Infusing Knowledge into the Textual Entailment Task Using Graph Convolutional Networks

**Pavan Kapanipathi,[†] Veronika Thost,[*†] Siva Sankalp Patel,[†] Spencer Whitehead,[§]
Ibrahim Abdelaziz,[†] Avinash Balakrishnan,[†] Maria Chang,[†] Kshitij Fadnis,[†]
Chulaka Gunasekara,[†] Bassem Makni,[†] Nicholas Mattei,[‡] Kartik Talamadupula,[†] Achille Fokoue[†]**
[†]IBM Research, [*]MIT-IBM Watson AI Lab, [§]University of Illinois at Urbana-Champaign, [‡]Tulane University
{kapanipa, avinash.bala, kpfadnis, krtalamad, achille}@us.ibm.com
{veronika.thost, siva.sankalp.patel, ibrahim.abdelaziz1, maria.chang, chulaka.gunasekara, bassem.makni}@ibm.com
srw5@illinois.edu
nsmattei@tulane.edu

Abstract

Textual entailment is a fundamental task in natural language processing. Most approaches for solving this problem use only the textual content present in training data. A few approaches have shown that information from external knowledge sources like knowledge graphs (KGs) can add value, in addition to the textual content, by providing background knowledge that may be critical for a task. However, the proposed models do not fully exploit the information in the usually large and noisy KGs, and it is not clear how it can be effectively encoded to be useful for entailment. We present an approach that complements text-based entailment models with information from KGs by (1) using Personalized PageRank to generate contextual subgraphs with reduced noise and (2) encoding these subgraphs using graph convolutional networks to capture the structural and semantic information in KGs. We evaluate our approach on multiple textual entailment datasets and show that the use of external knowledge helps the model to be robust and improves prediction accuracy. This is particularly evident in the challenging BreakingNLI dataset, where we see an absolute improvement of 5-20% over multiple text-based entailment models.

1 Introduction

Given two natural language sentences, a premise P and a hypothesis H, the textual entailment task – also known as natural language inference (NLI) – consists of determining whether the premise entails, contradicts, or is neutral with respect to the given hypothesis (MacCartney and Manning 2009). In practice, this means that textual entailment is characterized as either a three-class (ENTAILS/NEUTRAL/CONTRADICTS) or a two-class (ENTAILS/NEUTRAL) classification problem (Bowman et al. 2015; Khot, Sabharwal, and Clark 2018).

Performance on the textual entailment task can be an indicator of whether a system, and the models it uses, are able to reason over text. This has tremendous value for modeling the complexities of human-level natural language understanding, and in aiding systems tuned for downstream tasks such as question answering (Harabagiu and Hickl 2006).

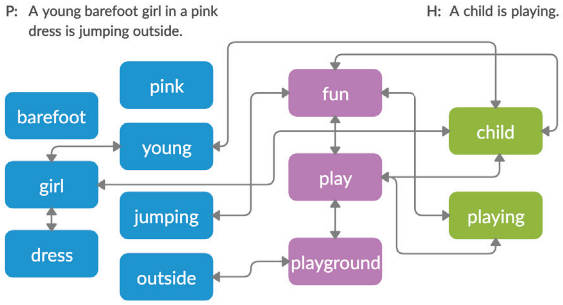

Figure 1: A premise and hypothesis pair along with a relevant subgraph from ConceptNet. Blue concepts occur in the premise, green in the hypothesis, and purple connect them.

Most existing textual entailment models focus only on the text of premise and hypothesis to improve classification accuracy (Parikh et al. 2016; Liu et al. 2019a). A recent and promising line of work has turned towards extracting and harnessing relevant semantic information from knowledge graphs (KGs) for each textual entailment pair (Chen et al. 2018; Wang et al. 2019). These approaches map terms in the premise and hypothesis text to concepts in a KG, such as Wordnet (Miller 1995), ConceptNet (Speer, Chin, and Havasi 2017), or DBpedia (Auer et al. 2007) and use these mapped concepts for the textual entailment task. Figure 1 shows an example of such mapping, where select terms from the premise and hypothesis are mapped to concepts from a knowledge graph (blue and green nodes, respectively). However these models suffer from one or more of the following drawbacks: (1) they do not possess the ability to explicitly capture the semantic and structural information from the KG. For example, in Figure 1, the ability for models to encode information from paths between blue and green nodes via purple nodes provides better context facilitating the system to more correctly judge entailment.; (2) they are not easily integrated with existing NLI models that exploit only the text of the premise and hypothesis; and (3) they are not flexible with respect to the type of KG that is used.

