DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Guided Research

Injecting Knowledge into Sentence Embedding Models for Information Retrieval using Adapters

Dennis Schneider

Abstract

This work investigates approaches of injecting structured knowledge from knowledge graphs into transformer language models and evaluates their effect on State-of-the-Art Sentence Embedding Models for semantic textual similarity (STS) tasks. In order to do this, this work focuses on using Adapter-networks to inject this knowledge.

To ensure a useful contribution to the research-field of Adapters, the results are kept comparably to current works and are evaluated on both general and domain-specific datasets.

The usefulness of Adapters for STS-tasks is evaluated both in supervised and unsupervised environments and an alternative process of Adapter-training is evaluated.

Adapters are found to perform well on STS-tasks, achieving similar performance to a full training of the entire model and to be able to provide a cheap method of domain-specific finetuning of general world STS-trained models.

0.1 Motivation

Semantic Textual Similarity (STS) describes the ability to attribute a similarity score to sentences with a similar meaning. This is an essential problem in natural language processing, as it has a significant impact on several downstream tasks such as information retrieval, text classification or question answering [1].

Sentence embeddings map complex sentences to a high-dimensional space in order to achieve similarity comparison and information retrieval from sentences. While transformer language models (TLM) can be used to generate semantic sentence embeddings by using the SBERT-Approach [2], they fail to capture rich factual knowledge [3]. The effect of injecting factual knowledge from knowledge graphs is thus expected to further improve the generated embeddings.

While a TLM is pre-trained on tasks like Language Modelling, finetuning these models (e.g. BERT) is a resource-intensive task [4]. To alleviate this problem, Adapters have been introduced to finetune a language modelling model while only training a small set of parameters.

This work looks into the effect of various Adapter-architectures on the performance of Sentence Embedding Models, with the goal to incorporate structural knowledge from knowledge graphs into sentence embeddings.

0.2 Related Works

The Houlsby Adapter [4] was introduced in 2019, and along with the Pfeiffer Adapter [5] belongs to the most commonly used Adapter models. In previous papers, Adapter models have sucessfully been used for multiple tasks, e.g. Language Translation [6] and Speech Recognition [7]. This work however is the first one to look into the applicability of Adapter models to STS-tasks and give an overview of their domain-adaptability.

Two examples of works looking into gathering knowledge from Knowledge Graphs are the K-Bert approach [8] and the KGLM approach [9]. Similar to this work, K-Bert uses an Adapter-like addition to a Bert model to inject domain-specific knowledge into sentence embeddings. Furthermore does this work build upon the works of SimCSE [10] as the current State of the Art when it comes to Sentence Embeddings. Most importantly, the supervised SimCSE approach uses a contrastive approach distinguishing similar and dissimilar sentences within a dataset. This contrastive training approach is leveraged in this work, as contrastive training is suitable when trying to extract information from a Knowledge Graph [10]. Another popular training methodology has been introduced by the SBERT-Paper [2], using a siamese BERT model which is trained to correctly classify either similar or dissimilar sentences.

Similar to this work, the SPECTER paper [11] introduces means to learn Document Embeddings from a Knowledge Graph, or more precisely a citation graph. As already introduced, it uses a contrastive training objective to do so, comparably to the SimCSE paper.

0.3 Datasets

In the following, the used datasets are introduced, motivated and their generation process is described, in order to guarantee reproducibility of the results. For the upcoming experiments, both a common-world dataset is needed, as well as domain-specific datasets.

0.3.1 TREx-rc Dataset

The TREx-rc dataset is a shortened version of the TREx dataset [12]. It provides sentences from Wikipedia abstracts aligned with knowledge triples (source object, relation, target object). In the case of contrastive STS-training, a notion of sentence-similarity is needed to provide a sentence deemed similar and a sentence deemed dissimilar to a certain anchor sentence. Thus, sentences are grouped by relations and two sentences having the same targets are deemed similar while sentences of different targets are seen as dissimilar. For clarification, consult the following example:

Yelena Isinbayeva and Bob Richards are the only two athletes to win two Olympic pole vault titles, and also the only two athletes to win more than two Olympic medals in the discipline.

