Teach the Rules, Provide the Facts: Targeted Relational-knowledge Enhancement for Textual Inference

Ohad Rozen[1] **Shmuel Amar**[1] **Vered Shwartz**[2,3] **Ido Dagan**[1]
[1]Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel
[2]Allen Institute for Artificial Intelligence
[3]Paul G. Allen School of Computer Science & Engineering, University of Washington
{ohadrozen,shmulikamar}@gmail.com, vereds@allenai.org, dagan@cs.biu.ac.il

Abstract

We present InferBert, a method to enhance transformer-based inference models with relevant relational knowledge. Our approach facilitates learning generic inference patterns requiring relational knowledge (e.g. inferences related to hypernymy) during training, while injecting on-demand the relevant relational facts (e.g. *pangolin* is an *animal*) at test time. We apply InferBERT to the NLI task over a diverse set of inference types (hypernymy, location, color, and country of origin), for which we collected challenge datasets. In this setting, InferBert succeeds to learn general inference patterns, from a relatively small number of training instances, while not hurting performance on the original NLI data and substantially outperforming prior knowledge enhancement models on the challenge data. It further applies its inferences successfully at test time to previously unobserved entities. InferBert is computationally more efficient than most prior methods, in terms of number of parameters, memory consumption and training time.

1 Introduction

Transformer-based pre-trained language models (LMs), such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) have recently achieved human-level performance on standard natural language inference (NLI) benchmarks (Wang et al., 2019). However, the performance on this complex task is achieved in part thanks to large training sets that facilitate learning of dataset-specific biases and correlations, and thanks to the similar distributions between the training and test sets, that rewards such models (Poliak et al., 2018; Gururangan et al., 2018). This contrasts with humans, who can learn a generalized solution from fewer examples (Linzen, 2020). Indeed, NLI models often fail on examples involving various linguistic phenomena such as co-hyponymy (Glockner et al., 2018) and negation (Naik et al., 2018), which they are expected to acquire indirectly from the NLI training set.

Prior work proposed to provide ("inoculate") NLI models with a small number of phenomenon-specific training examples in order to teach the model to address them (Liu et al., 2019a). However, Rozen et al. (2019) showed that when the distributions of the training and test sets differ with respect to syntactic and lexical properties, the performance of such inoculated models drops, concluding that they do not learn a generalized notion of the phenomenon. In this paper we are motivated by the following question: *how can we facilitate learning of generalized inference patterns, with respect to a given linguistic phenomenon, from a relatively small number of examples?*

Ideally, we would like an NLI model to learn inference patterns detached from their original context, and to be able to apply them in new contexts involving different concrete facts. For example, an NLI model may learn that a word entails its hypernym in upward monotone sentences from training examples such as: *Alice ate a* <u>*banana*</u> → *Alice ate a* <u>*fruit*</u>. Then, to be able to apply this rule to a test instance with the premise *Bob saw a* <u>*pangolin*</u> and the hypothesis *Bob saw an* <u>*animal*</u>, it needs to know that *animal* is a hypernym of *pangolin*. Training a model on every possible hyponym-hypernym pair is incredibly inefficient and requires re-training a model whenever the vocabulary expands. Instead, we propose to *decouple the learning of generic inference patterns from that of the factual knowledge*.

To that end, we develop InferBert, a method to enhance language models with *relational* knowledge from a knowledge base (KB). In contrast to recent knowledge-enhancement approaches such as KnowBert (Peters et al., 2019) and Ernie (Zhang et al., 2019) that incorporate into LMs knowledge about individual entities (e.g. *pangolin*), we inform

Proceedings of the 10th Conference on Lexical and Computational Semantics, pages 89–98
August 5–6, 2021, Bangkok, Thailand (online) ©2021 Association for Computational Linguistics

the LM of the *relation* between a pair of entities that are involved in an inference instance, e.g. `Hypernym`(*pangolin*) = *animal*. This approach is agnostic to the identity of the specific entities, allowing models to learn inference patterns separately from the individual facts involved in particular instances.