Contributions: We present an approach to the NLI problem that can augment any existing text-based entailment model with external knowledge. We specifically address the aforementioned challenges by: (1) introducing a neighbor-based expansion strategy in combination with subgraph filtering using Personalized PageRank (PPR) (Jeh and Widom 2003). This approach reduces noise and selects contextually relevant subgraphs from larger external knowledge sources for premise and hypothesis texts ; (2) encoding subgraphs using Graph Convolutional Networks (GCNs) (Kipf and Welling 2017), which are initialized with knowledge graph embeddings to capture structural and semantic information. This general approach to graph encoding allows us to use any external knowledge source that can be represented as a graph such as WordNet, ConceptNet, or DBpedia. We show that the additional knowledge can improve textual entailment performance by using four standard benchmarks: SciTail, SNLI, MultiNLI, and BreakingNLI. In particular, our experiments on the BreakingNLI dataset, where we see an absolute improvement of 3-20% over four text-based models, shows that our technique is robust and resilient.

2 Related Work

We categorize the related approaches for NLI into: (1) approaches that take only the premise and hypothesis text as input, and (2) approaches that utilize external knowledge.

Neural models focusing solely on the textual information (Wang and Jiang 2016a; Yang et al. 2019) explore the sentence representations of premise structure and max pooling layers. Match-LSTM (Wang and Jiang 2016a) and Decomposable Attention (Parikh et al. 2016) learn cross-sentence correlations using attention mechanisms, where the former uses a asymmetric network structure to learn premise-attended representation of the hypothesis, and the latter a symmetric attention, to decompose the problem into sub-problems. Latest NLI models use tranformer architectures such as BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019b). These models perform exceedingly well on many NLI leaderboards (Zhang et al. 2018; Liu et al. 2019a). In this work, we show that performance of text-based entailment models that use pre-trained BERT embeddings can be augmented with external knowledge.

Utilizing external knowledge has shown improvement in performance on many natural language processing (NLP) tasks (Huang et al. 2019; Moon et al. 2019; Musa et al. 2019). Recently, for NLI, Li et al. (2019) have shown that features from pre-trained language models and external knowledge complement each other. However, approaches that do utilize external knowledge for NLI are very few (Wang et al. 2019; Chen et al. 2018). In particular, the best model of Wang et al. (2019) combines rudimentary node information – in the form of concepts mentioned in premise and hypothesis text (blue and green nodes in Figure 1) – along with the text information. However, this approach misses the rich subgraph structure that connects premise and hypothesis entities (purple nodes in Figure 1). (Chen et al. 2018) have developed a model with WordNet based co-attention that use five engineered features from WordNet for each pair of words from premise and hypothesis. This model being tightly integrated with WordNet has the following drawbacks: (1) it is inflexible to be used with other external knowledge sources such as ConceptNet or DBpedia, and (2) it is non-trivial to be integrated with other state of the art text-based entailment systems. This work addresses the drawbacks of each of these approaches mentioned above with competitive performance on many NLI datasets.

The availability of large-scale datasets (Bowman et al. 2015; Williams, Nangia, and Bowman 2018; Khot, Sabharwal, and Clark 2018) has fostered the advancement of neural NLI models in recent years. However, it is important to discuss the characteristics of these datasets to understand what they intend to evaluate (Glockner, Shwartz, and Goldberg 2018). Particularly, datasets such as (Bowman et al. 2015; Khot, Sabharwal, and Clark 2018; Williams, Nangia, and Bowman 2018) contain language artifacts as significant cues for text-based neural models. These artifacts bias the models and makes it harder to evaluate the impact of external knowledge (Chen et al. 2018; Wang et al. 2019). In order to evaluate approaches that are more robust and not susceptible to such biases, Glockner, Shwartz, and Goldberg (2018) created BreakingNLI – an adversarial test set where most of the common text-based approaches show significant drop in performance. It is important to note that this test set is generated using a subset of relationships from online resources for English learning, making it more suitable for models exploiting KGs with lexical focus, such as WordNet. However, BreakingNLI represents a first and important step in the evaluation of models that utilize external knowledge sources.

One of the core contributions of this work is the application of Graph Convulutional Networks for encoding knowledge graphs. While (graph-)structured knowledge represents a significant challenge for classical machine learning models, graph convolutional networks (Kipf and Welling 2017) offer an effective framework for representation learning of graphs. In parallel, relational GCNs (R-GCNs) (Schlichtkrull et al. 2018) have been designed to accommodate the highly multi-relational data characteristics of large KGs. Inspired by these works, we explore the use of R-GCNs for infusing information from KG into NLI models.

3 KG-Augmented Entailment Model

In this section, we describe the central contribution of the paper – the KG-augmented Entailment System (KES). As shown in Figure 2, KES consists of two main components. The first component is a standard text-based model that creates a fixed-size representation of the premise and hypothesis texts. The second component selects contextual subgraphs for the premise and the hypothesis from a given KG, and encodes them using a GCN. The fixed size representations from the two components are used as input to a standard feedforward layer for classification. We opted for a combined graph and text approach because the noise and incompleteness of KGs renders a purely graph-based approach insufficient as a standalone solution. However, we show that the KG-augmented model provides valuable context and additional knowledge that may be missing in text-only representations.