Buisson represented France at the 2008 Summer Olympics in Beijing, where she successfully cleared a height of 4.15 metres in the women's pole vault, an event which was later dominated by world-record holder Yelena Isinbayeva of Russia.

Four days later, El Guerrouj outsprinted 10000 metres gold medalist Kenenisa Bekele to take the 5000 metres gold medal and never competed internationally again, officially retiring in 2006.

It can easily be seen, that the first two sentences are very similar, both being about Olympic pole vaulters. While the third, negative example is similar enough to be comparable to the first two, it is important to distinguish the fact that this sentence is about a sprinter, thus being dissimilar to the first two.

Such a dataset processing enables the model to distinguish the important parts of sentences and ensures an understanding of the actually important facts of sentences.

0.3.2 SciDocs Dataset

The SciDocs dataset consists of scientific papers and their citation information. For usage in this work, the abstracts of related and unrelated papers are of interest. More precisely, a sample consists of a paper's title, a separator and the abstract of the paper. Since the BERT-architecture only allows for 512 tokens, the samples are cut off at the respective length. This same approach was already taken in the SPECTER paper [11] which was published alongside the SciDocs dataset and is thus seen as a reasonable choice, especially given the gist of the paper being mostly described in the title and in the beginning of the abstract.

For a certain anchor sample a positive sample is taken directly from a referenced paper. The

Base Model	Parameters	Parameters Houlsby	Parameters Pfeiffer	Parameters K-Adapter
Bert-base	110M	4M	10M	47M
Roberta-large	355M	6M	12M	47M

Table 1: Amount of trainable Parameters for different base models and Adapter architectures

negative sample is once again deemed to be similar to the previous two papers, however less related than the positive sample. To ensure comparability, the same approach as in the SPECTER paper is used, which defines a negative sample as a paper which was referenced by the positive paper but not by the anchor paper itself.

This approach is reasonable as all abstracts are about the same topic but the positive sample is more related to the anchor paper than the negative one. This also justifies the very similar approach used for the TREx-rc dataset.

0.3.3 AskUbuntu Dataset

The AskUbuntu dataset [13] is another domain-specific dataset. It already consists of sentence-pairs which are deemed similar, anchor- and positive samples are thus easily found. Since the dataset inherently consists of sentences about a similar topic, being the operating system Ubuntu, negative sentences can easily be retrieved by sampling different sentences. The dataset is convenient for this work, as it is in the technical domain, similarly to the SciDocs dataset, however differs significantly in its use of language, since it is closer to everyday language which differs quite significantly from the technical language used in papers. Thus, both domain-specific datasets show the technical domain from different perspectives and are thus comparable.

0.4 Adapter Architectures

Adapters provide an efficient method of finetuning an existing model to a certain task or dataset [4]. In general they enable to further tune an already trained model to an extended set of data or a different task without overwriting the already trained weights. Thus knowledge from multiple sources can be combined and exchanged. This in general is achieved by adding intermediate layers within a base model which are initialized to the identity function, thus not changing the base models behaviour [4, 5]. Freezing the base models parameters ensures keeping the knowledge already learned in the base model.

As this work tries to give a holistic view on STS-Adapters in multiple use-cases, three different Adapter-architectures are compared. The Houlsby- and the Pfeiffer-Adapters are already well-established architectures and widely used in the field. Since they are both very similar in their core idea, the K-Adapter is also used for comparison following an entirely different structure.

For reference, the table 1 depicts the amount of parameters per Adapter model compared to the base model, highlighting the efficient nature of Adapters.

