline on average. The word embedding baselines and RoBERTa-NatCat perform significantly worse than random on several of the datasets. In contrast the SN and CA models only perform worse than random on COLA. The SN outperforms the CA on average, but the results for the individual datasets are mixed. The SN is significantly better for 4, significantly worse for 4 and on par for the remaining 3 datasets. Regarding the use of a hypothesis pattern from the literature or just an identity hypothesis (IH), we find that, while there are significant differences on individual datasets, the IH setup shows higher but still comparable (within 1 point) average performance.

For the few-shots setup (n=$\{8, 64, 512\}$), we find that all models out-perform a Char-SVM trained with the same number of instances by a large margin. Comparing SN and CA, we see that CA outperforms the SN on average but with a difference with-in the confidence interval. For n=8 and n=64, CA significantly outperforms SN on 3 datasets and performs comparably on the remaining 8. For n=512, we see an even more mixed picture. CA is on par with SN on 6 datasets, outperforms it on 3 and is out-performed on 2. We can conclude that for the English datasets, SN is more accurate for zero-shot while CA is more accurate for few-shot. The average difference is small in both setups and we do not see a significant difference for most datasets.

Table 3 shows the multi-lingual experiments. The `RoBERTa` XLM models were pre-trained on data from more than 100 languages and fine-tuned on an NLI data of 15 languages. The cross-lingual data and the fact that there is only 7500 examples

