The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

SCITAIL: A Textual Entailment Dataset from Science Question Answering

Tushar Khot, Ashish Sabharwal, Peter Clark
Allen Institute for Artificial Intelligence, Seattle, WA, U.S.A.
{tushark,ashishs,peterc}@allenai.org

Abstract

We present a new dataset and model for textual entailment, derived from treating multiple-choice question-answering as an entailment problem. SCITAIL is the first entailment set that is created solely from natural sentences that already exist independently "in the wild" rather than sentences authored specifically for the entailment task. Different from existing entailment datasets, we create hypotheses from science questions and the corresponding answer candidates, and premises from relevant web sentences retrieved from a large corpus. These sentences are often linguistically challenging. This, combined with the high lexical similarity of premise and hypothesis for both entailed and non-entailed pairs, makes this new entailment task particularly difficult. The resulting challenge is evidenced by state-of-the-art textual entailment systems achieving mediocre performance on SCITAIL, especially in comparison to a simple majority class baseline. As a step forward, we demonstrate that one can improve accuracy on SCITAIL by 5% using a new neural model that exploits linguistic structure.

Introduction

Recognizing textual entailment (RTE) involves assessing whether a given textual *premise* entails or implies a given *hypothesis*. It is a central problem in natural language understanding (Dagan et al. 2013) as it encapsulates the fundamental challenge of linguistic variability. The richness and subtlety of natural language, however, makes RTE highly challenging. To facilitate the development of strong RTE systems, increasingly larger datasets have been proposed, ranging in size from 100s to over 500,000 annotated premise-hypothesis pairs. Datasets such as RTE-n (Dagan, Glickman, and Magnini 2005), SICK (Marelli et al. 2014), and SNLI (Bowman et al. 2015) have played an important role in advancing the field.

A limitation of mainstream entailment datasets, however, is that they have been constructed in isolation from any end task.[1] Moreover, in several cases, either the hypothesis or the premise has been synthesized (e.g., rule-based in SICK and crowd-sourced in SNLI) specifically for creating the entailment dataset. Consequently, while helpful in advancing RTE research, these datasets do not always capture the kind of entailment queries that naturally arise in an end task. We present the largest entailment dataset that is directly derived from an end task and consists of naturally occurring text as both premise and hypothesis.

Question:
Which of the following best explains how stems transport water to other parts of the plant?
(A) through a chemical called chlorophyll.
(B) by using photosynthesis.
(C) through a system of tubes. ✓
(D) by converting water to food.

Assertion from question + answer candidate (C):
Stems transport water to other parts of the plant through a system of tubes.

Supporting sentence (*entails*):
Water and other materials necessary for biological activity in trees are transported throughout the stem and branches in thin, hollow tubes in the xylem, or wood tissue.

Non-supporting sentence (*neutral*):
Cut plant stems and insert stem into tubing while stem is submerged in a pan of water.

Figure 1: Example annotations for SCITAIL

Our new dataset, SCITAIL, is designed from the end task of answering multiple-choice school-level science questions. Each question and the correct answer for it are converted into an assertive statement to form a hypothesis H; see Figure 1 for an example. We use an information retrieval (IR) method to obtain relevant text from a large text corpus of web sentences, and use each of these sentences as a premise P. While each P, by construction, has a high lexical overlap with H, not every P entails or "supports" the statement in H. We crowdsource the annotation of each such premise-hypothesis pair as supports or not, in order to create the SCITAIL entailment dataset with 27K examples.

The entailment dataset focuses on the reasoning needed for QA by factoring out the retrieval step. By the nature of its construction, this dataset captures what a good textual QA

[1] A few small-scale datasets have been derived from an end task, such as RTE-6 for news summaries with fewer than 1000 examples.

system needs to be able to infer. A substantial performance improvement on this dataset is thus expected to translate into better QA performance as well.

Since both the premise and the hypothesis in SCITAIL were authored independently of each other and independent of the entailment task, linguistic variations in the dataset are not limited by the coverage of hand-designed rules or the creativityof crowd-workers, especially when shown one of the two pieces of text. Further, unfiltered web sentences, which are used to create the premise, tend to be highly diverse in various aspects (length, complexity, being well-formed for a parser, etc.), adding to the linguistic challenge.

We find that current RTE systems, including neural entailment models, have mediocre performance on this dataset, whether pre-trained on their own datasets or on SCITAIL. For instance, the state-of-the-art Decomposable Attention Model (Parikh et al. 2016) achieves an accuracy of 72.3%, which is only 2% higher than a simple n-gram overlap model and 12% higher than the majority class prediction baseline of 60.3%. In contrast, on the highly studied SNLI dataset, the 75% accuracy of even the basic entailment models is much higher than the 33.3% majority baseline.

