Policy Domain Prediction from Party Manifestos with Adapters and Knowledge Enhanced Transformers

Hsiao-Chu Yu **Ines Rehbein** **Simone Paolo Ponzetto**
Data and Web Science Group
University of Mannheim
hsiao-chu.yu@students.uni-mannheim.de, {rehbein,ponzetto}@uni-mannheim.de

Abstract

Recent work has shown the potential of knowledge injection into transformer-based pre-trained language models for improving model performance for a number of NLI benchmark tasks. Motivated by this success, we test the potential of knowledge injection for an application in the political domain and study whether we can improve results for policy domain prediction, that is, for predicting fine-grained policy topics and stance for party manifestos. We experiment with three types of knowledge, namely (1) domain-specific knowledge via continued pre-training on in-domain data, (2) lexical semantic knowledge, and (3) factual knowledge about named entities. In our experiments, we use adapter modules as a parameter-efficient way for knowledge injection into transformers. Our results show a consistent positive effect for domain adaptation via continued pre-training and small improvements when replacing full model training with a task-specific adapter. The injected knowledge, however, only yields minor improvements over full training and fails to outperform the task-specific adapter without external knowledge, raising the question which type of knowledge is needed to solve this task.

1 Introduction

Identifying policy domains in political text such as parliamentary speeches or party manifestos is an important ingredient for many analyses in political science. This type of information is crucial for studying party competition and voting behaviour or for investigating agenda setting and framing, and for many other research questions in the field. Many research projects have thus addressed this problem, either by creating annotated data sets for manual and automated analyses (Baumgartner et al., 2006; Bevan, 2019; Volkens et al., 2019b) or by developing systems for policy domain prediction (Subramanian et al., 2017; Glavaš et al., 2017; Abercrombie et al., 2019; Koh et al., 2021). This task, however, is quite challenging, due to the large number of fine-grained topic labels in the respective coding schemes. For many of these labels, only a small number of annotated instances exist in the training set. Furthermore, as this type of annotation has been adopted in different research projects and across countries and time, the annotations themselves include inconsistencies, as the defined classes might have been interpreted differently by the coders, depending on their background, situational context and training.

One way to address (at least part of) this problem is to enrich the models with external information, in order to make them more robust to inconsistencies in the data and to provide more information especially for the infrequent labels. A number of studies have looked into this problem, with promising results. Previous work has demonstrated improvements for various natural language understanding tasks by incorporating general human knowledge presented in knowledge bases (Zhang et al., 2019; Sun et al., 2019; Peters et al., 2019; Lauscher et al., 2020b) and by adapting pre-trained language models (PLMs) to specific domains (Lee et al., 2020; Beltagy et al., 2019; Gururangan et al., 2020). However, these approaches are resource intensive as they typically require either re-training the entire model from scratch (Lauscher et al., 2020b) or tuning pre-trained parameters (Zhang et al., 2019) on auxiliary pre-training tasks.

To alleviate these problems, researchers have turned to the lightweight adapter architecture (Houlsby et al., 2019; Pfeiffer et al., 2021) for knowledge integration. The *adapter module* (or simply *adapter*) is a set of parameters inserted into the original transformer layers in the pre-trained model. Unlike the standard fine-tuning of BERT-based models where the entire model is updated, the adapter-based tuning only updates the newly inserted adapter parameters when the model is tuned on downstream tasks, while the underlying pre-

Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023), pages 229–244
September 18–22, 2023. ©2023 Association for Computational Linguistics

trained model is frozen. This approach makes model tuning more efficient, due to the smaller size of parameters that need to be trained. In addition to its efficiency, several studies have demonstrated the effectiveness of adapters for knowledge injection into BERT-based models (Hung et al., 2022; Meng et al., 2021; Lauscher et al., 2020a).

Building upon this body of work, we use the adapter-based approach to incorporate multiple knowledge sources into multilingual RoBERTa (XLM-R) (Conneau et al., 2020). Different from past studies that mostly focused on integrating single knowledge sources, we enrich the pre-trained language model with multiple types of knowledge: (i) domain knowledge, (ii) lexical semantic knowledge (such as word synonyms) and (iii) factual knowledge about named entities (e.g., Angela Merkel is a politician). The main research questions addressed in this paper are:

RQ1 How does external knowledge, such as domain knowledge and structured knowledge from knowledge bases, impact the language model's capability to understand natural language in the political science domain?

RQ2 Can we use adapters to inject this knowledge into a pre-trained language model in a more parameter-efficient manner?

The paper is structured as follows. In Section 2, we outline related work on topic and policy prediction in the political domain and review recent studies that incorporate adapters into PLMs. Section 3 presents our approach for adapter-based knowledge injection, and Section 4 discusses our results for predicting policy domains from party manifestos, using adapters and external domain and world knowledge. In Section 5, we conclude and outline future work.

2 Related Work

2.1 Predicting Manifesto Policy Domains

Many studies in the context of computational political text analysis have focused on topic or policy issue prediction, using dedicated datasets created within the Comparative Agenda Project (Baumgartner et al., 2006; Bevan, 2019) or the Comparative Manifesto Project (CMP) (Mikhaylov et al., 2012; Werner et al., 2014). In our work, we use the Manifesto Corpus from the CMP which includes a large collection of party manifestos from over 50 countries. Each document in the corpus has been segmented into "quasi-sentences" (mostly clauses) and has been manually categorized into eight coarse-grained policy domains (see Table 1). Those main classes are further subdivided into a set of 57 fine-grained policy goals and issues that also encode the author's stance towards a specific policy issue (positive/negative), as illustrated in Example 2.1.

Label	Policy Domain	% of quasi-sentences
1	External Relations	6.6
2	Freedom & Democracy	4.7
3	Political System	10.6
4	Economy	24.9
5	Welfare & Quality of Life	30.9
6	Fabric of Society	11.2
7	Social Groups	10.0
0	Not Categorized	1.1

Table 1: Distribution of major policy domains in the manifesto dataset of Koh et al. (2021).