[7] https://en.wikipedia.org/wiki/Welch%27s_t-test

language		German	English						Spanish			
name	n	GNAD	Amazon	deISEAR	sb10k	Amazon	SemEval	Unified	Amazon	HeadQA	SAB s	Mean
random	0	11.1	20.0	14.3	33.3	20.0	33.3	10.0	20.0	16.7	33.3	21.2
FastText	0	$17.3_{1.0}$	$15.4_{0.5}$	$22.2_{2.1}$	$31.5_{1.5}$	$18.6_{0.5}$	**43.8**$_{0.4}$	$11.8_{0.3}$	$19.7_{0.5}$	**45.0**$_{0.9}$	$35.0_{2.2}$	$26.0_{1.2}$
xlm-roberta (CA)	0	$28.5_{1.3}$	$24.4_{0.6}$	$21.1_{1.8}$	$34.1_{1.4}$	$23.8_{0.5}$	$33.1_{0.2}$	**16.5**$_{0.3}$	$24.1_{0.5}$	$36.7_{0.9}$	$29.5_{2.2}$	$27.2_{1.2}$
xlm-roberta (CA-IH)	0	$29.4_{1.3}$	$26.1_{0.6}$	$18.3_{1.5}$	$31.8_{0.9}$	$29.2_{0.6}$	$34.6_{0.2}$	$15.7_{0.4}$	$25.0_{0.5}$	$37.8_{0.9}$	$24.3_{1.5}$	$27.2_{1.0}$
xlm-roberta (SN)	0	**41.5**$_{1.2}$	**31.1**$_{0.7}$	$22.1_{1.9}$	**38.4**$_{1.2}$	**37.0**$_{0.6}$	$43.1_{0.3}$	$15.3_{0.3}$	$28.0_{0.6}$	$35.4_{0.9}$	$32.0_{2.3}$	$32.4_{1.2}$
xlm-roberta (SN-IH)	0	$38.9_{1.2}$	$29.5_{0.5}$	$23.0_{2.4}$	$35.7_{1.4}$	$31.0_{0.6}$	$38.7_{0.3}$	$13.7_{0.2}$	**32.9**$_{0.6}$	$38.8_{0.8}$	**35.6**$_{2.3}$	$31.8_{1.3}$
Char-SVM	8	$56.1_{2.8}$	$30.5_{2.2}$	$29.4_{1.6}$	$45.4_{2.5}$	$30.0_{1.6}$	$33.6_{1.1}$	$12.2_{1.1}$	$30.8_{1.2}$	$36.3_{2.6}$	$50.6_{5.3}$	$35.5_{2.5}$
xlm-roberta (CA)	8	$61.6_{2.4}$	$43.3_{1.3}$	**39.5**$_{5.1}$	$53.6_{2.2}$	$41.2_{2.2}$	$55.0_{3.4}$	$18.3_{1.4}$	$41.1_{1.3}$	$49.5_{2.7}$	$53.9_{3.5}$	$45.7_{2.8}$
xlm-roberta (CA-IH)	8	$60.2_{2.3}$	**43.9**$_{1.2}$	$36.4_{1.8}$	**56.5**$_{1.5}$	$43.5_{2.0}$	**55.8**$_{2.9}$	**18.8**$_{2.2}$	**42.7**$_{1.7}$	$47.6_{2.6}$	**56.5**$_{3.2}$	$46.2_{2.2}$
xlm-roberta (SN)	8	**62.8**$_{0.6}$	$40.0_{0.9}$	$35.2_{3.0}$	$52.6_{0.6}$	**43.6**$_{0.6}$	$55.6_{2.3}$	$18.5_{0.9}$	$40.8_{2.8}$	**50.3**$_{1.2}$	$54.6_{3.6}$	$45.4_{2.0}$
xlm-roberta (SN-IH)	8	$59.2_{1.5}$	$41.5_{1.3}$	$33.8_{2.4}$	$53.4_{1.3}$	$43.2_{0.9}$	$51.8_{3.6}$	$17.2_{0.8}$	$41.4_{1.4}$	$50.2_{1.2}$	$52.6_{4.5}$	$44.4_{2.2}$
Char-SVM	64	$77.3_{0.8}$	$41.4_{0.8}$	$48.1_{2.9}$	$51.5_{0.7}$	$43.5_{0.8}$	$39.0_{0.8}$	$17.3_{0.4}$	$40.4_{1.0}$	$52.3_{0.8}$	$54.7_{0.9}$	$46.6_{1.2}$
xlm-roberta (CA)	64	**78.4**$_{1.1}$	**51.0**$_{1.6}$	$56.8_{1.6}$	**65.6**$_{0.8}$	$51.2_{1.5}$	**61.9**$_{1.1}$	$24.3_{1.7}$	**49.5**$_{0.7}$	$55.0_{0.7}$	$61.4_{2.0}$	$55.5_{1.3}$
xlm-roberta (CA-IH)	64	$78.3_{1.4}$	$50.8_{1.5}$	$57.2_{2.0}$	$64.3_{1.4}$	**51.3**$_{1.3}$	$61.6_{0.5}$	**24.6**$_{1.0}$	$48.4_{1.6}$	$56.0_{1.6}$	$60.7_{2.4}$	$55.3_{1.6}$
xlm-roberta (SN)	64	$77.4_{0.6}$	$49.6_{0.8}$	**59.3**$_{1.1}$	$58.8_{2.3}$	$49.7_{1.6}$	$58.3_{2.1}$	$23.6_{0.7}$	$47.3_{0.4}$	$56.0_{0.8}$	**61.8**$_{2.7}$	$54.2_{1.5}$
xlm-roberta (SN-IH)	64	$77.0_{0.9}$	$49.8_{0.9}$	$56.8_{0.6}$	$60.3_{1.3}$	$49.8_{1.4}$	$57.5_{1.8}$	$22.8_{0.8}$	$46.8_{0.3}$	**56.3**$_{1.1}$	$59.5_{2.7}$	$53.6_{1.3}$
Char-SVM	512	$85.0_{0.3}$	$48.2_{0.5}$	$48.1_{2.9}$	$59.0_{0.4}$	$50.4_{0.4}$	$46.0_{0.5}$	$23.0_{0.4}$	$46.4_{0.9}$	$64.7_{0.4}$	$63.8_{1.3}$	$53.5_{1.1}$
xlm-roberta (CA)	512	$84.7_{0.7}$	$56.3_{0.3}$	$56.5_{1.9}$	**68.5**$_{1.6}$	**58.6**$_{0.8}$	**62.7**$_{0.6}$	$29.2_{0.7}$	**53.0**$_{0.4}$	$65.9_{1.0}$	$67.9_{0.6}$	$60.3_{1.0}$
xlm-roberta (CA-IH)	512	**85.8**$_{1.3}$	**56.8**$_{0.7}$	$56.3_{1.8}$	$67.9_{1.4}$	$58.5_{0.5}$	$62.5_{1.3}$	$28.9_{1.0}$	$52.3_{1.4}$	$65.9_{0.5}$	**68.9**$_{1.2}$	$60.4_{1.2}$
xlm-roberta (SN)	512	$85.0_{0.6}$	$55.7_{0.4}$	**59.5**$_{1.8}$	$67.9_{0.5}$	**58.6**$_{0.4}$	$62.3_{0.8}$	**29.5**$_{0.4}$	$52.5_{0.8}$	**66.9**$_{0.1}$	$65.6_{1.1}$	$60.3_{0.8}$
xlm-roberta (SN-IH)	512	$84.9_{0.5}$	$56.1_{0.5}$	$57.6_{0.9}$	$67.8_{1.5}$	$58.3_{0.2}$	$61.3_{0.9}$	$29.1_{0.6}$	$52.4_{0.6}$	$66.8_{0.9}$	$68.3_{1.2}$	$60.3_{0.9}$

Table 3: Multi-lingual results for models based on roberta-xlm for cross attention (CA) and Siamese networks (SN). n denotes the number of training examples. Underlined results are significant. **Bold** font indicates maxima.

for the languages other than English, explains why quality is lower than for the English-only experiments. For the zero-shot scenario, all models outperform the random baseline on average, but with a smaller margin than for the English-only models. The FastText baseline performs comparable to CA on average (26.0 vs 27.2), while SN is ahead by a large margin (27.2 vs 32.4). The differences between models with hypotheses and identity hypothesis (IH) are smaller than for the English experiments.