3.1 A Standard Text-based Model

Given the premise $P = (p_1, \ldots, p_n)$ and hypothesis $H = (h_1, \ldots, h_m)$, let p_i and h_j be the embeddings of words occurring in sequence in the premise and hypothesis texts. These embeddings are input to a neural network T_{NLI} that outputs a fixed size representation $t_{out} \in \mathbb{R}^K$:

$$t_{out} = T_{NLI}(P, H) \tag{1}$$

where T_{NLI} can be any of the existing state of the art text-based NLI models (Wang and Jiang 2016a; Talman, Yli-Jyrä, and Tiedemann 2019; Liu et al. 2019a).

3.2 Contextual Subgraphs and their Representation using GCNs

This component uses an external KG to obtain a subgraph that is relevant with respect to the premise and the hypothesis, and then applies GCN to encode this subgraph into a fixed-size representation g_{out} (Figure 3).

Subgraph Extraction: In order to retrieve a subgraph from the KG, we first map the terms in premise and hypothesis text to concepts in KG by performing a max-substring match. For example, given the premise and hypothesis in Figure 1, the extracted and mapped concepts are shown in blue and green. Next, this initial set of concepts is then expanded to include (one-hop) neighbor concepts, and all the edges between them (initial set and their neighbors) from the KG. In the example in Figure 1, we extract a subgraph that includes the purple nodes because they are directly connected to green and/or blue nodes.

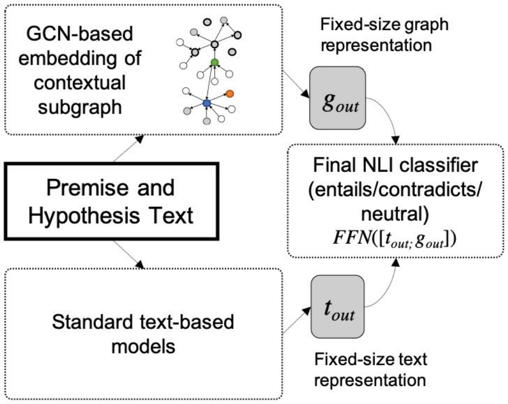
GCN-based embedding of contextual subgraph
Fixed-size graph representation
g_{out}
Premise and Hypothesis Text
Final NLI classifier (entails/contradicts/ neutral)
$FFN([t_{out;}\, g_{out}])$
Standard text-based models
t_{out}
Fixed-size text representation

Figure 2: Primary components of KES: standard text-based model, GCN-based graph embedder, and final feedforward classifier.

Personalized PageRank (PPR) to Filter Context: KGs are typically very large, and concept expansion by just one hop can introduce a significant amount of noise (Wang et al. 2019; Lalithsena et al. 2017). For example, the concept `girl` is directly connected to over 1000 other concepts in ConceptNet. For this reason, we create a *contextual* subgraph by further filtering the one-hop subgraph.

To obtain the most relevant neighbor nodes given the premise and hypothesis texts, we use Personalized PageRank (PPR) (Page et al. 1999). PPR adds a bias to the PageRank algorithm by scoring the nodes conditioned on a initial subset of nodes in the graph. The bias is introduced by changing the uniformly distributed jump probability vector **p** of PageRank to a non-uniform distribution with respect to the initial subset of nodes (Equation 2). In our settings, this initial subset of nodes S consists of the concepts mentioned in the premise and hypothesis.

$$p_i = \begin{cases} \dfrac{1}{|S|} & i \in S \\ 0 & i \notin S \end{cases} \tag{2}$$

PPR-scores $\mathbf{R}'$ are then computed as follows:

$$\mathbf{R}' = (1 - \alpha)\mathbf{A} \times \mathbf{R} + \alpha \mathbf{p} \tag{3}$$

where $\mathbf{R}$ is a vector with scores for each node (post convergence); $\mathbf{A}$ is a normalized adjacency matrix (transition probability matrix); and α is the damping factor.

We normalize the PPR-scores based on the maximum PPR-score of a node in the sub-graph. We then choose a filtering threshold θ, and exclude all the nodes that are not in the initial subset S and that have a PPR-score below θ; we also exclude the edges that link to the deleted nodes. The remaining nodes and edges make up the contextual subgraph for the premise-hypothesis pair under consideration.

Encoding Contextual Subgraphs: The contextual subgraph for premise and hypothesis is encoded using a relational graph convolutional network (R-GCN) (Schlichtkrull et al. 2018). GCNs compute node embeddings by iteratively aggregating the embeddings of neighbor nodes. R-GCNs extend standard GCNs (Kipf and Welling 2017) to deal with the multi-relational data of KGs. They learn different weight matrices for each type of relation occurring in the graph. We use an R-GCN to compute node embeddings, and then aggregate these embeddings to obtain a fixed-size representation for the contextual subgraph.