0.4.1 Houlsby-Adapter

The Houlsby Adapter is the first Adapter proposed by Houlsby et al. [4] and was mostly developed with a focus on circumventing the catastrophical forgetting problem, which originates in the issue that when training a model on a certain objective, weights previously trained on a different objective are overwritten. Thus, the learning procedure of freezing a base model and injecting trainable layers, which are initialized to the identity was developed. In consequence, multiple downstream tasks could be trained based on a certain base model, while only a small amount of intermediate weights are added per task.
Most notable was the discovery that performance stayed mostly the same as compared to training an entire model while only training a very small subset of parameters.
The model is classified as a bottleneck Adapter, meaning an architecture which is located in between the transformer layers of the base model and whose layers transform its input into a very low-dimensional representation and upsampling it again to the same dimension in the output. Thus a parameter-efficient lower-dimensional representation is generated while most information is kept.

0.4.2 Pfeiffer-Adapter

The Pfeiffer Adapter [5] was developed in 2020 with a focus on multi-task learning. In contrast to the already introduced Houlsby-Adapter, it introduced a way to merge multiple trained adapters, and thus merge the knowledge from multiple tasks.
In this work, however, this multitask learning capability is not needed, and it is only used in the single-task mode. Due to its popularity the Pfeiffer Adapter is still a relevant architecture to compare its performance.

0.4.3 K-Adapter

The K-Adapter [3] uses an architecture, that in contrast to the previously introduced models works as an external plug-in. An Adapter-Layer uses the output of an intermediate transformer layer in the base model and concatenates it with the output of a previous Adapter layer, if any. In the end, the final Adapter-Layers output is concatenated with the base models output and transformed into the correct output dimension with a simple Dense Layer.
Still, each Adapter layer is in structure very similar to the previously proposed models, architectures, again being a Bottleneck Layer. The important new feature of the K-Adapter is the possibility to combine multiple external K-Adapters, whose outputs are simply concatenated in the end to retrieve the output of the entire model.
This adapter is especially interesting due to its vastly different structure and its random initialization, compared to the identity-initialization of the two formerly introduced Adapters.

0.5 Suitability of Adapters on STS-Tasks

While Adapters have successfully been employed for multiple tasks, there is no record of Adapters being evaluated on Semantic Textual Similarity (STS)-tasks. This is a particularly interesting field of usage for Adapters, since STS-trained models can be used in a variety of downstream tasks [14]. The ability to simply plug in an adapter into an existing model and enhancing such downstream-tasks motivates the following experiment. It showcases the usability of Adapters for general-world knowledge STS-tasks while only training the efficiently low amount of parameters of the Adapter, thus lowering the effort to train and make use of STS-models.

0.5.1 Methodology

Since STS-tasks are used to evaluate the semantic similarity of sentences and short texts, the model has to learn a notion of similarity from a collection of sentences. The current State-of-the-art is achieved by SimCSE [10], which uses a contrastive approach to distinguish similar from less similar sentences.

A positive sample x_i^+ and negative sample x_i^- for a certain anchor sentence x_i results in the following contrastive training objective, using their respective embeddings h_i^+, h_i^- and h_i:

$$L = -\log \frac{e^{sim(h_i,h_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(h_i,h_j^+)/\tau} + e^{sim(h_i,h_j^-)/\tau}} \tag{0.1}$$

The formula also employs a temperature scaling factor τ, which is empirically set to 0.05. The preprocessing of the used datasets has already been described in section 0.3. As a base model, the roberta-large model is used, just as in similar papers [3]. The Adapter-models are finetuned for 5 epochs on the TREx-rc dataset using the above mentioned approach, and a learning rate of $1e^{-5}$.
To put the results into context, they are compared to the performance of the fully finetuned roberta-large model on the TREx-rc dataset.