Ex. 2.1

"We view the diversity of our nation not as a liability, but rather as a shared strength and source of pride"

Main topic: FABRIC OF SOCIETY

Minor topic: MULTICULTURALISM → POSITIVE

2.2 Introducing domain-specific knowledge into PLMs

Transfer learning based on large, pre-trained language models (PLMs) has shown to improve model performance of transformer-based architectures for a wide range of NLP tasks (Devlin et al., 2019; Liu et al., 2019). The model is trained on large amounts of text, using self-supervision, which provides the model with information about language structure and the meaning of words in context. Exploiting this generic knowledge to specific downstream tasks reduces the amount of training data needed for each task. However, many domains require the model to understand specialised vocabulary terms and information that the model cannot learn from generic corpora such as Wikipedia. Below, we describe a number of techniques that have been proposed to address this shortcoming.

Domain adaptation Many studies have demonstrated that continued pre-training of PLMs on domain-specific corpora before fine-tuning them for the final task can improve model performance of transformer-based models. BioBERT (Lee et al., 2020) and ClinicalBERT (Alsentzer et al., 2019) both adopted the continual pre-training framework. Other work has skipped pre-training on generic text collections and, instead, pre-trained domain-specific models from scratch (Beltagy et al., 2019; Gu et al., 2022). In our work, we use PolSciBERT, a PLM that has been adapted to the political domain through continual pre-training.

External knowledge injection Numerous studies have shown that integrating knowledge graphs into BERT-based models is beneficial for natural language understanding tasks (Sun et al., 2019; Zhang et al., 2019; Peters et al., 2019; Lauscher et al., 2020b; Peinelt et al., 2021). These studies mainly focused on two types of knowledge: facts about entities and linguistic knowledge. Zhang et al. (2019) aligned named entities in the Wikipedia corpora with entities in the knowledge base Wikidata (Vrandečić and Krötzsch, 2014) and trained the model, ERNIE, to learn the alignment, based on an *entity alignment masking objective*. Sun et al. (2019) proposed Baidu-ERNIE, which was pre-trained via knowledge masking strategies. Specifically, the authors used entity-level and phrase-level masking techniques on Chinese Wikipedia and in-house text collections in their masked language model pre-training. Peters et al. (2019) utilized the multi-head attention mechanism to fuse knowledge from multiple knowledge bases, while Peinelt et al. (2021) adopted the gating mechanism to combine linguistic embeddings and contextual embeddings from BERT. Lauscher et al. (2020b) effectively introduced word-level semantic similarity information into BERT via additional pre-training by predicting semantic relations in a knowledge graph.

Building on this line of work, we propose to enrich PolSciBERT with (1) lexical semantic information and (2) knowledge about named entities.

Adapter-based architectures Most of the work described above involves re-training the entire model with additional pre-training objectives which, due to the large number of parameters, is computationally expensive and might suffer from catastrophic forgetting (McCloskey and Cohen, 1989). To alleviate this problem, adapters have been proposed as an alternative strategy for downstream fine-tuning (Rebuffi et al., 2017; Houlsby et al., 2019; Pfeiffer et al., 2020a). Unlike the standard fine-tuning approach, adapter-based tuning does not require re-training the entire model. Instead, it injects a lightweight task-specific adapter layer in each transformer layer. During fine-tuning, these newly added adapter layers are trained along with the final classification layer, while the original pre-trained parameters are frozen. Fixing the original pre-trained model makes it easier to share its parameters across several different tasks. In addition, the adapter layer typically has a much smaller number of parameters than the original pre-trained model, making adapter-based fine-tuning much more efficient.

A number of studies have leveraged the adapter-based approach and demonstrated its potential not only for domain adaptation (Lu et al., 2021; Hung et al., 2022; Meng et al., 2021), but also for integrating structured knowledge bases into transformer-based models (Wang et al., 2021; Lauscher et al., 2020a). Inspired by these studies, this work focuses on incorporating knowledge bases into PolSciBERT using adapters, to investigate whether semantic similarity and/or entity knowledge can also be beneficial for NLP tasks in the political domain. We compare different methods for combining multiple adapters, namely adapter stacking (Pfeiffer et al., 2020b) and adapter fusion (Pfeiffer et al., 2021).

3 Training Knowledge Adapters

To introduce knowledge into PolSciBERT, we pre-train a number of specialized adapters, each of which encodes a certain type of knowledge. These pre-trained modular adapters allow us to transfer knowledge from external sources into our model. We first describe our base model, PolSciBERT, and then explain the training procedure of the adapters.

All models are implemented in PyTorch, using the HuggingFace Transformers library (Wolf et al., 2020)[1] and the adapter-transformer library from AdapterHub (Pfeiffer et al., 2020a).[2]

3.1 PolSciBERT

PolSciBERT is based on the multilingual XLM-R model (Conneau et al., 2020) and was further pre-trained in a multilingual setting with full fine-tuning. Specifically, the pre-training corpus is a collection of parliamentary speeches in 5 languages, German, English, Spanish, French and Italian, including debates from the European parliament (Koehn, 2005) and transcripts from parlia-

[1] v4.17.0. https://huggingface.co/transformers.

[2] v3.0.0. https://docs.adapterhub.ml.

mentary meetings (Rauh and Schwalbach, 2020; MIT Election Data and Science Lab, 2017).[3] Starting with the pre-trained XLM-R, we continued pre-training of PolSciBERT on the political text corpus, using the masked language modelling (MLM) objective.

3.2 Corpora for knowledge injection

We explore two publicly available datasets to acquire different types of knowledge: ConceptNet (Speer et al., 2017) for semantic (dis)similarity and the KELM corpus (Agarwal et al., 2021) for factual information about entities.

ConceptNet (Speer et al., 2017) is a large multilingual knowledge base which encodes commonsense knowledge, such as the *causes* of an event (e.g., exercise causes sweat) or the *synonyms* of a word. It integrates multiple knowledge sources, including Wiktionary and a subset of DBPedia (Lehmann et al., 2012). The latest version (ConceptNet 5.7) comprises 34 million edges and supports hundreds of languages. ConceptNet has been used in NLP research to incorporate external knowledge into large language models (Camacho-Collados et al., 2017; Zhong et al., 2019; Lauscher et al., 2020a; Yasunaga et al., 2021).