Looking at the few-shot scenarios, we see that both models out-perform the Char-SVM by a large margin. In general, the results are closer than for the English experiments, as well as in the number of datasets with significant differences (only 2-4 of datasets). Similarly to English, we can conclude that at multilingual level, SN is more accurate in the zero-shot scenario whereas CA performs better in the few-shot one. However, for few-shot we see only small average differences (less than 1 point except for n=64).

5.2 Label Tuning

Table 4 shows a comparison of different fine-tuning approaches on the English datasets. Appendix H contains the multi-lingual results and gives a similar picture. We first compare Label Refinement (LR) as discussed in Chu et al. (2021) (see Section 3). Recall that this approach makes use of unlabeled data. We find that in the zero-shot scenario LR gives an average improvement of more than 2 points and significantly out-performing the baseline (mpnet) for 7 of the 11 datasets. When combining LR with labeled data as discussed in Chu et al. (2021) we find this to only give modest improvements over the zero-shot model (e.g., 54.0 (zero-shot) vs 55.8 (n=8)). Note that we apply LR to the untuned model, while Chu et al. (2021) proposed to apply it to a tuned model. However, we find that to only give small improvements over an already tuned model (mpnet (FT) vs. mpnet (FT+LR)). Also, in this work we are interested in approaches that do not change the initial model so that it can be shared between tasks to improve scalability. Label Tuning (LT) improves results as n grows and out-performs LR and the Char-SVM baseline from Table 2.

Comparing regular Fine-Tuning (FT) and BitFit, we find them to perform quite similarly both on average and on individual datasets, with only few exceptions, such as the performance difference on TREC for the n=8 setup. In comparison with FT and BitFit, LT is significantly out-performed on most datasets. The average difference in performance is around 5 points, which is comparable to using 8 times less training data.

Using the knowledge distillation approach discussed before (LT-DIST), we find that for 8 and 64 examples, most of the difference in performance can be recovered while still keeping the high scalability. For n=8, we only find a significant differ-