0.5.2 Results

In figure 2, the finetuned models are evaluated against 7 STS-tasks, comprised of the STS 2012-2016 [15, 16, 17, 18, 14], STS Benchmark [19] and SICK-Relatedness [20] datasets, with the evaluation score being the average Spearman correlations achieved in all datasets.
These datasets are also used in the evaluation of the K-Adapter paper [3] and are thus reasonable to get a full evaluation of the model performances. Although far less parameters are trained (cf. 1), the results are very similar to that of the finetuned roberta-large model. This result is especially interesting due to the lower compute-power used to get very comparable results. The Pfeiffer-Adapter performs the best out of all the Adapter-architectures and as the only architecture manages to be on-par with the finetuned base model. On closer inspection of the results, however, the performance difference to the finetuned base model

Model	STS12	STS13	STS14	STS15	STS16	STS-B.	SICK-R.	Avg.
Houlsby-Adapter	76.69	86.87	82.18	86.30	84.14	86.87	79.59	83.23
Pfeiffer-Adapter	77.93	87.00	82.60	87.31	83.50	86.74	80.92	83.71
K-Adapter	76.00	86.93	81.28	86.50	83.76	86.23	80.08	82.97
Finetuned roberta-large	77.87	87.24	82.56	87.17	84.62	86.26	79.93	83.68

Table 2: Evaluation-scores for different architectures after finetuning on a common world STS-task.

varies between tasks and the Pfeiffer Adapter is outperformed considerably on the STS16 task while it outperforms on the SICK-Relatedness task.

0.5.3 Discussion

While the results are looking very promising, having achieved very similar performance compared to the finetuned base model, the performance on single tasks varies a lot, with no method clearly outperforming on all tasks. This phenomenon can be explained with the different nature of the tasks, with the STS16 task employing news-article headlines and Wikipedia data [14], similar to the TREx-rc dataset while the STS12 task focuses more on machine translation evaluation corpuses from far more diverse sources [15].

In the end, the Adapter-architectures did not outperform the finetuned base model and produced slightly worse results while the performance-offset varied across tasks.

0.6 Fine-tuning methods of Adapters

Adapters have been introduced as methods to finetune a certain model to a specific task or new data. In order to accomplish this, the approach of freezing the base model's weights and and only finetuning the Adapter on the new task at hand has initially been proposed [4, 5] and since an Adapter uses far less parameters than the entire base model, an efficient method of pretraining was introduced.

As already introduced in the preceding experiment, Adapters achieve a very similar performance to a model which was fully trained on the same task. This however motivates the question, if Adapter-induced models can further benefit from adding an additional phase to the training procedure.

After finetuning the Adapter, the entire base model is unfrozen and trained in conjunction with the Adapter on the task. This answers the question whether the remaining performance gap can be alleviated by providing more parameters to the Adapters.

0.6.1 Methodology

Using a common world dataset, the different Adapter architectures are evaluated after the classical training procedure and additionally after the finetuning of the entire model. The

Adapter Architecture	Pretraining Phase	Basemodel refining phase
Finetuned Houlsby-Adapter	83.23	83.31
Finetuned Pfeiffer-Adapter	83.71	83.70
Finetuned K-Adapter	82.97	83.11

Table 3: The evaluation scores for the Adapter architectures after the standard finetuning process and after the appended phase of unfreezing the basemodel's weights.

learning rate is kept at $1e^{-5}$ and the training is first conducted for 5 epochs. In the newly introduced step, the training is commenced for another 3 epochs.

As a base model, the roberta-large model is used, which extends on the results of the previous experiment. The Adapter-architectures are trained on the TREx-rc dataset, after which the base model is unfrozen as well and in the newly introduced step trained in conjunction with the Adapter.

0.6.2 Results

The following table 3 shows the evaluation-scores after the different training-phases. Interestingly, the phase of unfreezing the base model does not seem to further enhance the results as across the Adapter-architectures, no significant performance-gains can be seen.

0.6.3 Discussion

Unfreezing the base model and training the entire model on the task would be expected to increase performance, as a higher amount of parameters is introduced for training. It was found that this is not the case for multiple Adapter architectures. Thus, the amount of parameters within an Adapter are no bottleneck for its learning capabilities, and the performance of Adapters can mostly be attributed by their structure and the idea of spreading the Adapter-weights across the Transformer model.