We first extend the contextual subgraph by adding a self-loop edge for each node; this is to retain the information of the node during convolution. Previous work (Wang et al. 2019) showed that the concepts mentioned in premise and hypothesis played an important role to improve NLI performance. Inspired by this, we retain information of concepts (nodes) that occur in premise and hypothesis text by adding a premise supernode v_p and hypothesis supernode v_h. The premise supernode is connected to concepts that are mentioned in premise using bi-directional edges and similarly the hypothesis supernode is connected to the concepts mentioned in the hypothesis.

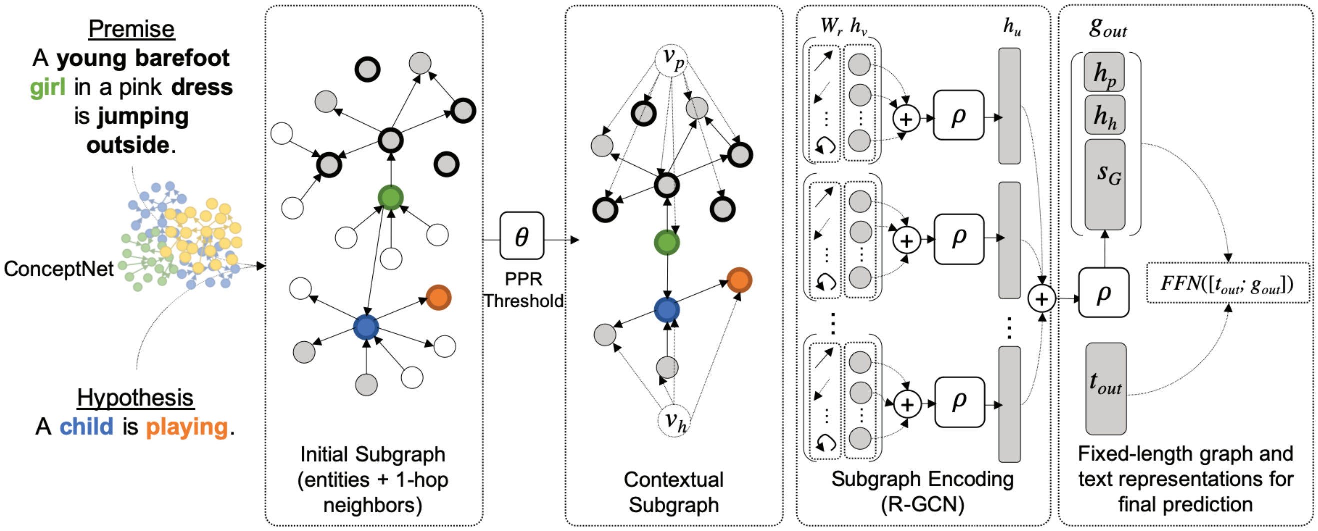
Premise
A young barefoot girl in a pink dress is jumping outside.
ConceptNet
Hypothesis
A child is playing.
Initial Subgraph (entities + 1-hop neighbors)
θ
PPR Threshold
v_p
v_h
Contextual Subgraph
W_r h_v
h_u
ρ
Subgraph Encoding (R-GCN)
g_{out}
h_p
h_h
s_G
$FFN([t_{out}; g_{out}])$
t_{out}
Fixed-length graph and text representations for final prediction

Figure 3: Overview of the KES approach. KES links terms in the premise and hypothesis to concepts in KG, creates contextual subgraphs via personalized page rank filtering, encodes those subgraphs with an R-GCN, and finally combines the aggregated node embeddings with text representations into a feedforward classifier. h_p and h_h in the figure denote $h^L_{v_p}$ and $h^L_{v_h}$ in Equation (6) respectively.

We then apply the algorithm suggested by Nguyen and Grishman (2018) – which uses a simple sum as the aggregation function – but we include a normalization factor and disregard bias (similar to Schlichtkrull et al. (2018)):

$$h_u^{l+1} = \rho\left(\sum_{r\in\mathcal{R}}\sum_{v\in\mathcal{N}_{u,r}}\frac{1}{c_{u,r}}W_r^l h_v^l\right). \qquad (4)$$

Here, $\mathcal{R}$ is the set of edge types; $\mathcal{N}_{u,r}$ is the set of neighbors connected to node u through the edge type r; $c_{u,r}$ is a normalization constant; W_r^l are the learnable weight matrices, one per edge type $r \in \mathcal{R}$; and ρ is a non-linear activation function. We use the (symmetric) normalized Laplacian as a normalization constant (Kipf and Welling 2017).

The final node embeddings are aggregated using a summation-based graph-level readout function (Xu et al. 2019):

$$s_G = \rho\left(\sum_{v\in V} W h_v^L\right). \qquad (5)$$

V is the set of nodes in our contextual graph, W is a learnable weight matrix, and ρ is an activation function. This summation-based readout function allows the encoder to learn representations that encode the structure of the graph.