Since we are interested in enriching PolSciBERT with semantic similarity and dissimilarity information, we extract edges from the knowledge graph for three types of lexical relations (*IsA*, *Synonym* and *Antonym* relations) and 5 languages (DE, EN, IT, FR, ES) as training data for our knowledge adapters. For each relation type, we extract all word pairs connected by this relation. Then we perform a simple clean-up and split the data into training (85%) and test set (15%) for adapter training. We only keep word pairs where both words exist in each of the 5 languages and remove duplicates from the data. We also remove triplets whose entities contain numbers or have a word length of ≤ 1 character. Note that the triplets can be cross-lingual (e.g., <*Synonym*, énorme (FR), enorme (IT)>).

To train the CN-SIMILARITY adapter, we merge the *IsA* and *Synonym* relation triplets into one training and test set since they both encode information on semantic similarity. This results in 1.3 million training instances for CN-SIMILARITY. The CN-ANTONYM adapter was trained solely on triplets from the *Antonym* relation, which comprises 30,000 training instances.

	Task	Train Size	Test Size
CN-SIMILARITY	TCL	1,317,027	232,417
CN-ANTONYM	TCL	30,501	5,383
KELM-ADAP	MLM	13,284,213	2,344,273

Table 2: Summary of adapter training tasks and data (TCL: triple classification; MLM: masked language modelling).

KELM In addition to word or phrase level semantic information, factual knowledge about named entities has also proven to improve the performance of pre-trained language models (Zhang et al., 2019; Sun et al., 2019). Thus, we utilize the Corpus for Knowledge-Enhanced Language Model pre-training (KELM) (Agarwal et al., 2021) to inject factual knowledge into PolSciBERT. The KELM corpus is a synthetic corpus generated by a T5 model (Raffel et al., 2020). The model has been fine-tuned on aligned data from English Wikidata (Vrandečić and Krötzsch, 2014) and Wikipedia by training the model to convert the Wikidata triples to natural text (Agarwal et al., 2021).

The raw dataset[4] includes more than 15 million instances. Each instance is a `JSON` object with three fields: (1) a list of triples where each triple is in the format `[head entity, relation, tail entity]`, (2) the serialized triple sequence which is concatenated by the list of triples and input to the T5 model, and (3) the generated text output of the T5 model. For an example, refer to Figure 1 in the Appendix. The average length of the generated sentences in the KELM corpus is 15.2 tokens.

To create the dataset for training the KELM adapter (KELM-ADAP), we extract the generated text (the `gen_sentence` field) from each instance in the raw dataset and split the resulting dataset into training set (85%) and test set (15%). The training (test) set includes about 13 million (2 million) sentences, as summarized in Table 2.

3.3 Adapter Training

For all our experiments, we adopt the adapter architecture proposed in Pfeiffer et al. (2021). That is, we insert a single adapter with a bottleneck hidden size M after the feed-forward sub-layer in the transformer layer (Vaswani et al., 2017).

[3] For a detailed list of datasets and information on preprocessing and pre-training, please refer to §A in the Appendix.

[4] Downloaded from `https://github.com/google-research-datasets/KELM-corpus` on April 23, 2022.

CN-SYNONYM and CN-ANTONYM The ConceptNet adapters aim at enriching PolSciBERT with semantic similarity and dissimilarity information. To learn this type of knowledge, we follow Lauscher et al. (2020b) and train the adapter in a relation classification task where we input a word pair from our data and predict whether a CN-SYNONYM (CN-ANTONYM) relation holds between the two words.

The negative samples needed for training have been created, using an approach similar to Yao et al. (2019). For each relation, a triple from the data is corrupted by replacing either its head h or its tail t (but not both) by a randomly selected entity h' or t' from the dataset. We make sure that the new, corrupted triple does not appear in the dataset, to avoid inserting false negatives. This way, we create k corrupted triples for k true triples, resulting in $2k$ triples in total. The set of negative samples can be presented as

$$D_R^- = \{(h', r, t) | r \in R \wedge h' \in E \wedge h' \neq h \wedge (h', r, t) \notin D_R^+\}$$
$$\cup \{(h, r, t') | r \in R \wedge t' \in E \wedge t' \neq t \wedge (h, r, t') \notin D_R^+\},$$

where E is the set of all entities in a semantic relation R, and D_C^+ is the set of positive triples for the semantic relation R.

Similar to previous work (Yao et al., 2019; Lauscher et al., 2020b), we model a word pair (h, t) as a sequence pair to perform the relation classification task. Specifically, each word pair (h, t) in a semantic relation,[5] including both positive and negative examples, is turned into a sequence pair that starts with the <s> token and is separated by the </s> token. For illustration, the true word pair in the *Similarity* relation <color blind, farbenblind> is transformed into

```
[s] _color _blind  [/s][/s] _far ben blind [/s]
```

The relation classification task can thus be modeled as a standard sequence pair classification task for transformer models (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020). The last output hidden state of the `[s]` token is used for prediction. For a true positive instance, the correct label is 1, and 0 for the generated negative examples.

KELM-ADAP For the KELM adapter, we seek to encode facts about named entities in the world. To achieve this goal, we train the adapter with the masked language modeling objective (MLM) (Devlin et al., 2019; Lauscher et al., 2020a; Lu et al., 2021) on the KELM dataset described above. We follow the standard MLM procedure to randomly mask 15% of the tokens in each input sequence and use the last hidden state of the masked token for prediction.[6]

4 Experiments

We now want to test our knowledge adapters on the task of predicting policy positions in political manifestos.

Baselines As baselines, we use multilingual RoBERTa (XLM-R) (Conneau et al., 2020) and PolSciBERT, our multilingual in-domain RoBERTa model, to assess whether domain-specific knowledge improves model performance **(RQ1)** and whether the effect of inserting additional lexical and/or factual knowledge in the model can further improve results **(RQ2)**.