We demonstrate that one can exploit linguistic structure to better capture the entailment relation in this dataset. Specifically, our asymmetric Decomposed Graph Entailment Model (DGEM) raises the accuracy to 77.3%.

In summary, we make the following contributions:

1. A natural entailment dataset[2] where the text and hypothesis were authored independent of each other and independent of the entailment task;
2. the first entailment dataset derived from the end task of multiple-choice question answering; and
3. a new model that exploits linguistic structure in the hypothesis to outperform existing techniques on this dataset.

Related Work

We discuss prior work on textual entailment and question answering that is most closely related to SCITAIL.

Textual Entailment

The PASCAL RTE challenges (Dagan, Glickman, and Magnini 2005) have played an important role in developing our understanding of the linguistic entailment problem. Due to the small size of these datasets, most earlier approaches relied on hand-designed features and alignment systems (Androutsopoulos and Malakasiotis 2010). With the advent of large entailment datasets (Bowman et al. 2015), novel neural network architectures have been developed for the entailment task. However, these datasets were designed in isolation from any end task and with synthesized sentences. As a result, while they help advance our understanding of entailment, they do not necessarily capture entailment queries that naturally arise in an end task.

With regard to using linguistic structure, deep learning entailment models mainly rely on generating a single vector representation for each of the premise and the hypothesis, using attention between the sentences (Chen et al. 2017; Parikh et al. 2016). Few models have incorporated syntactic structure from both premise and hypothesis, to help improve these representations. Our proposed model explicitly uses the syntactic structure, viewed as a graph, by identifying the entailment probability of individual nodes and edges in the hypothesis structure. This idea is similar to the approach of Zhao, Huang, and Ma (2016), who compute entailment probabilities on each node in a binarized tree from the premise and hypothesis. Our approach differs in that it does not rely on a binarized tree representation and uses structure only from the hypothesis. The hypotheses typically are short and thus result in a more reliable extracted structure.

Question Answering

Science QA task involves the challenging domain of school-level science exams, where questions often require complex reasoning to arrive at the correct answer (Clark et al. 2016). Consequently, most systems attempt to stitch together multiple rules (Khot et al. 2015), table rows (Khashabi et al. 2016), or Open IE tuples (Khot, Sabharwal, and Clark 2017) to produce an answer. However, these systems must both retrieve the relevant knowledge and perform the required reasoning, without really knowing whether the retrieved knowledge actually supports the answer or whether there even exists any knowledge in the underlying knowledge base that can, in principle, be used to answer the question. Our dataset identifies sentences that are annotated as supporting the correct answer for each question. It thus opens up the way for QA systems to factor out the retrieval aspect and focus on the reasoning challenge.

Reading comprehension (RC) datasets (Rajpurkar et al. 2016; Richardson, Burges, and Renshaw 2013; Joshi et al. 2017) are similar in that they allow systems to focus on the reasoning aspect of question answering. However, these datasets require the system to identify the answer span in the paragraph, which is a harder task than predicting textual entailment. At the same time, answer choices in Science QA need not be valid spans in the retrieved sentence(s), thus making the task out of scope for span prediction models.

Question Answering as Entailment

Consider the following question from 4th grade science test:

Which of the following best explains how stems transport water to other parts of the plant?
(A) through a chemical called chlorophyll.
(B) by using photosynthesis.
(C) through a system of tubes.
(D) by converting water to food.

Upon reading the statement, "*Water and other materials necessary for biological activity in trees are transported throughout the stem and branches in thin, hollow tubes in the xylem, or wood tissue.*", a layman can conclude that the answer '*through a system of tubes*' is correct. In other words, the question combined with answer choice (C) is *entailed* by this knowledge statement. We use this intuition to create an entailment dataset where each premise consists of a knowledge sentence and each hypothesis is a