name	n	Yahoo	AG News	Unified	COLA	SUBJ	TREC	IMDB	SemEval	Yelp pol	Yelp full	Amazon	Mean
mpnet	0	$55.0_{0.2}$	$65.6_{0.4}$	$20.5_{0.3}$	$47.6_{1.4}$	$62.8_{0.9}$	$43.0_{2.1}$	$79.5_{0.2}$	**48.9**$_{0.3}$	$79.9_{0.2}$	$32.1_{0.2}$	$37.0_{0.7}$	$52.0_{0.9}$
mpnet (LR)	0	**59.1**$_{0.2}$	**73.8**$_{0.5}$	**20.9**$_{0.3}$	**47.7**$_{1.5}$	**68.7**$_{0.8}$	**48.2**$_{2.2}$	**80.0**$_{0.2}$	$46.3_{0.3}$	**80.5**$_{0.2}$	$28.6_{0.2}$	**39.8**$_{0.6}$	$54.0_{0.9}$
mpnet (BitFit)	8	$62.6_{0.7}$	$80.1_{1.5}$	**27.0**$_{1.2}$	$49.0_{0.9}$	$79.6_{3.0}$	$57.9_{2.0}$	**83.9**$_{0.9}$	$54.6_{2.8}$	$90.3_{1.9}$	$50.1_{1.4}$	$46.1_{1.2}$	$61.9_{1.7}$
mpnet (FT)	8	$63.5_{0.8}$	$83.3_{1.9}$	**27.0**$_{0.8}$	**49.7**$_{0.9}$	$83.1_{4.8}$	**70.8**$_{7.1}$	$82.6_{2.3}$	$54.8_{3.3}$	**90.6**$_{1.1}$	$50.5_{1.6}$	**46.8**$_{1.6}$	$63.9_{3.0}$
mpnet (FT+LR)	8	**63.9**$_{1.0}$	**83.6**$_{1.8}$	$26.3_{0.8}$	$49.1_{1.1}$	$84.5_{3.4}$	$68.9_{7.3}$	$83.6_{2.5}$	**56.9**$_{1.5}$	$90.5_{1.2}$	**51.1**$_{1.2}$	$46.7_{1.9}$	$64.1_{2.8}$
mpnet (LR)	8	$59.7_{0.3}$	$76.0_{0.6}$	$22.4_{0.4}$	$47.8_{0.5}$	$71.3_{1.4}$	$48.4_{2.7}$	$80.4_{0.3}$	$50.9_{2.0}$	$81.7_{1.5}$	$33.6_{3.8}$	$41.2_{1.5}$	$55.8_{1.7}$
mpnet (LT)	8	$59.4_{0.9}$	$78.7_{0.9}$	$23.2_{0.4}$	$48.7_{1.4}$	$81.9_{3.4}$	$52.5_{4.4}$	$77.7_{0.5}$	$45.2_{2.0}$	$85.1_{2.2}$	$41.5_{1.1}$	$41.9_{2.9}$	$57.8_{2.2}$
mpnet (LT-DIST)	8	$62.9_{0.7}$	$83.0_{1.9}$	$26.6_{0.9}$	$47.7_{3.0}$	**84.6**$_{3.4}$	$67.8_{6.4}$	$83.7_{0.6}$	$54.9_{2.2}$	$89.9_{1.4}$	$49.2_{1.0}$	$45.6_{2.1}$	$63.3_{2.7}$
mpnet (BitFit)	64	**67.6**$_{0.6}$	$86.9_{0.9}$	**30.3**$_{0.9}$	$51.3_{0.9}$	$93.7_{0.9}$	$82.1_{2.9}$	$85.7_{1.0}$	$60.8_{1.4}$	**92.1**$_{0.5}$	**54.9**$_{0.7}$	$51.8_{1.2}$	$68.8_{1.3}$
mpnet (FT)	64	$67.3_{0.5}$	$87.3_{1.2}$	$29.5_{0.4}$	$55.4_{1.2}$	**93.8**$_{0.5}$	**88.5**$_{2.6}$	$86.1_{1.2}$	**61.4**$_{3.0}$	$91.8_{0.3}$	$54.5_{0.4}$	$53.6_{1.6}$	$69.9_{1.5}$
mpnet (FT+LR)	64	$67.5_{0.4}$	**87.6**$_{0.8}$	$29.4_{0.3}$	**55.5**$_{0.9}$	$93.7_{0.5}$	$86.5_{3.4}$	**86.2**$_{0.4}$	$60.4_{2.1}$	$91.4_{0.6}$	$54.6_{0.8}$	**54.1**$_{1.6}$	$69.7_{1.4}$
mpnet (LR)	64	$59.9_{0.1}$	$76.6_{0.3}$	$22.7_{0.2}$	$47.8_{0.5}$	$71.6_{0.5}$	$51.1_{1.0}$	$80.4_{0.1}$	$52.0_{0.7}$	$82.1_{0.7}$	$29.8_{1.3}$	$42.0_{0.5}$	$56.0_{0.7}$
mpnet (LT)	64	$64.8_{0.3}$	$85.0_{0.6}$	$27.1_{0.6}$	$49.3_{1.2}$	$89.9_{0.5}$	$70.8_{2.8}$	$81.2_{1.0}$	$54.5_{2.7}$	$89.0_{0.6}$	$50.0_{0.7}$	$49.1_{1.6}$	$64.6_{1.4}$
mpnet (LT-DIST)	64	$67.0_{0.5}$	$86.9_{0.9}$	$28.8_{0.4}$	$52.2_{1.2}$	$92.5_{0.2}$	$86.5_{1.1}$	$84.6_{0.3}$	$60.2_{2.3}$	$91.2_{0.3}$	$53.7_{0.7}$	$52.7_{1.2}$	$68.7_{1.0}$
mpnet (BitFit)	512	**70.4**$_{0.2}$	$90.3_{0.2}$	$32.9_{0.2}$	$72.9_{1.3}$	**96.3**$_{0.2}$	$92.2_{0.6}$	**88.2**$_{0.2}$	**64.4**$_{0.8}$	**93.3**$_{0.2}$	**58.5**$_{0.2}$	$60.7_{0.3}$	$74.5_{0.5}$
mpnet (FT)	512	$69.3_{0.2}$	$90.7_{0.3}$	**33.0**$_{0.4}$	**74.5**$_{1.2}$	$96.0_{0.2}$	**95.4**$_{1.3}$	$87.7_{0.4}$	$64.1_{0.8}$	$93.2_{0.3}$	**58.5**$_{0.2}$	$60.8_{0.7}$	$74.8_{0.7}$
mpnet (FT+LR)	512	$69.5_{0.2}$	**90.8**$_{0.3}$	$32.6_{0.5}$	$74.2_{0.9}$	**96.3**$_{0.3}$	$95.0_{0.9}$	$88.0_{0.6}$	$63.3_{0.7}$	**93.3**$_{0.2}$	$58.4_{0.2}$	**61.3**$_{0.3}$	$74.8_{0.5}$
mpnet (LR)	512	$60.1_{0.1}$	$76.7_{0.2}$	$22.6_{0.1}$	$47.8_{0.3}$	$72.0_{0.2}$	$51.4_{0.3}$	$80.3_{0.0}$	$52.6_{0.2}$	$81.5_{0.2}$	$29.7_{0.3}$	$42.7_{0.2}$	$56.1_{0.2}$
mpnet (LT)	512	$68.0_{0.2}$	$88.0_{0.3}$	$29.1_{0.4}$	$55.2_{1.1}$	$92.6_{0.5}$	$86.2_{0.2}$	$84.3_{0.3}$	$59.8_{0.7}$	$91.0_{0.2}$	$53.7_{0.3}$	$54.9_{0.5}$	$69.3_{0.5}$
mpnet (LT-DIST)	512	$68.7_{0.2}$	$88.9_{0.2}$	$30.8_{0.3}$	$58.6_{1.1}$	$93.7_{0.2}$	$89.4_{0.5}$	$85.5_{0.2}$	$61.3_{0.5}$	$91.7_{0.1}$	$55.8_{0.2}$	$57.0_{0.6}$	$71.0_{0.5}$