4.1 Predicting Manifesto Policy Domains

The Manifesto Project Database (Volkens et al., 2019a) has been widely used in political text analysis (Laver et al., 2003; Abercrombie et al., 2019; Menini et al., 2017; Glavaš et al., 2017; Koh et al., 2021).[7] It comprises a large collection of party manifestos from over 50 countries. The text in the party manifestos has been segmented into "quasi-sentences" (similar to clauses). Each quasi-sentence contains exactly one unique statement (Werner et al., 2021) and has been categorized into one of 57 fine-grained classes reflecting the most relevant policy goal and issue preference for this statement. These 57 policy goals and issues are grouped into 8 coarse-grained policy domains. Thus, each quasi-sentence in the dataset has a coarse-grained policy domain label (the "major label") and a fine-grained label capturing the policy goal and issue (the "minor label"). For illustration, see Example 2.1.[8]

To compare our results with related work, we evaluate our models on the dataset of Koh et al. (2021) which includes a subset of the manifesto corpus (version 2019) (Volkens et al., 2019a,c) consisting of all English manifestos.[9] Koh et al. (2021) split this subset into training, validation and test

[5] *Synonym* and *IsA* for CN-SYNONYM; *Antonym* for CN-ANTONYM

[6] For training details and hyperparameters, see §B in the Appendix.

[7] https://manifesto-project.wzb.eu/

[8] For more information, schema please refer to the codebook of the Manifesto Project (Volkens et al., 2019b,d)

[9] Note that there are two versions *2019a* and *2019b*, but the authors did not specify which version they used.

	Major topics	Minor topics
Number of labels	8	57
Number of quasi-sent.		
Total	99,279	99,279
Train (0.70)	69,499	69,499
Validation (0.15)	14,887	14,887
Test (0.15)	14,893	14,893

Table 3: Number of labels and examples in the final manifesto dataset

sets with a ratio of 70/15/15. We first remove examples with empty text fields from the data, and then follow the same split to evaluate our models. The final dataset includes 99,279 quasi-sentences (see Table 3).

Following Koh et al. (2021), we perform the quasi-sentence classification task for both major and minor topics. We model the task as a text classification problem and use the last hidden state of the `[S]` token as a pooled representation of the input sequence to predict labels and compute the loss. During evaluation, we noticed some preprocessing problems in the dataset, specifically missing tokens at the end of most quasi-sentences. We therefore tried to recreate the dataset with complete quasi-sentences and report results for both datasets (see Appendix, C for a more detailed description of the problem and information on the recreated dataset).

4.2 Experimental setup

To investigate the effectiveness of knowledge injection via adapters, we experiment with three different model setups for our semantic similarity knowledge adapters (CN-SIMILARITY) and the factual knowledge adapter KELM-ADAP, following previous work in this area (Lauscher et al., 2021; Pfeiffer et al., 2020b, 2021):

- **Adapter full fine-tuning** inserts one single pre-trained knowledge adapter into PolSciBERT and tunes the entire model, including the PolSciBERT parameters and the inserted adapter. That is, the model is initialized with the pre-trained parameters and updated during fine-tuning on the downstream task.

- **AdapterStack** utilizes the AdapterStack architecture (Pfeiffer et al., 2020b) and stacks adapters –the pre-trained knowledge adapter(s) and a randomly initialized task adapter on top– and only tunes the task adapter during fine-tuning while PolSciBERT and all knowledge adapters are frozen. This setup differs from *Adapter full fine-tuning* in that the model learns the task-specific information separately, which might be better at preserving the in-domain information encoded in PolSciBERT and the knowledge encoded in the pre-trained adapters (Lauscher et al., 2021).

- **AdapterFusion** (Pfeiffer et al., 2021) combines multiple pre-trained knowledge adapters and a pre-trained task adapter, using a randomly initialized fusion layer. Similar to the attention mechanism (Vaswani et al., 2017), the fusion layer learns to weight the different pre-trained adapters for the downstream task. During downstream fine-tuning, PolSciBERT and all adapters are frozen, only the parameters in the fusion layer are updated.

For all three setups, the final task-specific prediction head is randomly initialized. Additional task adapters are pre-trained for the *AdapterFusion* (Pfeiffer et al., 2021) setup. Specifically, we follow the standard single task training for adapters (Pfeiffer et al., 2021; Houlsby et al., 2019), in which randomly initialized task adapters are inserted into PolSciBERT and fine-tuned on the downstream task while PolSciBERT is kept frozen. For training details, also see Appendix B.1 and B.2.

4.3 Results

Table 4 reports results for major and minor topics on our dataset. Results for the original dataset from Koh et al. (2021) are included in the Appendix.

4.4 Baseline Results

Our baseline models (XLM-R, PolSciBERT) outperform the BERT-GRU and BERT-CNN models of Koh et al. (2021) by 2-3% Micro-F1 for the major topics and by around 5% Micro-F1 for minor topics (see Appendix, Table 8). For Macro-F1, the improvements are more profound, with around 10% for the fine-grained minor topics.

When training the same models on our new dataset (without missing tokens), we observe a slight increase in results across most settings, with one noteworthy exception. For Macro-F1 on the minor topics, results on the corrupted training (and test) data were higher (around 5% for PolSciBERT, from 36% to 31%). We will look into this issue in §4.7.