To evaluate the ability of NLI models to learn inference patterns for specific linguistic phenomena, we follow the evaluation approach taken in previous work (Naik et al., 2018; Liu et al., 2019a; Richardson et al., 2020), which demonstrated the learning ability of models over a few chosen inference phenomena. We focus on 4 target semantic relations: hypernymy, location, country of origin, and color, for which we create challenge sets[1] (see Table 1 for examples). We construct the challenge sets such that there is no overlap between the training, validation, and test sets with respect to the target entities (e.g. *pangolin*), to allow testing whether the model had learned an inference phenomenon in a generic manner, rather than performing lexical memorization. The training sets are deliberately small (660-960 instances), aiming to challenge models with learning from a relatively small number of examples per semantic phenomenon.

Our results confirm that InferBert manages to generalize inference patterns to new facts, substantially improving performance on the challenge sets upon the knowledge-enhanced baselines (up to +17.5 points in accuracy from the next best model), all while maintaining the performance on the original MultiNLI test set (Williams et al., 2018).

Moreover, InferBert not only learns from a small number of training examples (which are insufficient for the baselines), it is also considerably more efficient than prior knowledge-enhanced LMs in terms of training time, resources, and memory. InferBert doesn't require LM pre-training, which is a computationally expensive process, and doesn't embed entities, only a small number of relations, substantially reducing the number of parameters with respect to some of the prior work (e.g. only 23% of KnowBert's parameters).

Finally, while InferBert is demonstrated on NLI, it is a general method and may benefit additional tasks such as question answering and co-reference resolution which may rely on relational knowledge between words in given instances.

Hypernymy	
P:	He killed another **jay** this season.
H:	He took life away from a **bird** this season.
Label:	Entailment
Relation:	`Hypernym`(jay)= bird
Location	
P:	It is not located in **Baytown**.
H:	It is located in all cities in **Texas** except for one.
Label:	Neutral
Relation:	`LocationOf`(Baytown)= Texas
Color	
P:	Tommy ordered tea and apricots.
H:	Tommy did not order any dark brown fruits.
Label:	Neutral
Relation:	none*
Country of Origin	
P:	**Viesgo** deal, from beginning to end, took less than five weeks.
H:	The minimum amount of time it has ever taken a **Spanish** company to close a deal is six weeks.
Label:	Contradiction
Relation:	`CountryOfOrigin`(Viesgo) = Spain

Table 1: An example from each phenomenon-specific challenge set. *By design, for half of the examples there is no corresponding relation (See Section 3.2).

2 Related Work

2.1 Probing NLI Models

In natural language inference (NLI; Bowman et al., 2015), originally referred as recognizing textual entailment (RTE; Dagan et al., 2013), the goal is to determine whether a first text unit (premise) entails, contradicts, or is neutral with respect to a second text (hypothesis). The decision involves various syntactic and semantic phenomena, including lexical and world knowledge, coreference resolution, geographical reasoning, etc. (Clark, 2018). While neural models have achieved human performance on the GLUE and SuperGLUE benchmarks (Wang et al., 2018, 2019), the success of such models is often due to learning non-generalizable dataset-specific patterns (Poliak et al., 2018; Gururangan et al., 2018; McCoy et al., 2019).

Various challenge sets were developed to test the capabilities of state-of-the-art NLI models in addressing specific semantic phenomena. For example, Glockner et al. (2018) showed that substituting a single term in the premise with a similar but mutually-exclusive term (e.g. *guitar* and *piano*) confused NLI models that predicted entailment. Naik et al. (2018) further showed that NLI models perform poorly on examples involving antonyms, numerical reasoning, and distractions

[1] All datasets and resources are available at https://github.com/ohadrozen/inferbert.

such as high lexical overlap and spelling errors. NLI models also struggled with examples involving logic and monotonicity (Richardson et al., 2020; Yanaka et al., 2020; Geiger et al., 2020).[2] Finally, the GLUE benchmark dedicated a small set for diagnosing models' strengths and weaknesses on various phenomena (Wang et al., 2018).

Liu et al. (2019a) suggested that NLI models may perform poorly on specific phenomena they haven't observed enough during training, and proposed to "inoculate" LM-based models against challenge sets by fine-tuning them on a small number of phenomenon-specific training instances. Rozen et al. (2019) showed that the inoculation does not necessarily teach the model a generalized notion of the phenomenon of interest, and that when the challenge test set differs from the corresponding training sets in terms of, for example, syntactic complexity, the performance of the inoculated models drops. Richardson et al. (2020) highlighted the sensitivity of the inoculation training to hyper-parameters, that may result in "catastrophic forgetting", i.e. a substantial drop in performance on the original NLI task.