[2] Available at http://data.allenai.org/scitail

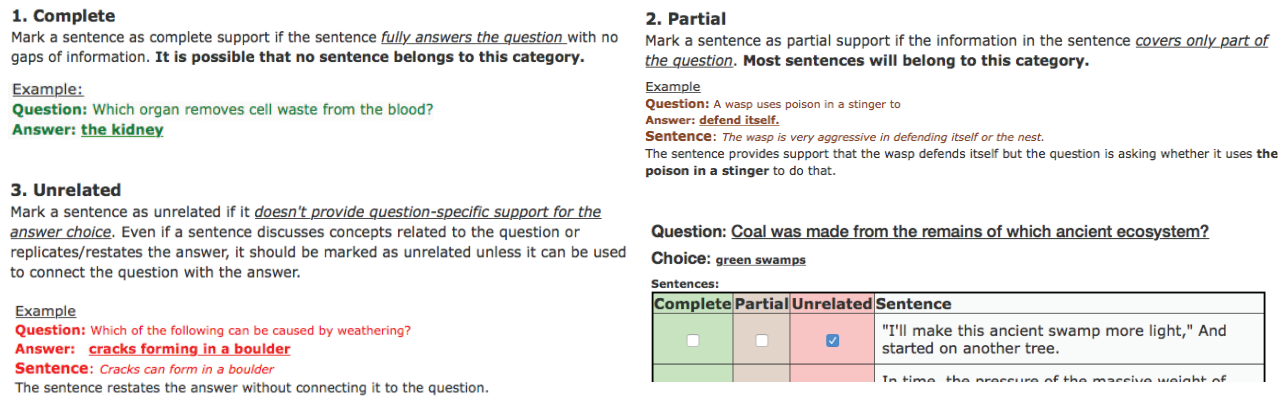

Figure 2: Snippets from the annotation tool. The original annotation page had more examples and a description of the task.

statement representing the question along with an answer candidate (e.g., "*Stems transport water to other parts of the plant through a system of tubes*".). Given an entailment model trained on such a dataset, multiple-choice questions can now be answered by returning the answer choice with the highest entailment score for a knowledge sentence as a premise and a hypothesis generated from the answer choice. We next describe a general methodology for annotating such an entailment dataset starting from a multiple-choice question set, and then discuss specifics of the SCITAIL dataset.

Entailment Annotation Task

To create an entailment dataset from a QA task, we start with a dataset $\mathcal{Q}$ of multiple-choice questions, and an indexed corpus $\mathcal{T}$ of sentences. For each multiple-choice question $q \in \mathcal{Q}$, we collect a set of candidate knowledge sentences P to generate the premises of the entailment dataset. Rather than synthesizing these sentences, we use for P the top K retrieved sentences from $\mathcal{T}$. Given a candidate knowledge sentence (premise) $p \in P$ for the question q and an answer choice a, we next collect an entailment label for the knowledge sentence and (q, a) pair. We convert the (q, a) pair into an assertion, $h = assertion(q, a)$ to create the entailment example (p, h) with the annotated label.

For an incorrect answer choice, one can assume that none of the knowledge statements will support this answer.[3] However, we can not make an analogous assumption about the sentences for the correct answers, i.e., not all retrieved sentences support the correct answer choice. Specifically, among the 43% of questions that were answerable by a single sentence, only 16.6% of the retrieved sentences provided sufficient support to answer the question. Hence we focus on obtaining annotations for the correct answer choices.

For each question and correct answer choice, we show a batch of 10 sentences to crowd-workers and ask them to classify each sentence into one of three categories:

1. *Complete Support*, if the sentence fully supports the answer choice;
2. *Partial Support*, if the sentence is related to the question but only provides partial support for the answer; or
3. *Unrelated*, if the sentence is unrelated to the question.

Figure 2 shows a snippet of the annotation guidelines and the task. We used the *Complete Support* label to create the examples with *entails* label and the *Unrelated* label to create *neutral* label examples. *Partial Support* examples were ignored but can potentially be useful to identify and retrieve the knowledge gaps in these sentences. We had also included a 'contradicts' label but, in our pilot annotations, we noticed that we rarely had examples of contradiction and so we simplified the task by dropping the additional label. Since entailment annotations can be ambiguous depending on the sentence being labeled, each sentence was annotated by 5 crowd-workers and only sentences with 80% agreement were retained (similar to RTE).

SCITAIL Dataset

We used this annotation scheme to create an entailment dataset for the Science question answering task. We use multiple-choice science questions from publicly released 4th grade (204 questions) and 8th grade (195 questions) exams[4] and the crowd-sourced questions from SciQ dataset (2,835 questions) (Welbl, Liu, and Gardner 2017) to create $\mathcal{Q}$. For the text corpus, $\mathcal{T}$, we use a large text corpus from Clark et al. (2016) containing 280 GB of plain text extracted from Web pages and 80,000 sentences from domain-targeted sources. We retrieved the top K=40 sentences from this corpus using the tokens from the question and answer choice as the query as described by Khot, Sabharwal, and Clark (2017). We used Amazon Mechanical Turk[5] to annotate our sentences.

In total, we annotated 3,234 questions and 115,564 sentences from these datasets. About 43.3% of the questions did not have a single supporting sentence, indicating that

[3] We manually verified that 88% of retrieved sentences do not support the incorrect choice and 4% only partially support the answer choice. The residual 8% of the sentences either contradicted the question assertion or involved a question that itself was noisy.