		Major Topics		**Minor Topics**	
Model Setup		Micro-F1	Macro-F1	Micro-F1	Macro-F1
Baselines (w/o adapters)	XLM-R	$62.3_{(0.2)}$	$51.0_{(0.3)}$	$49.1_{(0.4)}$	$32.8_{(1.2)}$
	PolSciBERT	$64.6_{(0.6)}$	$53.4_{(0.8)}$	$50.8_{(0.3)}$	$31.2_{(2.2)}$
(with adapter)	PolSciBERT + task adapter	$65.0{*}_{(\mathbf{0.1})}$	$\mathbf{54.5}_{(0.4)}$	$\mathbf{51.8}{*}_{(\mathbf{0.3})}$	$\mathbf{36.5}_{(0.2)}$
CN-SYNONYM	Full	$62.6_{(0.8)}$	$52.5_{(0.9)}$	$49.6_{(0.7)}$	$35.0_{(0.8)}$
	AdapterStack	$64.1_{(0.8)}$	$53.3_{(0.8)}$	$51.3_{(0.6)}$	$35.5_{(1.2)}$
	AdapterFusion	$65.0_{(0.5)}$	$54.3_{(0.2)}$	$51.7{*}_{(\mathbf{0.3})}$	$36.0_{(0.5)}$
KELM-ADAP	Full	$62.5_{(0.7)}$	$53.3_{(0.4)}$	$50.0_{(0.6)}$	$34.3_{(1.9)}$
	AdapterStack	$64.8_{(0.3)}$	$54.1_{(0.3)}$	$51.5{*}_{(\mathbf{0.3})}$	$36.0_{(0.4)}$
	AdapterFusion	$64.7_{(0.2)}$	$54.0_{(0.2)}$	$\mathbf{51.8}{*}_{(\mathbf{0.2})}$	$36.2_{(0.5)}$
CN-SYNONYM & KELM-ADAP	AdapterStack	$63.8_{(0.2)}$	$52.9_{(0.3)}$	$51.2_{(0.2)}$	$35.4_{(0.6)}$
	AdapterFusion	$\mathbf{65.2}{*}_{(\mathbf{0.4})}$	$54.4_{(0.3)}$	$51.6{*}_{(\mathbf{0.2})}$	$36.2_{(0.8)}$
Experiments including antonym relations					
CN-SYNONYM & CN-ANTONYM	AdapterStack	$61.8_{(1.0)}$	$50.6_{(1.3)}$	$51.3{*}_{(\mathbf{0.5})}$	$35.7_{(0.6)}$
	AdapterFusion	$65.0_{(0.5)}$	$54.2_{(0.4)}$	$51.7{*}_{(\mathbf{0.3})}$	$36.4_{(0.3)}$
CN-SYNONYM & CN-ANTONYM & KELM-ADAP	AdapterStack	$62.1_{(0.5)}$	$51.0_{(0.8)}$	$50.9_{(0.5)}$	$34.9_{(1.0)}$
	AdapterFusion	$65.1{*}_{(\mathbf{0.2})}$	$\mathbf{54.5}_{(0.2)}$	$51.5{*}_{(\mathbf{0.1})}$	$34.7_{(2.5)}$

Table 4: Test set results of the manifesto quasi-sentence domain classification (Major topics). The first column specifies the model setup, including the knowledge adapter(s) and the fine-tuning strategy applied. All evaluation metrics reported for our model setups were averaged over 5 random initializations. The number in the parenthesis indicates the standard deviation of the 5 runs. Micro-F1 results marked with $*$ are significantly better than the PolSciBERT baseline w/o adapters (Cochran's Q with $p <= .001$).

4.5 Domain Adaptation

We observe an increase in results of around 2% (major topics) for PolSciBERT, compared to the vanilla XLM-R. For the minor topics, results are mixed, with improvements in the same range for Micro-F1 while Macro-F1 decreases, probably caused by a high number of infrequent topics. Our results show that domain adaptation through continuous pre-training on in-domain data from the political domain has a positive effect (**RQ1**). When replacing full finetuning with a task adapter, we see further improvements especially for the minor topics. In addition, the task adapter seems more robust (increase in standard deviation). Next, we look into the performance of the knowledge adapters.

4.6 Knowledge Adapters

Full fine-tuning vs. freezing the LM parameters In general, PolSciBERT equipped with a single knowledge adapter, either CN-SIMILARITY or KELM-ADAP, brings performance benefits across different fine-tuning strategies compared to PolSciBERT without any adapters. When comparing results for AdapterStack and AdapterFusion with full fine-tuning, we see that for all settings it is beneficial to freeze the LM parameters as well as the knowledge adapter parameters and update only the weights for the task-specific adapter and (for AdapterFusion) the fusion layer.

Stacking vs. Fusion Our second observation concerns the performance of AdapterStack versus AdapterFusion. When inserting only one knowledge adapter, AdapterFusion works better or on par with AdapterStack. However, when combining multiple knowledge adapters, adapter fusion substantially outperforms stacking and yields improvements in the range of 3-4% for the major topics. This shows that letting the model learn the weights for the different adapters is beneficial. Overall, however, the knowledge adapters do not outperform the task-specific adapter.

Micro-F1 vs. Macro-F1 For the fine-grained minor topics, we observe more significant improvements for Macro-F1 than for the Micro-F1 metric. This implies that the improvements we gain from adapter training are mostly driven by improvements for the rare labels in the dataset. That is, the adapters seem to be mostly helpful for sparse data (i.e., topics with few instances). This observation is interesting, as it shows that the adapters seem to have learned additional information that our in-domain PolSciBERT has not yet learned (as evidenced by the lower Macro-F1 of PolSciBERT, compared to the vanilla XLM-R model).

Type of knowledge adapters When comparing the different types of knowledge that we inserted, we do not see any crucial differences between the entity-based knowledge and the semantic similarity adapters. Both types of information yield similarly small improvements. This raises some doubts whether the information we inserted is crucial to solve our task. We will come back to this question in §4.7.

Lessons learned Our results show that freezing the LM parameters and training only the weights of the adapter(s) can outperform full fine-tuning, at least in our setup. This provides more evidence that adapters are a good way to prevent "catastrophic forgetting" (Kirkpatrick et al., 2017; Lauscher et al., 2021).

4.7 Error Analysis

We will now look into some open questions mentioned above. First, we would like to know why Macro-F1 for PolSciBERT for the minor classes decreased (as compared to the vanilla XLM-R model) when training on the new dataset while, at the same time, Micro-F1 for PolSciBERT increased. This was in contrast to the results on the original dataset of Koh et al. (2021) where both, Micro and Macro-F1 for PolSciBERT were around 2% higher than the ones for the generic XLM-R. When looking into the data, we found that PolSciBERT trained on the newly created dataset does not predict labels for 14 out of the 57 classes. Those classes are the ones with few training (and test) instances only and the underlying reason for the different behaviour of the two models lies in the way the data was sampled. Koh et al. (2021) decided to create a training set where the different classes are equally distributed over the train/dev/test sets. In contrast to this approach, we did not distribute sentences from the same file over train, dev and test but selected 33 unseen manifestos and put all sentences from those documents in the test set. This results in a slightly less balanced, but more realistic test case. We assume that, as a result of our sampling decision, the model had more difficulties to predict the low-frequency classes which resulted in a lower Macro-F1 but higher accuracies for most other predicted classes.