This finding is very interesting, as it highlights the ability of an Adapter model to achieve results very competitive with State-of-the-Art models such as Bert and Roberta while using only a small fraction of parameters. However, the interpretation of this result might pose a more difficult task, as the possible performance-gain is limited by the fully trained model, as seen in section 0.5. Thus, a performance-gain which can be interpreted as significant might not be possible.

0.7 Domain-Adaptation on STS-Tasks

While a lot of general world knowledge-graphs exist for training an STS-model, domain-specific data is much more scarce. Thus, this field would highly benefit from a parameter-efficient and in consequence data-efficient way to train on domain-specific data. The following section analyzes the performance of an STS-model which is trained on a general world dataset and is finetuned to multiple different domains by the Adapter-model alone.

0.7.1 Methodology

To test the Domain-Adaptability of STS-Adapters, a generic STS-base model is enhanced with a domain-specific Adapter. In order to make these results comparable to the related SPECTER paper [11], a base model of same parameter-size is used. Thus, the princeton-nlp/sup_simcse_bert_base model from the SimCSE paper [10] is used as a base model as it is already pretrained on sentence similarity tasks using the TREx-rc dataset.
The 3 adapter models are finetuned on 2 domain-specific datasets, and their performance is compared to the initial performance of the general base model as well as the finetuned base model. The domain-specific datasets are divided into a training- and a test-split with a 90%/10%-split. To ensure a general discussion about the usefulness of the architectures for the task, the default values for all adapters are kept and are not tuned to each dataset.
Additionally, a different loss-function is evaluated, in addition to the already introduced loss function from the SimCSE-paper [10].
This loss-function was introduced in the SPECTER-paper [11] and is added here for comparability reasons. It is based on the hinge-loss, however used in a contrastive context, resulting in a similar loss-function to the SimCSE-loss. In this case, the distance-function $d(.,.)$ describes the $L2$ distance and the loss margin hyperparameter m was set to 1 as described in the paper.

$$L = \max\{(d(h_i, h_i^+) - d(h_i, h_i^-) + m), 0\} \tag{0.2}$$

As motivated in the SimCSE paper, the loss-function 0.1 ensures an distribution of embeddings around the entire embedding space. Since this is not ensured in this case, we expect a decline in performance using the loss-function 0.2.

0.7.2 Results

The table 4 shows the results of the respective experiments. We can firstly see the vastly suboptimal performance of the base model alone. Since the training data differs vastly from the domain-specific test data, the model fails to make accurate predictions on the test data. While this effect can be seen for both domains, the adapters in general vastly help the performance in all cases. We can see that both the Pfeiffer and the Houlsby-Adapter incorporate the new data similarly well, with the K-Adapter achieving slightly worse results. The finetuning of the entire base model outperforms the Adapter-models, however the Houlsby-Adapter achieves a very similar performance.
Lastly, the table 5 shows very similar results, with the same phenomena visible when using the SPECTER loss-function. As expected, the results are slightly worse than the previous experiment.

Model	AskUbuntu	SPECTER				Average
		Cite	CC	CR	CV	
princeton-nlp/sup_simcse_bert_base	60.3	79.3	82.10	76.87	78.36	75.39
Finetuned princeton-nlp/sup_simcse_bert_base	65.2	88.3	88.11	84.46	83.63	81.94
Finetuned Houlsby-Adapter	64.5	87.3	89.01	82.41	84.42	81.53
Finetuned Pfeiffer-Adapter	64.2	87.0	88.63	81.98	84.41	81.24
Finetuned K-Adapter	62.8	85.3	87.92	80.05	83.29	79.87

Table 4: The evaluation scores of the Adapter architectures on the two domain-specific datasets using the SimCSE loss 0.1.

Model	AskUbuntu	SPECTER				Average
		Cite	CC	CR	CV	
Finetuned princeton-nlp/sup_simcse_bert_base	65.3	88.0	87.74	84.15	83.32	81.70
Finetuned Houlsby-Adapter	64.0	88.2	88.69	82.42	83.99	81.46
Finetuned Pfeiffer-Adapter	63.8	87.8	88.73	81.65	83.27	81.05
Finetuned K-Adapter	62.5	85.6	87.70	80.09	82.85	79.75

Table 5: The evaluation scores of the Adapter architectures on the two domain-specific datasets using the SPECTER loss 0.2.