2.2 Knowledge-Enhanced Models

There is plenty of work on incorporating knowledge from KBs into neural models. Knowledge-based Inference Model (KIM; Chen et al., 2018) incorporated semantic relations from WordNet into an RNN-based NLI model, gaining a modest improvement on a challenge set. The incorporation at various components of the original NLI model is not straightforward to adapt to other models.

KnowBert (Peters et al., 2019) incorporated knowledge from Wikipedia and WordNet into a BERT model through entity embeddings, improving performance on relation extraction and entity typing. Ernie (Zhang et al., 2019) and K-Adapter (Wang et al., 2020a) both targeted similar downstream tasks. Ernie embeds entities and relations from a KB, and alters the BERT pre-training to predict entities in addition to words. K-Adapter does not re-train the LM weights, but takes a somewhat more efficient approach of training an additional neural component ("adapter") for each knowledge type as a plug-in for the LM. KEPLER (Wang et al., 2020b) learns entity embeddings from their textual descriptions. These entity-centric methods require *pre-training* the original LM or its plugins on the KB, while increasing training time and cost and storing the entity embeddings (increasing memory cost). In addition, by design, the knowledge can capture only entities seen during pre-training, thus requiring repeating the pre-training process each time the original input KB gets updated.

Finally, K-BERT (Liu et al., 2019b) is most similar to our model, incorporating knowledge regarding individual entities that occur in the input instance. Like our model, knowledge is augmented, per-instance, at inference time. Unlike our model, knowledge is augmented per entity, rather than per a relation between a pair of entities appearing in the inference instance. Further, K-BERT injects the KB knowledge in a textual form, which augments the input instance, while our model embeds directly structural knowledge. As we show in Section 6.1, this encoding is less effective than our structured incorporation method (Section 4.2), leading to weaker learning ability of different inference phenomena that require external knowledge.

3 Data

We focus on four types of semantic relations (Section 3.1), each corresponding to a set of *facts* in the form of semantic relation triplets. An NLI model may learn various *inference patterns* pertaining to the semantic relation type, such as "a word entails its hypernym in an upward monotone sentence".

To evaluate the models' ability to learn and apply these rules, we create an NLI challenge set for each semantic relation, that we derive from MultiNLI (Section 3.2). As usual, the goal is to determine the label of a premise-hypothesis pair (p, h) among entailment, neutral, and contradiction. For a given semantic relation, each instance in the corresponding challenge set requires applying an inference pattern associated with the semantic relation in order to determine the correct label (possibly along with other required inferences).

3.1 Semantic Relations

Hypernymy. An NLI system might learn that a term generally entails its generalization, for example "I ate an *apple*" entails "I ate a *fruit*".[3] The relevant facts for this semantic relation are pairs of (x, y) terms that appear in a direct or indirect

[2] The "Countries/Travel" genre in Richardson et al. (2020) is similar to our location phenomenon described in Section 3.1.

[3] The rule applies to upward monotone premises. Downward monotone premises (which typically include a negated predicate or certain quantifiers) reverse the inference direction.

KB entry: **Emporis** `CountryOfOrigin` (property): **Germany**

Extracted Premise: *These forms will be posted on **Apple** website.*
Premise: *These forms will be posted on **Emporis** website.*

Manually created hypotheses:
(1) *A company in **Germany** will make the forms available on its website.* (Entailment)
(2) *The forms cannot be accessed from the website of any **German** company.* (Contradiction)
(3) *Several **German** websites will feature the forms.* (Neutral)

Hypotheses with property replacement:
(4) *A company in **France** will make the forms available on its website.* (Neutral)
(5) *The forms cannot be accessed from the website of any **French** company.* (Neutral)
(6) *Several **French** websites will feature the forms.* (Neutral)

Table 2: Example of premise and hypotheses generation from a MultiNLI premise. Hypotheses (1)-(3) were created by crowdworkers for the altered premise, based on the Wikidata fact that Emporis' country of origin is Germany. Hypotheses (4)-(6) were created by replacing *German* with another country of origin (*France*) and annotated for entailment.

Inference Type	**Train**	**Dev**	**Test**	**All**
Hypernymy	960	114	300	1374
Location	660	114	230	1004
Color	840	108	318	1266
Country of Origin	834	114	252	1200
Total	3294	450	1100	4844

Table 3: Statistics of our challenge set.

hypernymy relation in WordNet (Miller, 1995).[4]

Location. A model may learn that in some contexts, substituting a city name by the state in which it is located yields a factually correct generalization (e.g. "John visited *Chicago*" entails "John visited *Illinois*"). We retrieve entities from Wikidata (Vrandečić and Krötzsch, 2014), focusing on US locations using the `state` property.