4.8 Zero-shot experiments for German

In our final experiment, we test our multilingual model on German data in a zero-shot setup where we predict policy domains and preferences in a new, unseen language. We apply our model that has been fine-tuned exclusively on English data to German manifestos that have been annotated within the same framework.[10] We are interested to see (i) how well the model does without any task-specific German training data and (ii) which of the different methods (if any) is able to improve results over the baseline.

		F1 (Major)		F1 (Minor)	
	Model Setup	Mic.	Mac.	Mic.	Mac.
English	XLM-R	62.3	51.0	49.1	31.8
	PolSciBERT	64.6	53.4	50.8	31.2
	PolSciB+Adap	**65.0**	**54.5**	**51.8**	**36.5**
	CN-SYN AdaptFus	**65.0**	54.3	51.7	36.0
	KELM AdaptFus	64.7	54.0	**51.8**	36.2
German	XLM-R	51.5	41.8	35.7	22.5
	PolSciBERT	**56.8**	48.0	41.4	24.6
	PolSciB+Adap	56.3	47.9	**41.5**	**27.6**
	CN-SYN AdaptFus	56.5	47.3	40.2	26.8
	KELM AdaptFus	56.5	**48.8**	**41.8**	25.8

Table 5: Results for English (from Tab. 4) and zero-shot results for German manifestos.

Our results for German show a decrease of more than 10% for the vanilla XLM-R for major topics (62.4% vs. 51.5% Macro-F1) and around 15-20% for the minor topics. The in-domain PolSciBERT is able to improve results for major and minor topics by around 5% (Micro-F1). However, as seen for English, none of the knowledge adapters is able to obtain further significant improvements over the best model trained without external knowledge, again questioning whether the information that we injected in the model is needed for solving the task at hand. The adapters, however, provide competitive results without the need to retrain the full model.

5 Conclusions

Inspired by previous work on enhancing transformer-based LMs with domain knowledge, common-sense knowledge and semantic similarity information, we tested the impact of knowledge injection for the task of policy domain prediction from party manifestos. Our results showed that (a) in-domain pre-training can yield substantial improvements (PolSciBERT vs. vanilla XLM-R); (b) freezing the LM parameters and training task-specific adapters can yield comparable or better results, compared to full model finetuning; and (c) adapter fusion is especially important when integrating more than one adapter in the model.

[10] Our test set includes German manifestos from 1998 – 2021 (88,694 quasi-sentences), downloaded from `https://manifesto-project.wzb.eu` (see Table 3 in the Appendix).

6 Limitations

While our results showed the effectiveness of adapters as a parameter-efficient alternative to full fine-tuning, our attempts to improve model performance based on the injection of external knowledge were not successful. This, however, does not prove that knowledge injection for the task at hand is not feasible. More thorough testing of different types of knowledge is needed to answer the question whether knowledge injection can improve results for policy domain prediction from party manifestos.

7 Ethical Considerations

While the task of policy domain prediction from party manifestos has attracted a lot of attention especially in the political sciences and in the field of Text-as-Data, it is clear that the results so far are not yet good enough for applications in the real world. We thus advise researchers not to use the output of our system for political text analyses without any manual post-correction.

Acknowledgements

The work presented in this paper is partly funded by the German Research Foundation (DFG) under the UNCOVER project (PO 3536/3-1 and 1900/7-1).

References

Gavin Abercrombie, Federico Nanni, Riza Batista-Navarro, and Simone Paolo Ponzetto. 2019. Policy preference detection in parliamentary debate motions. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 249–259, Hong Kong, China. Association for Computational Linguistics.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Adrien Barbaresi. 2018. A corpus of German political speeches from the 21st century. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 792–797, Paris, France. European Language Resources Association (ELRA).

Frank R. Baumgartner, Christoffer Green-Pedersen, and Bryan D. Jones. 2006. Comparative studies of policy agendas. *Journal of European Public Policy*, 13(7):959–974.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Shaun Bevan. 2019. Gone Fishing: The Creation of the Comparative Agendas Project Master Codebook. In *Comparative Policy Agendas: Theory, Tools, Data*. Oxford University Press.

Andreas Blaette. 2017. GermaParl. Corpus of Plenary Protocols of the German Bundestag. TEI files.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2017. Cross-lingual classification of topics in political texts. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 42–46, Vancouver, Canada. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Chia-Chien Hung, Anne Lauscher, Simone Ponzetto, and Goran Glavaš. 2022. DS-TOD: Efficient Domain Specialization for Task-Oriented Dialog. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 891–904, Dublin, Ireland. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Allison Koh, Daniel Kai Sheng Boey, and Hannah Béchara. 2021. Predicting policy domains from party manifestos with BERT and convolutional neural networks. In *Proceedings of the 1st Workshop on Computational Linguistics for Political Text Analysis (CPSS-2021)*, pages 67–77, Düsseldorf, Germany.

Anne Lauscher, Tobias Lueken, and Goran Glavaš. 2021. Sustainable Modular Debiasing of Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4782–4797, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anne Lauscher, Olga Majewska, Leonardo FR Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020a. Common sense or world knowledge? Investigating adapter-based knowledge injection into pretrained transformers. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online. Association for Computational Linguistics.

Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2020b. Specializing unsupervised pretraining models for word-Level semantic similarity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1371–1383, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael Laver, Kenneth Benoit, and John Garry. 2003. Extracting policy positions from political texts using words as data. *American political science review*, 97(2):311–331.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian. Bizer. 2012. Dbpedia–A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*.

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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. 2021. Parameter-Efficient Domain Knowledge Integration from Multiple Sources for Biomedical Pre-trained Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3855–3865, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press.

Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. 2021. Mixture-of-Partitions: Infusing Large Biomedical Knowledge Graphs into BERT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4672–4681, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stefano Menini, Federico Nanni, Simone Paolo Ponzetto, and Sara Tonelli. 2017. Topic-Based Agreement and Disagreement in US Electoral Manifestos. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2938–2944, Copenhagen, Denmark. Association for Computational Linguistics.