Table 4: English results for Siamese models based on MPNET and trained on NLI and paraphrasing datasets. Comparing fine-tuning (FT), label tuning (LT), label tuning with distillation (LT-DIST), and label refinement (LR). Results are grouped by the number of training examples (n). Underlined results are significant. **Bold** font indicates maxima.

name	2-3	4-6	10
W2V	192.90	195.82	208.40
mpnet-base (CA)	5.12	2.22	1.15
mpnet-base (SN)	26.08	18.30	18.85

Table 5: Processing speed in *thousand tokens/second*. We show the results grouped by the size of the label set. Calculated on the English test sets.

length	1-22	22-44	44-86	86-160	>160
SN	39.8	44.6	42.5	34.5	36.4
CA	36.7	41.8	44.0	35.2	40.3

Table 6: Average macro F1 score for sets of different token length measured across all test sets for n=0.

task	emotions		reviews		sentiment	
negation	no	yes	no	yes	no	yes
SN	23.0	14.3	49.0	44.4	37.3	45.1
CA	22.4	16.8	48.2	47.0	32.2	37.4

Table 7: Average macro F1 score for sets with and without a negation marker present. Measured across all test sets for n=0.

ence to mpnet (FT) for Yelp full. Recall that the distillation is performed on up to 10,000 unlabeled examples from the training set.

6 Analysis

We analyze the performance of the Cross Attention (CA) and Siamese Network-based (SN) models. Unless otherwise noted, the analysis was run over all datasets and languages. Table 5, gives a comparison of the processing speed of different models. Details on the hardware used is given in Appendix F. As expected, the performance of the cross attention model halves when the label size doubles. The performance of the Siamese network is independent of the number of labels. This shows that Siamese Networks have a huge advantage at inference time – especially for tasks with many labels.

Table 6 shows the average F1 scores for different token lengths. To this end the data was grouped in bins of roughly equal size. SN has an advantage for shorter sequences ($\leq$ 44 tokens), while CA performs better for longer texts ($>$ 160 tokens).

Table 7 shows an analysis based on whether the text does or does not contain negation markers. We used an in-house list of 23 phrases for German and Spanish and 126 for English. For emotion detection and review tasks, both models perform better on the subset without negations. However, while SN outperforms CA on the data without negations, CA performs better on the data with negations. The same trend does not hold for the sentiment datasets. These are based on Twitter and thus contain shorter and simpler sentences. For the sentiment datasets based on Twitter we also found that both models struggle to predict the neutral class. CA classifies

almost everything neutral tweet as positive or negative. SN predicts the neutral class regularly but still with a relative high error rate. Appendix E contains further analysis showing that label set size, language and task do not have a visible effect on the difference in accuracy of the two models.

7 Conclusion

We have shown that Cross Attention (CA) and Siamese Networks (SN) for zero-shot and few-shot text classification give comparable results across a diverse set of tasks and multiple languages. The inference cost of SNs is low as label embeddings can be pre-computed and, in contrast to CA, does not scale with the number of labels. We also showed that tuning only these label embeddings (Label Tuning (LT)) is an interesting alternative to regular Fine-Tuning (FT). LT gets close to FT performance when combined with knowledge distillation and when the number of training samples is low, i.e., for realistic few-shot learning. This is relevant for production scenarios, as it allows to share the same model among tasks. However, it will require 60 times more memory to add a new task: For a 418 MB mpnet-base model, BitFit affects 470 kB of the parameters. LT applied to a task with 10 labels and using a embedding dimension of 768 requires 7.5 kB. The main disadvantage of BitFit, however, is that the weight sharing it requires is much harder to implement, especially in highly optimized environments such as NVIDIA Triton. Therefore we think that LT is an interesting alternative for fast and scalable few-shot learning.

Acknowledgements

We would like to thank Francisco Rangel and the entire Symanto Research Team for early discussions, feedback and suggestions. We would also like to thank the anonymous Reviewers. The authors gratefully acknowledge the support of the Pro^{2}Haters - Proactive Profiling of Hate Speech Spreaders (CDTi IDI-20210776), XAI-DisInfodemics: eXplainable AI for disinformation and conspiracy detection during infodemics (MICIN PLEC2021-007681), and DETEMP - Early Detection of Depression Detection in Social Media (IVACE IMINOD/2021/72) R&D grants.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics.

Cecilia Ovesdotter Alm and Richard Sproat. 2005. Perceptions of emotions in expressive storytelling. In *Ninth European Conference on Speech Communication and Technology*.