Color. We retrieve entities from Wikidata and their `color` property. We substitute an entity (e.g. *banana*) for a generalization involving its color and hypernym (e.g. *yellow fruit*).

Country of Origin. We retrieve knowledge from Wikidata about companies and their country of origin, using the `country` property. We substitute an entity (e.g. *Apple*) for a generalization involving its country of origin (e.g. *American* organization).

3.2 Challenge Sets

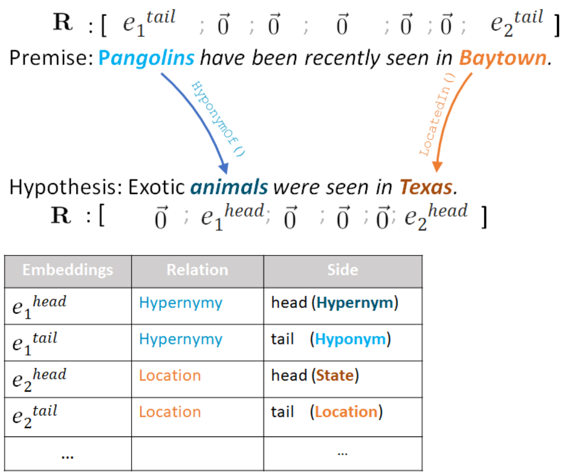
Some of the semantic relations we focused on are very rare in the original MultiNLI dataset, e.g. by heuristically searching for instances that exhibit these phenomena we found that less than 0.05% of the data contained locations. We therefore create challenge sets focusing on each semantic relation. In order to create challenge examples in a similar style and domain, we base our examples on premises in MultiNLI.

For a given semantic relation r, we extract premises in the MultiNLI training set that contain an entity I_0^{tail} whose type corresponds to the relation argument. For example, for the *country of origin* semantic relation we extract premises containing company names (e.g. $I_0^{tail} = Apple$) in our data. For a given premise p, we modify it by replacing I_0^{tail} by a random entity I_1^{tail} of the same type in the KB (e.g. *Emporis*), and manually check that the sentence still makes sense. We specifically select replacement entities I_1^{tail} such that there exists a KB assertion $R(I_1^{tail}) = I_1^{head}$. For example, `CountryOfOrigin`(*Emporis*) = *Germany*.

From each premise p we created 6 hypotheses as follows (See Table 2). Similarly to Williams et al. (2018), we showed p to crowdsourcing workers and asked them to generate a hypothesis for each label (entailment, neutral and contradiction). Our instructions further specified that the hypothesis must include I_1^{head} (e.g. *Germany*) but not I_1^{tail} (e.g. *Emporis*). Examples (1)-(3) in Table 2 demonstrate the instances created at this step.

After creating 3 hypotheses, all of which include I_1^{head} by design, we replaced I_1^{head} with $\hat{I}_1^{head}$, where $\hat{I}_1^{head} \neq I_1^{head}$ is a random value of the

[4] Excluding instance hypernyms.


R : [e_1^{tail} ; $\vec{0}$; $\vec{0}$; $\vec{0}$; $\vec{0}$; $\vec{0}$; e_2^{tail}]
Premise: Pangolins have been recently seen in Baytown.
HyponymOf ()
LocatedIn ()
Hypothesis: Exotic animals were seen in Texas.
R : [$\vec{0}$; e_1^{head} ; $\vec{0}$; $\vec{0}$; $\vec{0}$; e_2^{head}]

Embeddings	Relation	Side
e_1^{head}	Hypernymy	head (Hypernym)
e_1^{tail}	Hypernymy	tail (Hyponym)
e_2^{head}	Location	head (State)
e_2^{tail}	Location	tail (Location)
...		...

Figure 1: Relation embedding example. After extracting the related entity pairs for the relations r_1 =*hypernymy* and r_2 =*location*, we place the embedding vectors in $\mathbf{R}$ in the indices of the relevant tokens of the entities. For example, we place e_2^{head} in the index of *Texas* as this entity is the head in the relation *location*.

same property R (e.g. *France*). We then asked an annotator to label the new hypotheses with respect to p (Table 2, instances (4)-(6)).