The final representation of the contextual subgraph is obtained by concatenating s_G – the aggregated embeddings of all the nodes – with the embeddings of the premise and hypothesis supernodes as follows:

$$g_{out} = [s_G; h^L_{v_p}; h^L_{v_h}] \qquad (6)$$

3.3 Final Classifier

The final feedforward classifier takes as input the text encoding from Equation (1) and the graph encoding from Equation (6) to classify the premise and hypothesis as entailment/contradiction/neutral:

$$\mathbf{E_{pred}} = FFN([t_{out}; g_{out}]) \qquad (7)$$

4 Experiments & Results

In this section, we describe the experiments that we performed to evaluate our approach; the setup, including datasets, models, and implementations; and the results.

4.1 Datasets

We considered the most popular NLI datasets: SNLI (Bowman et al. 2015), SciTail (Khot, Sabharwal, and Clark 2018), and MultiNLI (Williams, Nangia, and Bowman 2018). While SNLI and MultiNLI are prominent datasets covering a wide range of topics, SciTail offers an in-depth focus on science domain questions. Since this difference is also reflected in linguistic variations, the two datasets allow evaluating very different settings. As mentioned in Section 2, these datasets carry linguistic cues that are easily captured by the neural text based models. Hence, to show the impact of knowledge graphs, we primarily evaluate our approach on the and BreakingNLI dataset (Glockner, Shwartz, and Goldberg 2018).

4.2 Knowledge Graphs

Prior work on NLI has shown that ConceptNet contains information more useful to this problem compared to DBpedia and WordNet (Wang et al. 2019). Furthermore, Speer, Chin, and Havasi (2017) showed that, when ConceptNet is combined with embeddings acquired from distributional semantics, it provides applications with a richer understanding than narrower resources like the latter KGs. We therefore focus on ConceptNet for now, leaving experiments with other KGs as future work.

4.3 Models for Text Representations

We experimented with four different text-based models to obtain numerical representations of premise and hypothesis text (Equation (1)). Our selection criteria: (1) performance on leaderboards, (2) relevance for NLP in general, and (3) ease of implementation and availability. Our goal is to augment each of these models with external knowledge and hence test the generalizability of KES, which also shows the benefits of its modularity. We used the AllenNLP library[1] to implement the models described below (see also Section 2).

Decomposable Attention Model (DecompAttn). One of the earlier and most common baseline models used for NLI (Parikh et al. 2016; Wang et al. 2019; Glockner, Shwartz, and Goldberg 2018; Chen et al. 2018). Hence, our hypothesis is that KES can add more value and have a larger delta in performance.

match-LSTM. A NLI model with good performance on not only on multiple NLI leaderboards such as SciTail and SNLI but also applicable to other NLP tasks such as question answering (Wang and Jiang 2016b).

BERT + match-LSTM. Version of match-LSTM using BERT embeddings instead of the GLoVe embeddings in the former. We opted for this model to take advantage of the improvements BERT embeddings have generated for numerous NLP tasks.

Hierarchical BiLSTM Max Pooling (HBMP). Shows superior performance on multiple NLI benchmarks including SciTail, SNLI, and MultiNLI.

4.4 Models using External Knowledge

There are two other models exploiting external knowledge for NLI. We compare them to KES:

KIM (Chen et al. 2018) uses five different features for every pair of terms from premise and hypothesis. The features are extracted from WordNet and they are infused in the model as knowledge-based co-attention mechanism.

ConSeqNet (Wang et al. 2019) takes the concepts mentioned in premise and hypothesis as input to a match-LSTM model (with a GRU encoding). It is important to note that the match-LSTM model better suits text than graph structure because it uses a seq2seq encoder to account for the inherent sequential nature of text, which is not present in graphs.

4.5 Experimental Setup and Implementation

To evaluate the impact of KES on NLI in general and its compatibility with various existing models, we compared all text-based models described above (Section 4.3) to a combined text+graph model. Because the BreakingNLI test set is derived from the SNLI training set, all models trained on SNLI were evaluated on both the SNLI and BreakingNLI test sets.

Text Model Parameters. We chose hyperparameters as reported in related works. For match-LSTM and BERT-match-LSTM, we refer to (Wang et al. 2019). For HBMP and DecomAttn, we used the parameters from (Talman, Yli-Jyrä, and Tiedemann 2019) and (Parikh et al. 2016) respectively.

KES Setup and Training. As initial graph embeddings, we considered TransH (Wang et al. 2014) and ComplEx (Trouillon et al. 2016). For each model (i.e., text-only + graph model combination), we experimented with both embedding approaches and selected the one that performed best on the validation sets. All GCNs were configured as follows: two edge types (one for edges in ConceptNet and one for the self-loops); 300 dimensions for each embedding across all layers; one convolutional layer; one additional linear layer (after the convolution); and ReLU for all activations. These parameters yielded best average accuracy on the validation sets, so that we chose them uniformly for all models for consistency across our approaches.