Ebba Cecilia Ovesdotter Alm. 2008. *Affect in* text and speech*. University of Illinois at Urbana-Champaign.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv*, abs/2106.10199.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O'Reilly Media, Inc.

Timo Block. 2019. Ten thousand german news articles dataset. https://tblock.github.io/10kGNAD/. Accessed: 2021-08-25.

Laura Ana Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119. Association for Computational Linguistics.

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.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Zewei Chu, Karl Stratos, and Kevin Gimpel. 2021. Unsupervised label refinement improves dataless text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4165–4178, Online. Association for Computational Linguistics.

Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli. 2017. A Twitter corpus and benchmark resources for German sentiment analysis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 45–51, Valencia, Spain. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Crowdflower. 2016. The emotion in text, published by crowdflower.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *CICLing (2)*, pages 152–165.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Antonio Gulli. 2005. AG's corpus of news articles. `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`. Accessed: 2021-07-08.

Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. 2020. Task-aware representation of sentences for generic text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3202–3213, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. 2014. Distilling the knowledge in a neural network. In *The NIPS 2014 Learning Semantics Workshop*.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 646–651. AAAI Press.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

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

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

V. Liu, C. Banea, and R. Mihalcea. 2007. Grounded emotions. In *International Conference on Affective Computing and Intelligent Interaction (ACII 2017)*, San Antonio, Texas.

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

Robert L. Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Advances in Neural Information Processing Systems*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Saif Mohammad. 2012. #Emotional Tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada. Association for Computational Linguistics.

Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and sentiment in tweets. *ACM Transactions on Internet Technology (TOIT)*, 17(3):1–23.

Saif M Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. 2015. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California. Association for Computational Linguistics.

María Navas-Loro, Víctor Rodríguez-Doncel, Idafen Santana-Perez, and Alberto Sánchez. 2017. Spanish corpus for sentiment analysis towards brands. In *Speech and Computer*, pages 680–689, Cham. Springer International Publishing.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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*. Association for Computational Linguistics.

K. Scherer and H. G. Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66 2:310–28.

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

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Enrica Troiano, Sebastian Padó, and Roman Klinger. 2019. Crowdsourcing and validating event-focused emotion corpora for German and English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4005–4011, Florence, Italy. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

David Vilares and Carlos Gómez-Rodríguez. 2019. HEAD-QA: A healthcare dataset for complex reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 960–966, Florence, Italy. Association for Computational Linguistics.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *ArXiv*, abs/2104.14690.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. 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.

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

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Unified Emotions

Unified Emotions is a meta-dataset comprised of the following datasets: DailyDialog (Li et al., 2017), CrowdFlower (Crowdflower, 2016), TEC (Mohammad, 2012), Tales (Alm et al., 2005; Alm and Sproat, 2005; Alm, 2008), ISEAR (Scherer and Wallbott, 1994), Emoint (Mohammad et al., 2017), ElectoralTweets (Mohammad et al., 2015), GroundedEmotions (Liu et al., 2007) and EmotionCause (Ghazi et al., 2015).

B Hypotheses

Table 9 lists all the hypothesis patterns used in our experiments.

C Paraphrase datasets

`paraphrase-mpnet-base-v2` has been trained on these datasets: SNLI, MNLI, sentence-compression, SimpleWiki, altlex, msmarco-triplets, quora_duplicates, coco_captions, yahoo_answers_title_question, S2ORC_citation_pairs, stackexchange_duplicate_questions and wiki-atomic-edits. Details on these dataset are provided here.

D Hyperparameters

For the label tuning experiments we used the following hyper-parameters:

- learning rate $\in \{0.01, 0.1\}$
- number of epochs $\in \{1000, 2000\}$
- regularizer coefficient $\in \{0.01, 0.1\}$
- dropout rate $\in \{0.01, 0.1\}$

E Additional Analysis

The following table shows the F1-score breakdown by hypothesis length. One could think that the CA model performs better for longer hypothesis but this cannot be observed. Potentially because all hypotheses are relatively short.

name	3-5	5-7	>7
SN	42.2	32.9	30.3
CA	41.4	30.1	25.2

Table 10: Average macro F1 score by length of the reference hypothesis, averaged over all test sets for n=0.

For completeness, we also add similar breakdowns by task type, label set size, and language. None of them indicate an effect on the difference between SN and CA model performance.

name	2-3	4-6	>6
SN	51.1	36.7	34.7
CA	52.1	32.0	31.2

Table 11: Average macro F1 score by label set size, averaged over all test sets for n=0.

name	emotions	other	reviews	sentiment	topic
SN	21.8	40.4	46.4	39.0	48.3
CA	22.2	34.7	47.8	33.7	44.4

Table 12: Average macro F1 score by task, averaged over all test sets for n=0.

name	de	en	es
SN	33.3	47.7	31.8
CA	27.0	46.8	30.1

Table 13: Average macro F1 score by language, averaged over all test sets for n=0.