The annotation task was performed using Amazon Mechanical Turk. To ensure the quality of the work, we required that workers had a minimum of 96% acceptance rate for prior HITs and pass a qualification test. We paid $1 for each premise. The test set was further validated by two trained annotators. The first annotator re-labeled an example, and, in case of disagreement with the original label (11.9% of the annotations), the second annotator also labeled the example, and the majority vote[5] was used.

Data Split. The statistics of the challenge sets are shown in Table 3. We split the datasets to 68%-9%-23% train, dev and test, respectively. The datasets are split lexically, i.e. such that head and tail entities in one set do not appear in the other sets. That way, a good performance on the test set indicates that the model learned a generalized notion of an inference rule rather than specific facts, and that it is capable of applying the rule when provided with the necessary yet not previously observed facts.

4 InferBert

We present InferBert, a BERT-based NLI model with a relational knowledge enhancement component. The key idea in InferBert is incorporating into the model relational knowledge (*facts*) from external knowledge resources regarding entities mentioned in the input instance. We adopt an inclusive definition of entity, which can refer either to a named entity (such as entries in Wikidata) or a common noun (such as WordNet lemmas).

As we discussed in Section 2.2, most prior work injects external knowledge into models through an entity's knowledge base embedding, which captures in a soft way its relationships with other KB entities. The limitation of such methods is the coupling of an inference pattern with the related factual knowledge. Suppose that a model observed during training that "*The boy ate an* $\underline{\textit{apple}}$" entails "*The boy ate a* $\underline{\textit{fruit}}$". The test example with the premise "*The woman has a* $\underline{\textit{dog}}$" and the hypothesis "*The woman has a* $\underline{\textit{pet}}$" is represented differently from the training example due to the distance between the entities (e.g. *apple* and *dog*) in the KB. Such a model is likely to fail on examples consisting of unseen entities.

We propose to decouple learning the *inference pattern* from the *facts* by directly embedding the semantic relations between entities in the text. In the above example, InferBert can access the KB during both the training and inference phases, and add an indicator that `fruit=Hypernym(apple)`. After observing enough training examples with the hypernym indicator, the model can learn a general rule like "a word entails its hypernym in certain common context". During inference, the model can apply this rule to unseen entities in the KB.

We first describe the KnowBert model (Peters et al., 2019, Section 4.1) which is the basis for InferBert. Next, we describe how we replace KnowBert's Knowledge Attention and Recontextualization component (KAR) by our Simplified KAR (S-KAR, Section 4.2).

4.1 KnowBert's KAR

KnowBert is a method to incorporate knowledge from KBs into transformer-based language models, which was specifically applied to BERTBASE. For a given input $X = (x_1, ..., x_N)$ of N word pieces, the BERT contextual embeddings are computed as $\mathbf{H_i} = \mathrm{TransformerBlock}(\mathbf{H_{i-1}})$ where $\mathbf{H_i} \in \mathbb{R}^{N \times D}$ is the i-th hidden layer ($i \in \{1, ..., L\}$, and $L = 12$ layers) and D is BERT's embedding dimension. TransformerBlock operates over a query, key, and

[5] All three annotations were given an equal weight.

value, and is defined as $\mathrm{TransformerBlock}(\mathbf{H_i}) = \mathrm{MLP}(\mathrm{MultiHeadAttn}(\mathbf{H_i}, \mathbf{H_i}, \mathbf{H_i}))$.

The Knowledge Attention and Recontextualization component (KAR) is added between BERT layers i and $i-1$, changing the embedding mechanism to: $\mathbf{H'_i} = \mathrm{KAR}(\mathbf{H_i}, \mathcal{C})$, which is computed as follows:

Retrieval: The KB entity candidate selector provides a list of C potential entity links for X, along with their mention spans in X.

Disambiguation: Each mention span is represented by applying self-attention pooling over all word pieces in the span (after projection to the entity embedding dimension E), yielding $S \in \mathbb{R}^{C\times E}$. To select the relevant entities in the context, mention-span self-attention is applied to compute $S^e = \mathrm{TransformerBlock}(S)$, followed by computing candidate entity scores ψ based on S^e.

Knowledge incorporation: The entity embeddings are averaged to $\tilde{e}$ based on their weight ψ, and are used to enhance the mention-span representations, yielding $S'^e = S^e + \tilde{e}$.