Slava Mikhaylov, Michael Laver, and Kenneth R. Benoit. 2012. Coder reliability and misclassification in the human coding of party manifestos. *Political Analysis*, 20(1):78–91.

MIT Election Data and Science Lab. 2017. U.S. House 1976–2020.

Nicole Peinelt, Marek Rei, and Maria Liakata. 2021. GiBERT: Enhancing BERT with Linguistic Information using a Lightweight Gated Injection Method. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2322–2336, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

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

Christian Rauh and Jan Schwalbach. 2020. The ParlSpeech V2 data set: Full-text corpora of 6.3 million parliamentary speeches in the key legislative chambers of nine representative democracies.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning Multiple Visual Domains with Residual Adapters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 506–516, Long Beach, California, USA.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Shivashankar Subramanian, Trevor Cohn, Timothy Baldwin, and Julian Brooke. 2017. Joint sentence-document model for manifesto text analysis. In *Proceedings of the Australasian Language Technology Association Workshop, ALTA 2017, Brisbane, Australia, December 6-8, 2017*, pages 25–33.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Naomi Truan. 2019. Parliamentary Debates on Europe at the Assemblée nationale (2002-2012) [Corpus].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Andrea Volkens, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, and Bernhard Weßels. 2019a. The Manifesto Data Collection. Manifesto Project (MRG/CMP/MARPOR). Version 2019a.

Andrea Volkens, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, and Bernhard Weßels. 2019b. The Manifesto Project Dataset - Codebook. Manifesto Project (MRG/CMP/MARPOR). Version 2019a.

Andrea Volkens, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, and Bernhard Weßels. 2019c. The Manifesto Data Collection. Manifesto Project (MRG/CMP/MARPOR). Version 2019b.

Andrea Volkens, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, and Bernhard Weßels. 2019d. The Manifesto Project Dataset - Codebook. Manifesto Project (MRG/CMP/MARPOR). Version 2019b.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*, 57(10):78–85.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Annika Werner, Onawa Lacewell, and Andrea Volkens. 2014. *Manifesto Coding Instructions: 5th fully revised edition*. Manifesto Project.

Annika Werner, Onawa Lacewell, Andrea Volkens, Theres Matthieß, Lisa Zehnter, and Leila van Rinsum. 2021. Manifesto Coding Instructions. 5th re-revised edition.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*, pages 535–546. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations. 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 165–176, Hong Kong, China. Association for Computational Linguistics.

Supplementary Material

A PolSciBERT

We list the collection of corpora utilized for pre-training PolSciBERT in Table 6. The raw texts have been split into sentences using Spacy (Version 2.3.1. https://spacy.io). Sentences without any lower-case Latin characters have been removed from the data.

A.1 Hyperparameters for PolSciBERT pre-training

The pre-training of PolSciBERT was continued on the political text corpus, using a batch size of 16. Note that the gradient accumulation step was set to be 4, meaning model weights were updated once every 4 batches. The learning rate was $5e - 05$. A new checkpoint was saved every 50,000 steps. We use the checkpoint at the 5950000-th step as our base model.

A.2 The KELM Corpus

```
{
  "triples": [
    ["Valentin Lavigne", "member of sports team", "FC Lorient"],
    ["Valentin Lavigne", "FC Lorient", "start time", "01 January 2014"],
    ["Valentin Lavigne", "FC Lorient", "end time", "01 January 2016"]
   ],
  "serialized_triples":
    "Valentin Lavigne member of sports team FC Lorient, FC Lorient"
    "end time 01 January 2016, FC Lorient start time 01 January 2014.",
  "gen_sentence":
    "Valentin Lavigne played for FC Lorient between 2014 and 2016."
}
```

Figure 1: An example instance in the KELM corpus

Language	Name	Time Period	# Tokens	Link
German	GermanParl (Blaette, 2017)	1996 - 2016	77,661,778	https://github.com/PolMine/GermaParlTEI
	Europarl-de (Koehn, 2005)	1996 - 2011	46,747,617	https://opus.nlpl.eu/Europarl-v3.php
	Austrian Nationalrat (Rauh and Schwalbach, 2020)	2003 - 2018	56,687,818	https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/L4OAKN
	Bundestag	2017 - 2020	11,542,765	https://www.bundestag.de/services/opendata
	Bundestag Barbaresi (Barbaresi, 2018)	1982 - 2017	11,257,316	https://politische-reden.eu
English	Europarl-en (Koehn, 2005)	1996 - 2011	48,984,323	https://opus.nlpl.eu/Europarl-v3.php
	NZ House of Representatives (Rauh and Schwalbach, 2020)	1996 - 2016	135,135,640	https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/L4OAKN
	UK House of Commons (Rauh and Schwalbach, 2020)	1989 - 2019	361,921,136	https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/L4OAKN
	US Congressional Record (MIT Election Data and Science Lab, 2017)	1989 - 2010	439,913,096	https://www.bundestag.de/services/opendata
Spanish	Europarl-es (Koehn, 2005)	1996 - 2011	54,617,946	https://opus.nlpl.eu/Europarl-v3.php
	Congreso de los Disputados (Rauh and Schwalbach, 2020)	1996 - 2018	66,395,968	https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/L4OAKN
French	Europarl-fr (Koehn, 2005)	1996 - 2011	54,956,800	https://opus.nlpl.eu/Europarl-v3.php
	Assemblee Nationale (Truan, 2019)	2002 - 2012	113,765	https://www.ortolang.fr/market/corpora/fr-parl/5
	TAPS Assemblée Nationale	2017 - 2020	30,415,252	https://data.assemblee-nationale.fr/travaux-parlementaires/debats
Italian	Europarl-it (Koehn, 2005)	1996 - 2011	50,488,760	https://opus.nlpl.eu/Europarl-v3.php
	Camera	2008 - 2020	68,419,585	https://www.camera.it

Table 6: Source and links to the pre-trained corpora for PolSciBERT

B Training Details

B.1 Baseline models

We perform downstream fine-tuning for all model setups with a batch size of 16 and a linear learning rate decay and use AdamW (Loshchilov and Hutter, 2019) as optimizer. The learning rate is $5e^{-5}$ for the baseline models and adapter full fine-tuning. The maximum number of epochs is 30, with early stopping and a patience of 5, meaning the model will stop training if the evaluation results on the development set stop improving for 5 consecutive epochs.