[4] Using AI2 Science Questions v1 from http://allenai.org/data/science-exam-questions.html

[5] https://www.mturk.com

Question	Answer	Sentence (Premise)	Q+A as sentence (Hypothesis)	Label
When waves of two different frequencies interfere, what phenomenon occurs?	beating	Beats are the periodic and repeating fluctuations heard in the intensity of a sound when two sound waves of very similar frequencies interfere with one another.	When waves of two different frequencies interfere, beating occurs.	*entails*
Because trees add water vapor to air, cutting down forests leads to longer periods of what?	drought	During periods of drought, trees died and prairie plants took over previously forested regions.	Because trees add water vapor to air, cutting down forests leads to longer periods of drought.	*neutral*
What material comprises the sun and other stars, as well as lightning and the northern lights?	plasma	Our Sun and other stars are in the plasma state.	Plasma comprises the sun and other stars, as well as lightning and the northern lights.	–

Table 1: Randomly selected examples from the entailment dataset. The first sentence supports the right answer but also provides lot more information that needs to be ignored. Second example has some word overlap but can not be used to answer the question. In the third example, we only have partial support for the question, i.e., "Plasma comprises the sun and other stars."

these questions either need multiple sentences for question answering or better retrieval results. From the remaining 56.7% of the questions (1,834 questions), we obtained 10,101 examples with *entails* label and 16,925 examples with *neutral* label. The remaining sentences (33,792) were ambiguous for crowd-workersto annotate or only provided partial support. Our goal is to capture the obvious reasoning that a layman is able to perform, which in itself is a challenging task (as evidenced by our results). Hence these ambiguous sentences are ignored. To create an entailment dataset, we also had to reliably convert questions and answer choices into statements, i.e. the $h = assertion(q, a)$ function. We manually converted every question in our dataset into the best possible fill-in-the-blank statement (based on the answer choices) and then replaced the blanks with the given answer choice to create a valid assertion, h.

The final SCITAIL dataset contains 1,834 questions[6] with 10,101 *entails* examples and 16,925 *neutral* examples. As mentioned earlier, we do not have any examples with the *contradicts* label and only focus on the binary classification task. Some sample annotations are presented in Table 1. We next compare this dataset with previous published datasets to highlight some of the challenges relative to these dataset.

Dataset Size We compare SCITAIL against four popular datasets, listed chronologically.

1. RTE-6 (Bentivogli et al. 2010): The sixth PASCAL Recognizing Textual Entailment challenge dataset generated from news articles. Outputs from the summarization task from previous year are used to generate the hypothesis and sentences retrieved from a large news corpus are used as premises. While they have a large dev set with 16K examples, they only have 897 examples with *entails* label.
2. SICK (Marelli et al. 2014) (9.8K examples): The Sentences Involving Compositional Knowledge (SICK) dataset was created automatically using rules to capture compositional knowledge (active, passive, negation, etc).
3. SNLI (Bowman et al. 2015) (570K examples): Largest entailment dataset created by asking annotators to write statements that would be true (or false) for an image given its caption.
4. SCITAIL (27K examples): The dataset presented here using statements derived from independently written multiple choice questions and web sentences.

Token Lengths We compare the average number of tokens on the training sets for all the datasets in Table 2. We ignore the stop-words[7] and average the counts for premises and hypothesis per label. Apart from the RTE dataset, the number of tokens do not change much based on the gold label. Across all datasets, premises tend to be longer than the hypothesis which is expected for the entailment task. On average, SCITAIL dataset contains much longer premises and hypotheses (apart from the much smaller RTE-6dataset).

Datasets	Premise Length		Hypothesis length	
	entails	*neutral*	*entails*	*neutral*
RTE-6	17.96	15.41	5.60	7.19
SICK	5.09	5.04	4.58	5.07
SNLI	7.35	7.35	3.61	4.45
SCITAIL	10.79	10.28	6.69	7.01

Table 2: Average number of stop-word filtered tokens in the premise and hypothesis in the training set per gold label.

Premise vs. Hypothesis Next, we compare the premises and hypothesis tokens (as computed before) for each example in these datasets in Table 3. We calculate the proportions of hypothesis tokens that overlap between the hypothesis and the premise. In all the datasets, *entails* class tends to have a higher word overlap (as expected) with SCITAIL proportions being similar to that of SNLI. We also compute the difference between the premise and hypothesis tokens for each example. On an average, the *entails* example tend to have longer premises as also seen in the SCITAIL dataset. In general, SCITAIL is statistically similar to existing datasets with no easy wins by relying on overlap proportions or token lengths.