0.7.3 Discussion

This experiment yields multiple very interesting insights, since independently of the domains and datasets, the Adapters succeed in incorporating the new data into their predictions. The results are very competitive, but fall slightly short of the finetuning of the entire base model. This however is to be expected and still a very good result, since the Adapter-training is far more efficient than training all of the base model's parameters. The results are very similar for both domains and motivate the usage of Adapters for domain-adaptation of general STS-models.

Due to the Adapters freezing the pretrained STS-base model, there would be reason to justify an increase in knowledge in the Adapter-induced models since the previously known STS-solving capabilities of the base model are preserved. However, such a combination of the general world STS-trained base model and the domain-specific Adapter-training can not be seen in these results. Thus, the performance-difference between the two approaches is surprising and can be attributed to the architecture of the Adapters which only can act upon a certain subset of layers within the base model.

Hence, the usage of the pretrained state of the base model only further shortens the training time needed to adapt to the new domain.

Furthermore we can see the effect of the two different loss-function, which was already discussed in the SimCSE-paper; the SimCSE-loss seems to more accurately learn from the data as the loss introduced in the SPECTER-paper.

0.8 Unsupervised improvement of STS-Adapters

Textual data benefits largely off of unsupervised training procedures, such as Masked Language Modelling. This is due to the fact, that the grammatical structure of textual data can easily be learned without human supervision or change of datasets. This largely increases the datasets usable for learning, since supervised data-annotation in general is a tedious and costly task.
Due to this, unsupervised models are of great interest to the research community, however often fail to produce competitive results. This encourages the following experiment which aims at improving an unsupervised learning process of a model by plugging in an Adapter which was previously trained on supervised knowledge.

0.8.1 Methodology

The experiment is based on the idea that a model which was trained in a supervised fashion yields a good starting point for an unsupervised training and thus increases the performance resulting after the unsupervised training procedure. Given a roberta-large base model and a dataset of sentence-triples, similarly to the previous experiments, the unsupervised training strategy of the SimCSE paper [10] is used to train the model on sentence similarity. This unsupervised training strategy is very similar to the already introduced supervised SimCSE strategy, but given a certain sentence x_i, produces an embedding h_i by piping it through the transformer layers of the used model. Due to the stochastic nature of transformers, piping the same sentence through the model again, results in a different embedding h_i^+, which is used as the positive sample.

A negative sample h_i^- is produced by following this process for an arbitrary, different sentence. Having created an artificial sentence-triple, the already introduced loss-function 0.1 is used. Thus, the model can learn STS-tasks from any unlabelled collection of sentences, enabling the use of nearly arbitrarily large datasets.
Since pretrained Adapters are widely available, the suggested improvement is to first plug-in a pretrained Adapter, adding a certain STS-solving capability from the start. The model thus trains both the base-model and the Adapter in the unsupervised fashion, capitalizing on the pretrained Adapter-weights, and being able to better grasp the relevant factors in contrast to the fully unsupervised start.
For this experiment, the general world dataset TREx-rc is used.

0.8.2 Results

The table 6 shows the results for the unsupervised training and the Adapter-induced unsupervised training. Interestingly, we can see the superiority of an Adapter-initialization of an unsupervised training-procedure while the actual training-process was left untouched.
This result is highly encouraging and a possible topic for further research. The Houlsby-Adapter has very consistently outperformed the Pfeiffer-Adapter, which is surprising due to its smaller amount of parameters.

Model	STS12	STS13	STS14	STS15	STS16	STS-B.	SICK-R.	Avg.
Unsupervised	71.27	83.75	75.26	85.04	81.17	81.69	70.84	78.43
Houlsby-Adapter	73.37	84.89	76.21	87.24	83.17	81.75	72.63	79.90
Pfeiffer-Adapter	72.49	83.62	76.14	85.54	82.63	81.55	72.43	79.20
K-Adapter	72.47	83.55	75.20	85.09	82.42	81.64	71.80	78.88

Table 6: The evaluation scores of the Adapter-induced unsupervised learning processes.