Recontextualization: The BERT word piece representations are recontextualized using a modified transformer layer in which S'^e is used as both the key and value for $\mathrm{MultiHeadAttn}$. The resulting vectors $\mathbf{H'_i}$ are projected back into the BERT dimension D.

4.2 S-KAR

The main component of InferBert is the Simplified Knowledge Attention and Recontextualization component (S-KAR). Rather than enhancing BERT with KB entity embeddings, InferBert embeds the KB relations.

Similarly to KAR, S-KAR replaces BERT's embedding mechanism between two particular layers, computing: $\mathbf{H'_i} = \text{S-KAR}(\mathbf{H_i}, \mathcal{C})$, which is then used to compute the next layer: $\mathbf{H_{i+1}} = \mathrm{TransformerBlock}(\mathbf{H'_i})$, and the remainder of BERT is run as usual. S-KAR operates as follows:

Retrieval: We follow KnowBert (Peters et al., 2019) and adopt a broad definition for a KB as a collection of (tail entity, relation, head entity) triplets, focusing on K relation types of interest: $\mathcal{R} = \{R_1, ..., R_K\}$. For each relation type R_k we learn two embedding vectors, e_k^{head} and e_k^{tail}, representing the head and the tail entity slots in this relation.[6]

We assume that for a given relation set $\mathcal{R}$, the KB is accompanied by a relation extractor, which takes a text X as input and returns a list of triplets:

$$\mathcal{C} = \{(\text{head}_m, \text{tail}_m, r_m) | m \in 1..|\mathcal{C}|, r_m \in \mathcal{R}\}$$

where head_m and tail_m are the indices of the first token of the head and tail entities in the text $(1, ..., N)$, and r_m is the relation, as illustrated in Figure 1.

Disambiguation: We focus on unambiguous entities, i.e. those with a single KB entry, with respect to relation type, and extract only entities of the relevant type.[7] For example, though *Pitcher* has multiple entries in Wikidata, only one of them is a location.

Knowledge incorporation: For a given list of triplets $\mathcal{C}$, S-KAR creates the relation embedding matrix $\mathbf{R} \in \mathbb{R}^{N\times E}$ such that the head embedding e_m^{head} is in index head_m, the related tail embedding vector e_m^{tail} is in index tail_m, and the remaining entries are set to $\vec{0}$. We incorporate this relation embeddings into the BERT vectors: $\mathbf{S'_i} = \mathbf{H_i^{proj}} + \mathbf{R}$, where $\mathbf{H_i^{proj}}$ is the projection of $\mathbf{H_i}$ into the relation embedding dimension $E = 768$.

Recontextualization: the recontextualization step is identical to KAR.

5 Experimental Setup

BERT model. Our model assumes access to a pre-trained BERT model with or without additional fine-tuning on the target downstream task. Specifically, we used the English uncased BERT_BASE model (Devlin et al., 2019) fine-tuned on the MultiNLI dataset (Williams et al., 2018). Based on preliminary experiments, the S-KAR layer was inserted between the first and second layers of BERT.

Relational data. We retrieve relational data from WordNet and Wikidata (See Section 3.1). For a given premise p and hypothesis h we retrieve a relevant KB tuple list of triplets $\{(\text{head}_m, \text{tail}_m, r_m)\}$ (Section 4.2) when the head is in the premise, tail is

[6] We did not explore symmetric relations in this work, but they can be straightforwardly implemented by learning a single vector for both entity slots of the relation.

[7] We use spaCy NER (Honnibal and Montani, 2017) to extract the relevant entity types: LOC for locations, ORG for names of companies, and nouns for hyponyms and colors.

	Model	Entities	Hypernymy	Location	Color	Origin	MultiNLI*
LM-based Model	BERT	seen	64.7	77.6	62.2	70.6	-
		unseen	65.7	68.3	58.5	69.1	83.4
Knowledge-Enhanced Models	KnowBert	seen	74.0	83.5	67.2	78.2	-
		unseen	66.7	69.1	59.1	70.6	82.3
	K-BERT	seen	68.0	81.7	62.2	75.0	-
		unseen	68.3	67.6	56.6	71.4	83.2
	InferBert	seen	81.7	83.3	**77.2**	86.9	-
		unseen	**82.0**	**83.9**	72.0	**88.9**	82.3

Table 4: Performance on the challenge test sets and MultiNLI. Models were tested on either entities that appear in the training set (**seen**) or new entities (**unseen**). *Seen and unseen results are not relevant for MultiNLI.

in the hypothesis, and head $\neq$ tail. Since we focus on unambiguous entities (in the context of a given relation), we do not need to use an entity linker. We make sure that the target entities in the train, validation, and test sets are distinct, but that they all have entries in the relevant KB.