Final dataset Our final released dataset is available at http://data.allenai.org/scitail/ along with the raw annotations

[6] 4th grade: 102 qns, 8th grade: 83 qns, SciQ: 1649 qns

[7] As per NLTK English stopword list.

Datasets	Overlap proportion		Token difference	
	entails	*neutral*	*entails*	*neutral*
RTE-6	0.53	0.23	12.36	8.22
SICK	0.79	0.43	0.51	-0.03
SNLI	0.65	0.45	3.74	2.90
SCITAIL	0.67	0.48	4.11	3.28

Table 3: Average proportion of the hypothesis tokens that overlap with the premise and average difference between the number of tokens in premise and the hypothesis in the training set per gold label.

collected for all the questions. We use the same train/dev/test splits from the original question sets so that QA systems trained on this dataset can be evaluated against the original test questions. Table 4 gives the distribution of examples and questions in our splits. In addition, we also present the percentage of sentences with 'S'-rooted parses, percentage with at least one Open IE extraction and the number of distinct words in Table 5. Since the hypotheses in our datasets are created from a relatively small set of questions, we have much fewer unique words in the hypotheses as compared to the premises.

Splits	Examples	Questions	*entails*	*neutral*
Train	23,596	1,542	8,602	14,994
Dev	1,304	121	657	647
Test	2,126	171	842	1,284
Total	27,026	1834	10,101	16,925

Table 4: Distribution of entailment examples and underlying questions in the SCITAIL train/dev/test split.

	Premise	Hypothesis
'S'-rooted parses	89.5%	99.1%
Open IE	85.8%	96.5%
Distinct words	23,968	4,010

Table 5: Percentage of sentences with 'S'-rooted parse trees, percentage of sentences with at least one Open IE extraction, and number of distinct words in the SCITAIL dataset.

Decomposed Graph Entailment Model

As we show in Table 1, the examples in the SCITAIL dataset can be challenging for methods that ignore the semantics of the data. While LSTM embeddings used by current models can learn to capture the semantics, they need a much larger training set to do so. Even training the model on SNLI, a much larger but out-of-domain set, did not result in any improvement. We hypothesize that providing syntactic/semantic structure to the model can mitigate this issue.

However, the premises in our dataset are much longer than the previously published datasets and harder to parse. For example, Open IE (Banko et al. 2007) v4 [8] was able to parse only 85.8% of the premises on our test set, while it can extract 92.6% of the premises in the test set for SNLI. Note that this statistic only captures the failure to parse; the percentage of noisy extractions would be even larger. On the other hand, we can extract Open IE tuples from 96.5% of the hypotheses (as compared to 93.25% for SNLI).

Based on this analysis, we design a new entailment model that exploits structure from the hypothesis only. Instead of extracting structure from the premises that tend to be much longer and harder to parse, we focus on finding words in the premise that can *prove* the hypothesis structure. Our main goal with this proposed model is to show the value of structured representation on just the hypothesis for this task.

Graph Definition

We first start with extracting graph structure from the hypothesis. Given the success of Open IE on this domain (Khot, Sabharwal, and Clark 2017), we also use Open IE tuples as our graph representation for the hypothesis. However, our model can use any graph with labeled edges.

For each *(subject; predicate; obj(s))* tuple $T(S, P, O_i)$, we describe the edges added to our graph. The source and target nodes of these edges form the nodes in our graph. We create an edge between S and P named $subj$ and between S and O_1 named $subj - obj$. Since we use Open IE v4, it also extracts additional tags for certain objects (such as location and time). For such objects, we use these tags as the label for the edge from P to O_i. Also edges from P to objects beginning with a preposition (e.g., "through a system of tubes") are labeled with the corresponding preposition[9] (e.g., $through$) similar to the collapsed dependencies (de Marneffe and Manning 2008).

While phrases with prepositional attachment to the verbs are converted into an object, other prepositional phrases tend to be collapsed into a single object. For example, "to other parts of the plant" would be a single object and as a result a single node in our graph. To capture these within-object relations, we split objects using the same list of prepositions and add an edge from each prepositional phrase to the previous split phrase. For example, "to other parts of the plant" would be converted into "to other parts" – of→ "the plant". Finally, objects, O_i with no Open IE tags or recognized prepositions have edges from P labeled as "obj". We collect the edges and corresponding nodes from all the tuples in our hypothesis to get $\mathcal{G}_H = (V_H, E_H)$. Instead of computing the probability distribution over the output labels using the entire graph, we decompose this problem into first computing the node and edge probability distributions and aggregating them. These probability distributions basically capture the probability of a node (or edge) in the hypothesis being supported/not supported by the premise.