0.8.3 Discussion

The results clearly show the ability of the models to make use of the supervised Adapter-initializations and incroporate these into the further unsupervised training procedure. Similarly to the previous experiments, the K-Adapter is outperformed by both the Pfeiffer- and the Houlsby-Adapter. In this case, the Houlsby-Adapter has outperformed the Pfeiffer-Adapter, in contrast to the results in previous experiments, showing the highly task-dependent performance of the Adapters.

0.9 Conclusion

This work has looked into the training-peculiarities of Adapters for STS-tasks. Multiple different Adapter-models have been evaluated and different datasets have been used to answer the following research questions:

0.9.1 How to inject structured knowledge into Sentence Embedding Models with Adapters?

The experiments have shown that the knowledge obtained from the Knowledge Graph datasets have successfully been learned by the Adapter-induced models. Since Knowledge Graphs are made of data-triples, contrastive learning methods were seen as intuitive and shown to be effective.

In order to make this result meaningful, two different training objectives have been employed, based upon the State-of-the-art SimCSE-work [10] and the SPECTER-paper [11], which introduced a similar loss-function. Both training objectives have succeeded in injecting the knowledge from the data-triples into the models, with the SimCSE-loss performing slightly better than the SPECTER-loss.

In conclusion, structured knowledge was proven to be embeddable into Sentence Embedding Models using contrastive loss functions. The training process consisting of freezing the base model and training only the Adapter weights was seen as sufficient and nearly matching the results compared to a full training of the entire model. The injection of structured knowledge can thus be seen as a success.

0.9.2 Do Knowledge Adapters improve information retrieval tasks of Sentence Embedding Models?

Similar to previous works, this work has successfully shown a very similar performance of Adapter models compared to the full finetuning of the entire base model. Especially interesting is the performance being on-par in a multitude of different situations, such as the domain-specialization on different domains, the general-world knowledge extraction and across multiple different architectures of Adapters. This performance is achieved while only training a small amount of parameters, leading to Adapter models being deemed superior to training the entire basemodel in a situation of limited compute-power.
Furthermore does the smaller amount of parameters lead to faster training and less carbon emissions during training. Notable also is the improvement of unsupervised methods using supervised Adapters, which is applicable since the Adapters are freely available in a pre-trained state and can thus be embedded into any unsupervised learning process. The positive results of these experiments highlight the improvements Adapters can bring to sentence embedding models. As a conclusion, Knowledge Adapters do improve information retrieval tasks by firstly introducing a method of finetuning to new data and secondly introducing a very cheap way to build upon pretrained common knowledge models.
However, the performance of the finetuned base model was not exceeded in the case of supervised training, thus limiting the benefits of Adapters to the amount of compute time and comparable metrics, while introducing a small decrease in performance.
Lastly, there was not found to be an architecture of Adapters that outperformed its counterparts continuously, highlighting the task-dependent performance of Adapters relatively to eachother. Only the K-Adapters performance was consistently worse than the other architectures.

0.9.3 How to combine domain-specific knowledge adapters for the scholarly domain?

This work has looked into two different domains, firstly the scholarly domain, using the SPECTER-dataset, and secondly the technical domain, within the AskUbuntu-dataset. The experiments have shown a successful domain-adaptation using the Adapters in both cases, leveraging the knowledge distilled in the base models and using the domain-specific data to easily get to a similar performance compared to finetuning the entire model. However, there was not found any evidence of combining knowledge from the base model and the domain-specific adapter, since the performance of the base model was not reached. In general we can see that both the Houlsby and the Pfeiffer Adapter perform very similarly in domain adaptation, making both architectures very convenient for usage in multiple use-cases. The K-Adapter architecture however was consistently outperformed and is thus not recommended for usage.