The Personalized PageRank threshold θ for filtering the subgraphs was also tuned as a hyperparameter. We experimented with θ values of 0.2, 0.4, 0.6, and 0.8. We did not experiment with whole one-hop graphs (θ = 0.0), as they have been shown to increase in size very rapidly over single hops in ConceptNet (Wang et al. 2019).

Training (of the combined models) consisted of 140 epochs with a patience of 20 epochs. Batch size and learning rate over all the experiments remained 64 and 0.0001 to make the models comparable to each other.

4.6 Results

Table 1 gives an overview of our results. They demonstrate that KES, and thus external knowledge, has the biggest impact on the BreakingNLI test set. The accuracy of text-only models is improved, for BERT-based match-LSTM model by 18 percentage points, match-LSTM by 13 percentage points, HBMP by 3 percentage points, and DecompAttn by 8 percentage points. Notably, the most dramatic impact of KES is on the BERT-based match-LSTM model, which is generally the strongest text-only model on the other datasets.

Despite their competitive performance on SNLI, all text-only models perform significantly worse on the BreakingNLI test set when compared to the SNLI test set, which is consistent with observations from the original BreakingNLI paper. The DecompAttn text-only model shows the biggest drop in performance (28 percentage points) between SNLI and BreakingNLI. The match-LSTM text-only model shows the smallest drop in performance between SNLI and Breaking NLI – still a substantial 18 percentage points. In contrast, KES shows only modest decreases in performance between SNLI and BreakingNLI when a GloVe- or BERT-based match-LSTM text model is used, with accuracy decreasing only 5 and 8 percentage points respectively. However, there is a significant decrease in performance between SNLI and BreakingNLI when KES uses HBMP or DecompAttn as its text model (20 and 26 percentage points re-

[1] https://github.com/allenai/allennlp. AllenNLP includes the Decompattn model.

Models	**Scitail**		**MultiNLI**		**SNLI**		**BreakingNLI**	
	Text	**KES**	**Text**	**KES**	**Text**	**KES**	**Text**	**KES**
match-LSTM	82.54	82.22 (0.6)	71.32	71.67 (0.8)	83.60	83.94 (0.6)	65.11	**78.72**
BERT+match-LSTM	89.13	**90.68** (0.2)	77.96	76.73 (0.6)	85.78	85.97 (0.6)	59.42	**77.59**
HBMP	81.37	**83.49** (0.2)	69.27	68.42 (0.6)	84.61	83.84 (0.2)	60.31	**63.60**
DecompAttn	76.57	72.43 (0.8)	64.89	**71.93** (0.6)	79.28	**85.56** (0.6)	51.3*	**59.83**
Existing Models with KG	**Text**	**Text+Graph**	**Text**	**Text+Graph**	**Text**	**Text+Graph**	**Text**	**Text+Graph**
KIM (Chen et al. 2018)	-	**NE**	-	76.4*	-	88.6*	-	83.1*
ConSeqNet (Wang et al. 2019)	84.2*	85.2*	71.32	70.9	83.60	83.34	65.11	61.12

Table 1: Entailment accuracy results of KES with different text models compared to text-only entailment models (Text). Bold values indicate where KES improves performance. PPR θ-values are shown in parentheses.*Reported values from related work.

spectively), suggesting a potentially complex interaction between text and external knowledge features. Overall, while KES models perform comparably to its text-based counterparts for SNLI, SciTail, and MultiNLI, they perform significantly better on BreakingNLI dataset.

These results support three important claims. First, they demonstrate that KES is modular in that it can be combined with existing text models with different architectures. Second, the KES approach effectively infuses external knowledge into existing entailment models to improve performance on the challenging BreakingNLI dataset. Third, KES is robust to dataset changes that dramatically decrease the performance of other NLI models.

Comparison to Other KG-based Models. Table 1 also contains the results for the graph-based models KIM and ConSeqNet. Both show comparable performance to match-LSTM KES, with KIM performing best on all datasets. We discuss important differences between KES, KIM, and ConSeqNet below.

KIM introduces an external knowledge-based co-attention mechanism, using five manually engineered features from WordNet for every term pair of words in premise and hypothesis. These features are specific to WordNet relations, which means that the model can only be used with WordNet or comparable KGs with the same set of relations. One can argue that, because KIM depends on WordNet, it is especially suited to BreakingNLI, as WordNet contains exactly the type of lexical information that is targeted by BreakingNLI. Another difference between KIM and KES is that it is not clear how to adapt KIM's five engineered features to a different textual entailment system. In contrast, KES is not tied to any particular KG, KG vocabulary, or existing entailment system. One of the practical goals of KES is to develop an approach that is easily adaptable to different datasets, knowledge graphs, and existing entailment models. Although tuning only the PPR threshold as the hyper parameter, our knowledge augmented approaches perform almost on par with KIM on on SNLI and MultiNLI except BreakingNLI dataset (-4.4 percentage).