F Computing Requirements

All experiments were run on a system with an AMD Ryzen Threadripper 1950X CPU and a Nvidia GeForce GTX 1080 Ti GPU. Most of the computing time was spent training the NLI models used in our experiments. Training the CA models took approx. 20h while training the SN models took approx. 10h.

G NLI Training sets

name	examples
SNLI (Bowman et al., 2015)	569,033
MNLI (Williams et al., 2018)	412,349
ANLI (Nie et al., 2020)	169,246
XNLI (Conneau et al., 2018)	112,500

Table 14: Sizes of NLI traininig sets. SNLI, MNLI and ANLI are English only. XNLI contains 15 languages with 7,500 examples per language.

H Multilingual Label Tuning Results

Table 8 multilingual results for label tuninig and fine-tuning.

language		German				English			Spanish			
name	n	GNAD	Amazon	deISEAR	sb10k	Amazon	SemEval	Unified	Amazon	HeadQA	SAB s	Mean
random	0	11.1	20.0	14.3	33.3	20.0	33.3	10.0	20.0	16.7	33.3	21.2
FastText	0	$17.3_{1.0}$	$15.4_{0.5}$	$22.2_{2.1}$	$31.5_{1.5}$	$18.6_{0.5}$	$43.8_{0.4}$	$11.8_{0.3}$	$19.7_{0.5}$	$45.0_{0.9}$	$\underline{\mathbf{35.0}}_{2.2}$	$26.0_{1.2}$
xlm-roberta	0	$\underline{\mathbf{37.8}}_{1.1}$	$\underline{\mathbf{28.4}}_{0.7}$	$\underline{\mathbf{43.1}}_{2.7}$	$\underline{\mathbf{46.6}}_{1.3}$	$\underline{\mathbf{35.4}}_{0.7}$	$\underline{\mathbf{50.5}}_{0.4}$	$\underline{\mathbf{21.3}}_{0.3}$	$\underline{\mathbf{32.8}}_{0.6}$	$\underline{\mathbf{50.6}}_{0.9}$	$31.6_{2.0}$	$37.8_{1.3}$
Char-SVM	8	$56.1_{2.8}$	$30.5_{2.2}$	$29.4_{1.6}$	$45.4_{2.5}$	$30.0_{1.6}$	$33.6_{1.1}$	$12.2_{1.1}$	$30.8_{1.2}$	$36.3_{2.6}$	$50.6_{5.3}$	$35.5_{2.5}$
xlm-roberta (FT)	8	$\underline{66.3}_{3.7}$	$\underline{\mathbf{45.1}}_{0.9}$	$\underline{\mathbf{56.6}}_{2.1}$	$\underline{\mathbf{55.9}}_{2.6}$	$\underline{45.2}_{1.2}$	$\underline{\mathbf{55.7}}_{3.8}$	$\underline{\mathbf{25.4}}_{0.7}$	$\underline{\mathbf{42.5}}_{1.1}$	$\underline{\mathbf{55.0}}_{2.3}$	$\underline{58.1}_{5.2}$	$50.6_{2.8}$
xlm-roberta (LT)	8	$\underline{64.6}_{1.2}$	$42.1_{1.5}$	$50.6_{2.4}$	$50.2_{1.8}$	$41.7_{2.0}$	$46.5_{2.7}$	$23.0_{0.4}$	$40.4_{1.3}$	$\underline{53.7}_{2.9}$	$\underline{52.2}_{4.8}$	$46.5_{2.4}$
xlm-roberta (LT-DIST)	8	$\underline{\mathbf{67.0}}_{3.2}$	$\underline{44.3}_{0.8}$	$\underline{53.2}_{3.0}$	$\underline{55.8}_{2.0}$	$\underline{\mathbf{45.4}}_{1.6}$	$\underline{53.1}_{3.3}$	$\underline{25.3}_{0.6}$	$\underline{41.7}_{1.4}$	$\underline{54.6}_{2.3}$	$\underline{\mathbf{59.4}}_{4.2}$	$50.0_{2.5}$
Char-SVM	64	$77.3_{0.8}$	$41.4_{0.8}$	$48.1_{2.9}$	$51.5_{0.7}$	$43.5_{0.8}$	$39.0_{0.8}$	$17.3_{0.4}$	$40.4_{1.0}$	$52.3_{0.8}$	$54.7_{0.9}$	$46.6_{1.2}$
xlm-roberta (FT)	64	$\underline{\mathbf{79.7}}_{0.7}$	$\underline{\mathbf{51.5}}_{1.0}$	$\underline{\mathbf{67.7}}_{0.9}$	$\underline{\mathbf{63.0}}_{0.9}$	$\underline{\mathbf{53.1}}_{1.9}$	$\underline{\mathbf{61.0}}_{1.6}$	$\underline{\mathbf{28.1}}_{0.2}$	$\underline{\mathbf{49.4}}_{0.3}$	$\underline{\mathbf{60.5}}_{1.0}$	$\underline{64.9}_{1.8}$	$57.9_{1.2}$
xlm-roberta (LT)	64	$76.9_{0.6}$	$48.4_{0.6}$	$62.6_{0.9}$	$59.1_{0.6}$	$49.1_{1.6}$	$54.2_{1.9}$	$26.9_{0.7}$	$48.7_{0.4}$	$\underline{59.3}_{0.8}$	$\underline{61.8}_{3.1}$	$54.7_{1.4}$
xlm-roberta (LT-DIST)	64	$\underline{78.9}_{0.5}$	$\underline{50.0}_{1.1}$	$64.7_{0.3}$	$\underline{62.5}_{0.9}$	$\underline{51.7}_{1.3}$	$\underline{59.5}_{1.0}$	$\underline{27.6}_{0.4}$	$\underline{48.9}_{0.7}$	$\underline{59.3}_{0.9}$	$\underline{\mathbf{65.4}}_{1.8}$	$56.9_{1.0}$
Char-SVM	512	$\underline{85.0}_{0.3}$	$48.2_{0.5}$	$48.1_{2.9}$	$59.0_{0.4}$	$50.4_{0.4}$	$46.0_{0.5}$	$23.0_{0.4}$	$46.4_{0.9}$	$64.7_{0.4}$	$63.8_{1.3}$	$53.5_{1.1}$
xlm-roberta (FT)	512	$\underline{\mathbf{85.4}}_{0.6}$	$\underline{\mathbf{57.2}}_{0.7}$	$\underline{\mathbf{67.8}}_{1.2}$	$\underline{\mathbf{68.6}}_{0.9}$	$\underline{\mathbf{58.8}}_{0.4}$	$\underline{\mathbf{64.7}}_{0.7}$	$\underline{\mathbf{32.1}}_{0.3}$	$\underline{\mathbf{53.3}}_{0.6}$	$\underline{\mathbf{68.8}}_{0.5}$	$\underline{\mathbf{69.7}}_{0.5}$	$62.6_{0.7}$
xlm-roberta (LT)	512	$80.8_{0.6}$	$52.5_{0.7}$	$62.6_{0.8}$	$63.3_{0.9}$	$54.3_{0.3}$	$60.6_{0.7}$	$28.9_{0.4}$	$51.4_{0.4}$	$62.9_{0.3}$	$66.8_{0.4}$	$58.4_{0.6}$
xlm-roberta (LT-DIST)	512	$80.7_{0.4}$	$54.1_{0.3}$	$64.6_{0.2}$	$66.0_{1.3}$	$55.6_{0.3}$	$62.9_{1.0}$	$30.5_{0.4}$	$52.4_{0.2}$	$63.1_{0.4}$	$68.7_{0.6}$	$59.9_{0.6}$