Bibliography

[1] G. Majumder, D. P. Pakray, A. Gelbukh, and D. Pinto. “Semantic Textual Similarity Methods, Tools, and Applications: A Survey”. In: *Computacion y Sistemas* 20 (Dec. 2016), pp. 647–665. DOI: 10.13053/CyS-20-4-2506.

[2] N. Reimers and I. Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410.

[3] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, J. Ji, G. Cao, D. Jiang, and M. Zhou. “K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1405–1418. DOI: 10.18653/v1/2021.findings-acl.121. URL: https://aclanthology.org/2021.findings-acl.121.

[4] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. “Parameter-Efficient Transfer Learning for NLP”. In: *CoRR* abs/1902.00751 (2019). arXiv: 1902.00751. URL: http://arxiv.org/abs/1902.00751.

[5] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych. “AdapterFusion: Non-Destructive Task Composition for Transfer Learning”. In: (2020). DOI: 10.48550/ARXIV.2005.00247. URL: https://arxiv.org/abs/2005.00247.

[6] H. Le, J. Pino, C. Wang, J. Gu, D. Schwab, and L. Besacier. *Lightweight Adapter Tuning for Multilingual Speech Translation*. 2021. arXiv: 2106.01463 [cs.CL].

[7] B. Thomas, S. Kessler, and S. Karout. “Efficient Adapter Transfer of Self-Supervised Speech Models for Automatic Speech Recognition”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 7102–7106. DOI: 10.1109/ICASSP43922.2022.9746223.

[8] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang. “K-BERT: Enabling Language Representation with Knowledge Graph”. In: *CoRR* abs/1909.07606 (2019). arXiv: 1909.07606. URL: http://arxiv.org/abs/1909.07606.

[9] J. Youn and I. Tagkopoulos. *KGLM: Integrating Knowledge Graph Structure in Language Models for Link Prediction*. 2022. arXiv: 2211.02744 [cs.CL].

[10] T. Gao, X. Yao, and D. Chen. *SimCSE: Simple Contrastive Learning of Sentence Embeddings*. 2021. DOI: 10.48550/ARXIV.2104.08821. URL: https://arxiv.org/abs/2104.08821.

[11] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld. "SPECTER: Document-level Representation Learning using Citation-informed Transformers". In: *ACL*. 2020.

[12] H. ElSahar, P. Vougiouklis, A. Remaci, C. Gravier, J. S. Hare, F. Laforest, and E. Simperl. "T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.* 2018.

[13] T. Lei, H. Joshi, R. Barzilay, T. S. Jaakkola, K. Tymoshenko, A. Moschitti, and L. M. i Villodre. "Denoising Bodies to Titles: Retrieving Similar Questions with Recurrent Convolutional Models". In: *CoRR* abs/1512.05726 (2015). arXiv: `1512.05726`. URL: `http://arxiv.org/abs/1512.05726`.

[14] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe. "SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 497–511. DOI: `10.18653/v1/S16-1081`. URL: `https://aclanthology.org/S16-1081`.

[15] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre. "SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity". 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)*. Montréal, Canada: Association for Computational Linguistics, July 2012, pp. 385–393. URL: `https://aclanthology.org/S12-1051`.

[16] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. "*SEM 2013 shared task: Semantic Textual Similarity". In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 32–43. URL: `https://aclanthology.org/S13-1004`.

[17] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe. "SemEval-2014 Task 10: Multilingual Semantic Textual Similarity". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 81–91. DOI: `10.3115/v1/S14-2010`. URL: `https://aclanthology.org/S14-2010`.

[18] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, L. Uria, and J. Wiebe. "SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability". In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 252–263. DOI: `10.18653/v1/S15-2045`. URL: `https://aclanthology.org/S15-2045`.

[19] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. DOI: `10.18653/v1/S17-2001`. URL: `https://aclanthology.org/S17-2001`.

[20] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. "A SICK cure for the evaluation of compositional distributional semantic models". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 216–223. URL: `http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf`.