Node Attention

To compute whether a node in the graph is supported by the premise words, we first identify the premise words similar to the node. We compute the attention (Bahdanau, Cho, and Bengio 2015) of the words in the node over the words in

[8] https://github.com/allenai/openie-standalone

[9] We only consider a fixed list of prepositions; cf. Appendix.

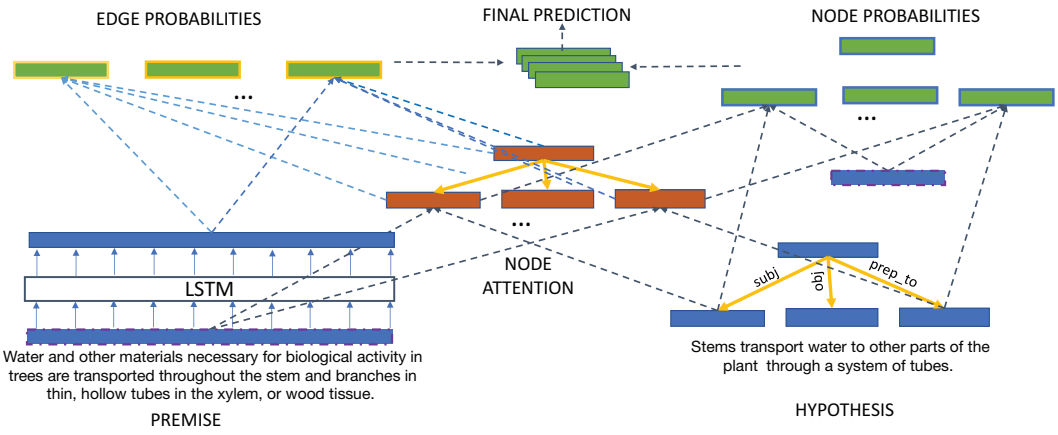

Figure 3: The decomposable graphical model architecture. Blue rectangles are used to indicate vector representations of words, brown rectangles indicate attention vectors and green rectangles indicate distribution over the output labels.

the premise. Similar to the decomposable attention model, we use the raw word embeddings of the premise to compute this attention. Consider a node, $v_i \in V_H$ with J words $\{h_{ij}\}$. The attention weight for the hypothesis word h_{ij} on the premise word p_k in the premise (of length K) is computed as: $\alpha_{ijk} = h_{ij} \cdot p_k$

We compute the normalized attention, ω_{ijk} for the word, h_{ij} and average the attentions from all the words in the node, $v_i = \{h_{ij}\}$ to compute the node attention , ρ_{ik}:

$$\omega_{ijk} = \frac{\exp(\alpha_{ijk})}{\sum_{k'} \exp(\alpha_{ijk'})} \qquad \rho_{ik} = \frac{1}{J} \sum_j \omega_{ijk} \tag{1}$$

Node Probabilities

Next we calculate the probability of each node being supported by the premise by comparing the vector representation of the node with an attention-weighted representation of the premise. We compute the hypothesis node representation, $\hat{v}_i$ and the weighted premise representation, $\hat{\rho}_i$ as:

$$\hat{v}_i = \frac{1}{J} \sum_j h_{ij} \qquad \hat{\rho}_i = \sum_k \rho_{ik} p_k$$

Intuitively, if the words in a node are present in the premise, the attention-weighted vector representation $\hat{\rho}_i$ should be similar to average vector representation $\hat{v}_i$. We compare these vector representations using a single-linear perceptron, f_v with dropout in the output layer (Srivastava et al. 2014) and ReLU (Glorot, Bordes, and Bengio 2011) activation function. Along with their vector representations, we use element-wise difference and products (Mou et al. 2016) as inputs to this linear perceptron.

$$\Pr(v_i) = f_v([\hat{v}_i; \hat{\rho}_i; \hat{v}_i - \hat{\rho}_i; \hat{v}_i * \hat{\rho}_i]) \tag{2}$$

The linear perceptron outputs a two-dimensional vector corresponding to the final output labels: {*entails*, *neutral*}. While our dataset only contains two output classes, the model can be easily extended for the three-way entailment classification task.