Training data. We train a single model on the combination of the challenge sets to learn phenomena related to all the semantic relations. To avoid "catastrophic forgetting", i.e. decrease in the performance on the original task (MultiNLI), we mix the challenge training set with a random sample of 10K MultiNLI training set instances and train on the mixed datasets. The training objective assigns more weight to the challenge examples: $\mathcal{L}'_{\text{BERT}} = \gamma \cdot \mathcal{L}_{\text{BERT}}$, where $\gamma > 1$ is a hyperparameter fine-tuned on the validation set.

Training procedure. The model consists of a pre-trained BERT model and randomly initialized InferBert parameters (S-KAR weights and relation embeddings). To embed both sets of parameters in the same space, we follow KnowBert and train the model in two phases. In the first phase, we freeze the pre-trained BERT parameters and update only the S-KAR and the relation embeddings for 3 epochs. In the second phase we freeze the recently trained InferBert parameters and unfreeze the BERT parameters, training for another epoch.[8]

Baselines. We compare InferBert with two representative knowledge-informed models, KnowBert and K-BERT, as well as a BERT$_{\texttt{BASE}}$ NLI model. All the baselines are trained on MultiNLI and further fine-tuned on the the joint challenge set (mixed with a subset of MultiNLI).

For fair comparison, K-BERT used the same entity extraction mechanism, followed the same fine-tuning procedure, and was given access to the exact same data as InferBert. KnowBert, on the other hand, requires re-training a new model on new data. Because of its resource requirements, we used the available pre-trained KnowBert model. It is enhanced with knowledge about 470K entities from Wikipedia and all of WordNet, fully covering the knowledge in our hypernymy and location challenge sets, but only some of the entities in the color and country of origin sets[9]. Thus, the results for KnowBert on these two phenomena are not fully comparable to those of InferBert.

Hyper-parameters. Fine-tuning on MultiNLI followed the original hyper-parameters described in Devlin et al. (2019). When fine-tuning InferBert on the challenge sets, we selected the best hyper-parameter values based on the performance on the validation sets. The learning rate for S-KAR was chosen between 0.003-0.007 in steps of 0.001, and was set to 0.006. The rest of the parameters were trained with a learning rate of 9e-6 (selected between 3e-6 and 4e-5). We tested γ values among $\{2, 4, 6, 8, 10, 12\}$ and selected $\gamma = 4$. Fine-tuning was done on a single GeForce GTX 1080 GPU with batch size of 32. A single InferBert forward and backward pass took 0.35 seconds. K-BERT's best validation performance was achieved after 3 epochs with a learning rate of 3e-5 and KnowBert's after 4 epochs with a learning rate of 2e-5.

6 Experiments

We present the results of InferBert and the baselines on the various challenge sets (Section 6.1). We also test the ability of models to learn relational knowledge about entities seen during training (Section 6.2). Finally, we analyze InferBert's efficiency

[8] The first phase of KnowBert trains only the entity embeddings but not KAR, while we also include the S-KAR weights.

[9] All entities in our hypernymy challenge set are covered in WordNet, and all entities in our location set has corresponding entries in the Wikipedia subset used by KnowBert.

in terms of memory and runtime compared to the baselines (Section 6.3).

6.1 Performance on the Challenge Sets

Table 4 (“unseen" lines) shows the performance of InferBert and the baselines on the various challenge sets and on the MultiNLI[10] development set. The knowledge-enhanced baseline models slightly outperform BERT on all semantic relations. InferBert performs the best, with a large gap from the baselines (up to 20 points), demonstrating its ability to learn and generalize inference patterns and apply them to new relation instances, as well as to new entities.

K-BERT performs slightly better than KnowBert, yet worse than InferBert. We hypothesize that K-BERT and InferBert enjoy the advantage of having access to relational knowledge at inference time, which facilitates learning general inference patterns and applying them to new facts, on demand. With that said, the K-BERT method of incorporating relational knowledge as free text is less structured and likely leads to less efficient learning of inference patterns (with the limited amount of available training data).