ConSeqNet, similar to our model and unlike KIM, does provide an architecture to plug in any text based entailment model. However, there are two primary differences between our work and ConSeqNet. First, we are the first to encode the graph structure of the knowledge graph where as ConSeqNet uses on the concepts mentioned in text encoding them using RNNs. Also, in comparison to ConSeqNet, our approach performs better with different entailment models over all the datasets. Particularly, on the BreakingNLI dataset, our implementation of ConSeqNet shows a drop in performance in comparison to its text-based method. This is in turn surprising and may need further investigations.

In summary, in addition to the performance goals of KIM and ConSeqNet, KES seeks to infuse entailment models with knowledge in a way that is modular and sensitive to graph structure, independent of a specific KG.

Harnessing External Knowledge. Table 2 shows the average number of concepts (nodes) and relations (edges) in contextual subgraphs generated by KES, ConSeqNet, and KIM, excluding those that were explicitly mentioned in the premise and hypothesis texts. Unlike ConSeqNet and KIM, KES is able to use a great amount of external knowledge that is related to the premise and hypothesis but not explicitly mentioned. As observed in prior work (Wang et al. 2019), expanding subgraphs by even one hop results in very large graphs, making PPR filtering very important.

5 Discussion

Negative Results: In Table 1, we observe two results that did not confirm our hypotheses: (1) the reduced text+graph improvement on BreakingNLI for HBMP, and (2) lower text+graph performance for DecompAttn on SciTail (>2 percentage points). We are investigating these issues, but one possible explanation for the reduced improvement on HBMP is that it is one of the few text based models that has a large final hidden layer (14K feature vector) in comparison to the features from the GCN model (900) which is possibly biasing the final classifier towards the text-based features.

Personalized PageRank Threshold: Our initial plan for using PPR thresholds was to make it a preprocessing step and fix one threshold for a dataset on a base model. However, as shown in Table 1, using PPR thresholding as a hyperparameter for each model trained showed better performance. Also, the PPR threshold, in particular 0.8 filters very few concepts that aren't mentioned in premise and hypothesis text, whereas contextual subgraphs from 0.2 can contain the equal number of concepts from external knowledge as mentioned in text (Table 2). PPR filtering is just one possible method for reducing noise that results from neighborhood-based expansion techniques. In our future work, we intend

PPR	Scitail (17.74*)		SNLI (11.5*)		MultiNLI (17.5*)	
	Edges	**Nodes**	**Edges**	**Nodes**	**Edges**	**Nodes**
0.2	42.65	10.14	80.29	19.83	76.27	16.15
0.4	26.72	7.48	25.70	8.15	33.82	6.48
0.6	15.53	4.35	14.08	4.65	23.97	3.44
0.8	11.67	3.04	9.98	3.18	20.27	2.05
ConSeqNet	0	0 (17.74*)	0	0 (11.5*)	0	0 (17.5*)
	No edges or new concepts are added from ConceptNet.					
KIM	Features based on fixed WordNet relations. No new concepts are added.					

Table 2: Average number of nodes and edges (not explicitly mentioned in text) in combined premise and hypothesis subgraphs by PPR threshold. *Average number of concepts explicitly mentioned in each premise and hypothesis text.

to investigate a different filtering approach where only those paths that connect premise and hypothesis are included.

Dataset characteristics. We evaluated our KES approach on NLI datasets that are widely used in the literature. However, there has been criticism regarding the way these datasets are created and the resulting biases that can be exploited by learning algorithms (Glockner, Shwartz, and Goldberg 2018; Li et al. 2019; Gururangan et al. 2018; Poliak et al. 2018). Even in our work, in Table 1 where we see that the DecompAttn model is consistently improved by KES on SNLI, MNLI, and BreakingNLI, we also see the opposite effect on SciTail. Some qualitative analysis of the SciTail dataset showed us that use of KG can negatively impact the performance because of high overlap between premise and hypothesis terms.

Text-based models trained on SNLI perform significantly worse on the BreaklingNLI test set, consistent with the results reported above. Notably, the estimated human performance on the BreakingNLI test set is higher than that of the original SNLI test set, providing further evidence that models that perform well on SNLI but poorly on BreakingNLI are poor approximations for human inference. On the other hand, NLI models that generalize well to BreakingNLI are more likely to be better approximations for human-like inference. The complexity of the BreakingNLI test set and its characteristics make it the most interesting evaluation set.