Edge Probabilities

The edges in our hypothesis graph capture relations between the nodes and computing the support of an edge corresponds to extracting this relation from the hypothesis. Standard relation identification models (Ji and Grishman 2011) assume the entity spans are provided as inputs to the model. Since we do not have the exact spans in the hypothesis corresponding to the nodes in the edge, we use the previously computed attention, $\hat{\rho}_{ik}$ to identify *soft* spans. Also, relation extraction methods (Miwa and Bansal 2016) commonly use the output embeddings from an LSTMs (Hochreiter and Schmidhuber 1997) to calculate context-dependent representation of the entities. Hence, we compute the representation for each node using the attention-weighted representation of the LSTM embeddings for each word in the premise, $\tilde{p}_k$ as $\tilde{\rho}_i = \sum_k \rho_{ik} \tilde{p}_k$.

We also learn an n-dimensional embedding for each edge label, emb_e. Given an edge $e_l = (v_s, label, v_t)$, we compute the edge probability using the LSTM-weighted embedding $\tilde{p}$ and edge embedding as

$$\Pr(e_l) = g_e([\tilde{\rho}_s; emb_e(label); \tilde{\rho}_t]) \tag{3}$$

We use a single-layer perceptron with a ReLU activation function and dropout in the output layer.

Final Prediction

Finally we combine the probability predictions on the nodes and edges to get the final prediction by averaging the node and edge probabilities.

$$\Pr(\mathcal{V}_H; P) = \frac{1}{|\mathcal{V}_H|} \sum_{v_i \in \mathcal{V}_H} \Pr(v_i)$$

$$\Pr(\mathcal{E}_H; P) = \frac{1}{|\mathcal{E}_H|} \sum_{e_l \in \mathcal{E}_H} \Pr(e_l)$$

$$\Pr(\mathcal{G}_H; P) = Pr(\mathcal{V}_H; P) + Pr(\mathcal{E}_H; P)$$

Note that we use *probabilities* to intuitively describe the outputs of the node and edge modules. These "probabilities"

range between $(-\infty, \infty)$ i.e. correspond to the logit function value of the actual probability values. We finally use the cross-entropy loss on the logit graph probability $\Pr(\mathcal{G}_H; P)$ to train this model.

Implementation Details

We implement our model using AllenNLP toolkit[10] (Gardner et al. 2017) in PyTorch.[11] We use the 300-dimensional 840B Glove embeddings (Pennington, Socher, and Manning 2014) projected down to 100 dimensions. We set the dimensionality of the hidden vectors in LSTM and MLP_e as 100. We used the cross-entropy loss with Adam optimization (Kingma and Ba 2015). We halved the learning rate at every epoch and used early-stopping (patience=20) based on the validation set accuracy. We set the dropout to 0.5 and the edge embedding dimensionality to 10. We selected these parameters based on the accuracies on the validation set. Our implementation is also available from the dataset page at http://data.allenai.org/scitail.

Experiments

We compare our system against two state-of-the-art neural entailment systems along with a simple overlap-based model trained on the SCITAIL dataset.

Baselines

Decomposable Attention Model (DecompAtt) (Parikh et al. 2016): A simple model that decomposes the problem into parallelizable attention computations. We used the AllenNLP (Gardner et al. 2017) implementation of the decomposable attention model with 341K parameters

Enhanced LSTM (ESIM) (Chen et al. 2017): Enhanced Sequential Inference model using only sequential information[12] with 4.3M parameters.

Ngram Overlap: We also implement a simple word-overlap baseline to show that simple overlap measures are not sufficient. We compute the proportion of unigrams, 1-skip bigrams, and 1-skip trigrams (Guthrie et al. 2006) in the hypothesis that are also present in the premise as three features[13]. We feed these features into a two-layer perceptron(hidden dimension=2). (20 parameters).

DGEM: Our proposed decomposed graph entailment model with 112K parameters.

Results

Table 6 shows the accuracies of the baseline systems on this dataset. State-of-the-art neural methods achieve 10-12% above the majority class baseline. Surprisingly the ngram-based model, is also able to achieve similar results on the test set.[14] This shows the sequence-based neural models barely capture any semantics. On the other hand, our structure-based approach is able to achieve about 5% gain over the best baseline system on this task. The important of considering structure is further illustrated by the drop in test accuracy when we ignore the edge probabilities in our model.

Models	Validation Accuracy	Test Accuracy
Majority class	63.3	60.3
DecompAtt	75.4	72.3
ESIM	70.5	70.6
Ngram	65.0	70.6
DGEM w/o edges	75.1	70.8
DGEM	**79.6**	**77.3**

Table 6: Validation and test set accuracy on the entailment dataset. Our proposed models outperforms the state-of-the-art by exploiting the structure of the hypothesis.

Qualitative Analysis

For some insight into the model, we depict in Figure 4 the node attention and probabilities computed by our model on one of the examples in our dev set. Even though the model is not able to find support for the phrase '*from water droplets*', it is able to use the edge probability model on the LSTM-embeddings to identify the 'from' relation between '*water droplets*' and '*are formed*'.