B.2 Adapters

The training arguments and configurations for the adapters are presented in Table 7. Following the settings in Pfeiffer et al. (2021), all adapters are trained with a learning rate of $1e-4$ with linear learning rate decay. The warm-up ratio is 0.1. We train adapters for different batch sized and number of epochs, depending on the size of the training data. For CN-SIMILARITY and CN-ANTONYM, we perform early stopping based on the accuracy on the test set: If the accuracy stops improving for 5 consecutive evaluation steps, the training is stopped. We use AdamW (Loshchilov and Hutter, 2019) with a weight decay of 0.01 for optimization.

C Results on the Koh et al. (2021) dataset

To compare our results with previous work, we downloaded the data from Koh et al. (2021) from their github repository.[11] We found that, probably due to some preprocessing problem, the quasi-sentences in the dataset were not complete (see examples below). For a fair comparison, we proceeded as follows. First, we trained and tested our models on the original dataset of Koh et al. (2021), to assure that differences in results are not simply due to the missing tokens. We used the same train/test splits as specified in the data. Next, in order to evaluate the impact of the missing tokens on the results, we downloaded English manifestos from the Manifesto Project homepage[12] and recreated the dataset with manifestos from Australia, Canada, Ireand, New Zealand, South Africa, the UK and the US (Table 3). Our new dataset is substantially smaller than the original dataset and we did not balance the label distribution across the different splits. To ensure replicability, we will make our train/dev/test splits available upon publication.

A	Once people have what
B	Once people have offended, what next?
A	The manifesto is
B	The manifesto is comprehensive.
A	not has turned things
B	Choice, not chance, has turned things round.

Figure 2: Examples for missing tokens in the dataset (A: quasi-sentence taken from Koh et al.; B: recreated from the original manifestos data).

	lang	train	dev	test
Koh et al.	(EN)	69,500	14,888	14,894
recreated	(EN)	59,559	14,419	13,722
zero-shot	(DE)	–	–	88,694

Figure 3: Statistics for the recreated manifestos dataset (en) and for the German test set used for zero-shot prediction.

[11] https://github.com/allisonkoh/bertcnn-classifying-manifestos, (file: 02.FINAL_minor.csv).

[12] https://manifesto-project.wzb.eu

	CN-SIMILARITY	CN-ANTONYM	KELM-ADAP
Training Arguments			
batch size	32	32	16
number of epochs	10	30	1
learning rate	1e-4	1e-4	1e-4
warm-up ratio	0.1	0.1	0.1
weight decay	0.01	0.01	0.01
early stopping	True	True	False
patience	5	5	5
evaluation steps	15000	500	15000
gradient accumulation steps	1	1	4
Adapter Configurations			
adapter hidden size	96	96	96

Table 7: Training details for adapter training.

Model Setup		**Major Topics**		**Minor Topics**	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1
(Koh et al., 2021)	BERT-GRU (Base model)	59.3	47.9	43.2	23.9
	BERT-CNN (Base model)	59.1	47.3	44.8	26.0
Baselines	XLM-R	61.7$_{(0.4)}$	50.6$_{(1.0)}$	48.6$_{(0.2)}$	34.1$_{(1.7)}$
	PolSciBERT	63.2$_{(0.2)}$	51.9$_{(0.7)}$	50.5$_{(0.3)}$	36.1$_{(0.7)}$
CN-SYNONYM	Full	61.7$_{(0.3)}$	52.0$_{(0.3)}$	49.4$_{(0.3)}$	37.4$_{(1.0)}$
	AdapterStack	63.5$_{(0.1)}$	52.2$_{(0.5)}$	51.0$_{(0.2)}$	37.7$_{(1.1)}$
	AdapterFusion	63.5$_{(0.2)}$	52.3$_{(0.5)}$	50.4$_{(0.2)}$	35.7$_{(2.3)}$
KELM-ADAP	Full	62.3$_{(0.2)}$	52.5$_{(0.4)}$	49.9$_{(0.5)}$	37.9$_{(0.5)}$
	AdapterStack	**63.8**$_{(0.2)}$	**53.5**$_{(0.3)}$	**51.2**$_{(0.2)}$	**38.5**$_{(1.0)}$
	AdapterFusion	63.5$_{(0.2)}$	52.2$_{(0.1)}$	50.8$_{(0.2)}$	37.0$_{(1.6)}$
CN-SYNONYM & KELM-ADAP	AdapterStack	62.8$_{(0.5)}$	51.5$_{(0.9)}$	50.6$_{(0.3)}$	37.0$_{(0.7)}$
	AdapterFusion	63.6$_{(0.2)}$	52.3$_{(0.2)}$	50.8$_{(0.3)}$	37.7$_{(1.4)}$
Experiments with semantic dissimilarity knowledge					
CN-SYNONYM & CN-ANTONYM	AdapterStack	62.1$_{(0.8)}$	50.7$_{(1.0)}$	50.3$_{(1.1)}$	36.3$_{(2.2)}$
	AdapterFusion	63.6$_{(0.2)}$	52.5$_{(0.3)}$	50.7$_{(0.3)}$	37.7$_{(0.6)}$
CN-SYNONYM & CN-ANTONYM & KELM-ADAP	AdapterStack	61.9$_{(0.4)}$	50.3$_{(0.6)}$	51.0$_{(0.2)}$	37.8$_{(0.8)}$
	AdapterFusion	63.5$_{(0.3)}$	52.3$_{(0.2)}$	50.9$_{(0.1)}$	38.4$_{(0.9)}$

Table 8: Test set results for manifesto quasi-sentence policy domain classification (Koh et al., 2021). The results for Koh et al. (2021) were taken from Table 7 in their paper. The first column specifies the model setup, including the knowledge adapter(s) and the fine-tuning strategy applied. All evaluation metrics reported for our model setups were averaged over 5 random initializations. The numbers in parentheses indicate standard deviation over the 5 runs.