Table 8: Multi-lingual results for Siamese models based on paraphrase-multilingual-mpnet-base-v2, comparing fine-tuning (FT), label tuning (LT) and label tuning with distillation (LT-DIST). Results are grouped by the number of training examples (n). Underlined results are significant. **Bold** font indicates maxima.

dataset	type	lang.	pattern
Unified	Emotions	en	This person feels {anger, disgust, feat, guilt, joy, love, sadness, shame, surprise}. This person doesn't feel any particular emotion.
deISEAR		de	Diese Person empfindet {Schuld, Wut, Ekel, Angst, Freude, Scham, Traurigkeit}.
AG News	Topic	en	It is {business, science, sports, world} news.
GNAD		de	Das ist ein Artikel aus der Rubrik {Web, Panorama, International, Wirtschaft, Sport, Inland, Etat, Wissenschaft, Kultur}.
HeadQA		es	Está relacionado con la {medicina, enfermería, química, biología, psicología, farmacología}.
Yahoo		en	It is related with {business & finance,computers & internet, education & reference, entertainment & music, family & relationships, health, politics & government, science & mathematics, society & culture, sports}.
Amazon	Review	en	This product is {terrible, bad, okay, good, excellent}.
		de	Dieses Produkt ist {furchtbar, schlecht, ok, gut, exzellent}.
		es	Este producto es {terrible, mal, regular, bien, excelente}.
IMDB, Yelp (2)		en	It was {terrible, great}.
Yelp (5)			It was {terrible, bad, okay, good, great}.
SemEval	Sentiment	en	This person expresses a {negative, neutral, positive} feeling.
sb10k		de	Diese Person drückt ein {negativ, neutral, positiv}es Gefühl aus.
SAB		es	Esta persona expresa un sentimiento {negativo, neutro, positivo}.
COLA	Acceptability	en	It is {correct, incorrect}.
SUBJ	Subjectivity	en	It is {objective, subjective}.
TREC	Question Type	en	It is {expression, description, entity, human, location, number}.

Table 9: Hypotheses patterns used.