Complexity of Knowledge Graphs and their usage: As mentioned above, the current state-of-the-art for BreakingNLI is the KIM model, which achieves an 83% accuracy, while our best performing KES model (KES with the match-LSTM text model) achieves an accuracy of 79%. This difference can be attributed to aspects of the KIM model that make it particularly well suited to the BreakingNLI dataset at the expense of model flexibility and generality. KIM relies on WordNet, which has lexical information that aligns very closely with the challenging aspects of the BreakingNLI. This focus clearly benefits performance on the task. However, WordNet is relatively small (117k triples, i.e., edges) compared to ConceptNet (3.15M triples) and has a very specific scope that is unlikely to cover the broad classes of entailment that occur in natural language. For example, recognizing textual entailment may depend on world knowledge that is not lexical in nature. In such cases it would be necessary to invoke a model that is not primarily focused on lexical knowledge. This is one of the motivations behind the KES approach: to support very large KGs (e.g., ConceptNet) and to avoid dependencies on any single KG or domain area. An important topic for future work will be to understand the shortcomings of various knowledge sources, how to manage choosing the appropriate knowledge sources for a given task, and to continue exploring graph filtering and selection methods to leverage large scale KGs while minimizing noise. KIM mitigates the noise issue by using a restricted set of relations to provide greater focus and minimize intrusion of potentially irrelevant knowledge. Again, this is a characteristic of KIM that will not necessarily generalize well to other NLI datasets, such as SciTail, which may depend less on hyper- and hyponym relations, and more on knowledge about everyday physical objects and processes.

6 Conclusion

In this paper, we presented a systematic approach for infusing external knowledge into the textual entailment task using contextually relevant subgraphs extracted from a KG and encoded with graph convolutional networks. These graph representations are combined with standard text-based representations into a KG-augmented entailment system which yields significant improvement on the challenging BreakingNLI dataset. Additionally, the KES approach is modular, can be used with any knowledge graph, and is generalizable to multiple datasets. In our future work, we plan to consider other KGs and to investigate alternative graph representations. Furthermore, it would be interesting to see how KES performs on the popular question answering datasets.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer. 722–735.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of Empirical Methods in Natural Language Processing Conference*, 632–642.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Inkpen, D.; and Wei, S. 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)*, 2406–2417.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Glockner, M.; Shwartz, V.; and Goldberg, Y. 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)*, 650–655.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference: Human Language Technologies, Volume 2 (Short Papers)*, 107–112.

Harabagiu, S., and Hickl, A. 2006. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 905–912. Association for Computational Linguistics.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 105–113. ACM.

Jeh, G., and Widom, J. 2003. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, 271–279. ACM.

Khot, T.; Sabharwal, A.; and Clark, P. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Lalithsena, S.; Perera, S.; Kapanipathi, P.; and Sheth, A. 2017. Domain-specific hierarchical subgraph extraction: A recommendation use case. In *International Conference on Big Data (Big Data)*, 666–675. IEEE.

Li, T.; Zhu, X.; Liu, Q.; Chen, Q.; Chen, Z.; and Wei, S. 2019. Several experiments on investigating pretraining and knowledge-enhanced models for natural language inference. *arXiv preprint arXiv:1904.12104*.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4487–4496. Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

MacCartney, B., and Manning, C. D. 2009. *Natural language inference*. Stanford University Stanford.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Moon, S.; Shah, P.; Kumar, A.; and Subba, R. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 845–854.

Musa, R.; Wang, X.; Fokoue, A.; Mattei, N.; Chang, M.; Kapanipathi, P.; Makni, B.; Talamadupula, K.; and Witbrock, M. 2019. Answering science exam questions using query rewriting with background knowledge. *Automated Knowledge Base Construction*.

Nguyen, T. H., and Grishman, R. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Parikh, A.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A decomposable attention model for natural language inference. In *Proceedings of Empirical Methods in Natural Language Processing Conference*, 2249–2255.

Poliak, A.; Naradowsky, J.; Haldar, A.; Rudinger, R.; and Durme, B. V. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference: Human Language Technologies*, 180–191. Association for Computational Linguistics.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the European Semantic Web Conference*, 593–607. Springer.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Talman, A.; Yli-Jyrä, A.; and Tiedemann, J. 2019. Sentence embeddings in nli with iterative refinement encoders. *Natural Language Engineering* 25(4):467–482.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *Proceedings of the International Conference on Machine Learning*, 2071–2080.

Wang, S., and Jiang, J. 2016a. Learning natural language inference with LSTM. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference: Human Language Technologies*, 1442–1451.

Wang, S., and Jiang, J. 2016b. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, X.; Kapanipathi, P.; Musa, R.; Yu, M.; Talamadupula, K.; Abdelaziz, I.; Chang, M.; Fokoue, A.; Makni, B.; Mattei, N.; and Witbrock, M. 2019. Improving Natural Language Inference Using External Knowledge in the Science Questions Domain. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference: Human Language Technologies, Volume 1*, 1112–1122.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How powerful are graph neural networks? In *Proceedings of the International Conference on Machine Learning*.

Yang, X.; Zhu, X.; Zhao, H.; Zhang, Q.; and Feng, Y. 2019. Enhancing unsupervised pretraining with external knowledge for natural language inference. In *Proceedings of the Canadian Conference on Artificial Intelligence*, 413–419.

Zhang, Z.; Wu, Y.; Li, Z.; He, S.; Zhao, H.; Zhou, X.; and Zhou, X. 2018. I know what you want: Semantic learning for text comprehension. *arXiv preprint arXiv:1809.02794*.