InferBert retains a high performance on the MultiNLI matched development set, with 2.3% reduction from the original BERT$_{\text{BASE}}$ model (84.6%). KnowBert achieve similar performance, while K-BERT performs slightly better on it.

6.2 Seen vs. Unseen Entities

In contrast to our original test sets, in which the entities has not been seen during training, in this experiment we analyze how the models perform with entities that were all seen in the challenge training set. For that, we duplicated our test sets, while replacing test triplets (head, tail, relation) with others that are included in the training set. The rest of the words remained the same, and we made sure (manually) that the new test examples are analogously sensible and that their entailment labels have not changed. Results are shown in Table 4 in the *seen* rows. Evidently, InferBert shows impressive robustness when facing unseen entities, unlike other models that seem to depend significantly on seeing the test entities already in training time. In fact, when faced with new entities, the other models performance gets closer to that of original BERT (with no knowledge injected).

6.3 Efficiency Analysis

While large language models lead to performance boost on standard benchmarks, the NLP community had begun paying more attention to developing more resource-efficient NLP models (Moosavi et al., 2020). In the design of InferBert we took efficiency into consideration. First, InferBert is significantly less memory consuming than KnowBert, which stores up-front the embeddings for all entities in memory. KnowBert trained on Wikipedia and WordNet uses BERT$_{\text{BASE}}$ (110M parameters), to which it adds the KAR component (7.3M) and the embeddings of 471K entities (406M parameters), resulting in 523.3M parameters. Conversely, instead of entity embeddings, InferBert supports up to $K = 500$ relation types × 2 vectors (tail and head) × each with dimension $E = 768$, resulting in 768K parameters. The SKAR component takes up 8.3M parameters. Overall, InferBert has 119.1 parameters, only 23% of KnowBert's parameters. Second, as opposed to InferBert, KnowBert required a pre-training step in which the 471K instances (corresponding to KB entities) were processed.

InferBert achieved better performance than KnowBert on the challenge set with as little as 1,000 examples per relation (Table 4). We conjecture that the InferBert training is more data efficient as it is not required to learn about specific head or tail entities (e.g. *Emporis* and *Germany*) but about relationships, (e.g. `Hypernymy`) which occur more frequently in the training data.

Finally, we note that, similar to our model, K-BERT is also memory and parameter efficient since it does not store entity embeddings (as KnowBert does). Rather, it only involves fine-tuning the BERT parameters, thanks to representing the enhanced knowledge in textual form as part of the instance input. Our model does incorporate a modest number of additional parameters for structured relation embeddings, which, as shown in our experiment, leads to substantial performance gains over K-BERT's textual representations.

7 Conclusions

We presented InferBert, a generic and efficient method to incorporate relational knowledge into transformer-based inference models. Our approach targets specific inference phenomena that require external relational knowledge, allowing the model to learn generic inference patterns decoupled from

[10] We used MultiNLI dev-matched.

the factual knowledge required for a particular instance, which is injected at inference time. Our experiments show that InferBert successfully applies the learned patterns to unseen facts, where other knowledge enhancement models fail. Unlike most prior work, InferBert does not require pre-training the LM on a KB, and consumes less memory.

Our work joins the effort of others to improve models by teaching them specific inference phenomena (Liu et al., 2019a; Richardson et al., 2020). A natural direction for future work would be to apply our methodology to a broader range of inference phenomena and adapt them for additional inference tasks.

Acknowledgments

The work described herein was supported in part by grants from Intel Labs, Facebook, the Israel Science Foundation grant 1951/17, the Israeli Ministry of Science and Technology and the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia. Association for Computational Linguistics.

Peter Clark. 2018. What knowledge is needed to solve the rte5 textual entailment challenge? *arXiv preprint arXiv:1806.03561*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

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.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. 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 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019b. K-BERT: enabling language representation with knowledge graph. *CoRR*, abs/1909.07606.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41.

Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavaš, Shafiq Joty, Alex Wang, and Thomas Wolf, editors. 2020. *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics, Online.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. 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.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*.

Kyle Richardson, Hai Hu, Lawrence S Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *AAAI*.

Ohad Rozen, Vered Shwartz, Roee Aharoni, and Ido Dagan. 2019. Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Neurips*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020a. K-adapter: Infusing knowledge into pre-trained models with adapters.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2020b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *TACL*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117, Online. 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.