Comparison to Decomposable Attention Next we present two cases where the decomposable attention model incorrectly labels the example but our model is able to use structure to accurately label the example. Consider the following *entails* example:

> *premise*: Upwelling upward movement of deep (abyssal), cold water to the surface.
> *hypothesis*: Upwelling is the term for when deep ocean water rises to the surface.

While most of the hypothesis words can be found in the premise, the key phrase "is the term for" is mostly missing. Our model is able to use the learned edge predictor model to identify this relation, where as the decomposable attention model mainly relies on word attentions and labels this incorrectly. On the other hand, consider the *neutral* example:

> *premise*: If the conditions are not cold enough, the precipitation will be rain.
> *hypothesis*: Precipitation commonly occur(s) along a cold front.

Here, the decomposable attention model incorrectly predicts this as *entails* due to the largely similar words in the sentences. Our model does not find any support for the edges in the hypothesis, specifically the (commonly occurs -subj→ Precipitation) and (commonly occurs -along→ a cold front) edge and correctly predicts the *neutral* label.

Error Analysis

We analyzed 10 false positives (*neutral* examples marked as *entails*) and 10 false negatives (*entails* examples marked as

[10] http://allennlp.org
[11] http://pytorch.org
[12] From https://github.com/lukecq1231/nli
[13] stemmed and stop-word filtered
[14] MaxEntropy-based model from the Excitement Platform (Magnini et al. 2014) also achieved similar scores on this dataset.

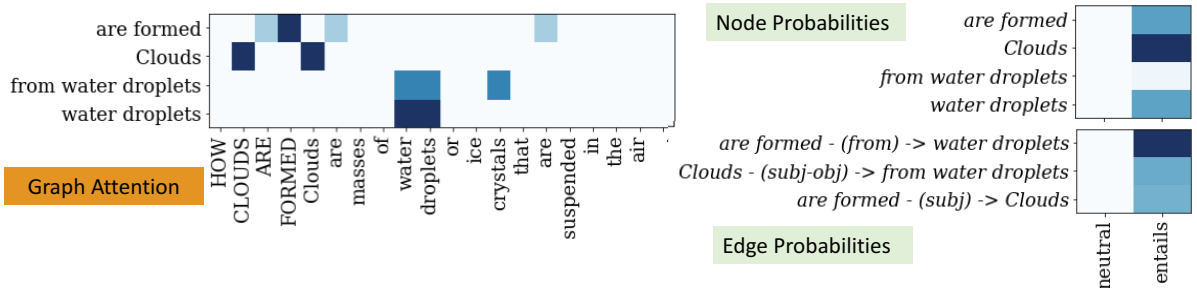

Figure 4: Attention map and graph probabilities on a sample entailment example where *premise*="*HOW CLOUDS ARE FORMED Clouds are masses of water droplets or ice crystals that are suspended in the air.*" and *hypothesis*="*Clouds are formed from water droplets.*" While the model does not find the phrase "from water droplets" as indicated by the low *entails* probability, it recognizes the "from" relation between "are formed" and "water droplets". Thus, even though the premise does not explicitly mention '*from*', our edge prediction model is able to identify this implicit relation using the context.

neutral) from the dev set. We identified five key causes of errors ignoring one-off error cases.[15]

Noisy structure: In 35% of the examples, noisy extracted structure from the hypothesis resulted in incorrect predictions. For example, the Open IE extraction from "*The energy content of foods is often expressed in calories.*" does not contain the key phrase '*in calories*'[16].

Term Importance: In 20% of the examples, a key term from the question is not supported by the premise but the model still predicts the *entails* label due to other well-supported terms. For example, '*invisible gas*' is the key term that needs to be supported for the hypothesis "*Water vapor exists in the atmosphere as an invisible gas*". However, the model still identifies "*of the water budget is present as a gas (water vapor) in the atmosphere.*" as a supporting hypothesis. Identifying essential terms (Khashabi et al. 2017) from the hypothesis could be one way to mitigate this problem.

Misleading premise words: In 15% of the examples, our model fails to identify the key phrase in the hypothesis to *attend* over for the node attention due to multiple similar words. For example, the model has very low attention probabilities over the premise words '*are a key contributor*' for the hypothesis node '*is the main cause of*' in the premise due to other very similar words in the premise.

Long phrases: In 10% of the examples, averaged per-word attention for longer phrases was not very useful as the attention is spread across the entire sentence. Instead of a simple averaged attention, a model using the parse structure or the head-word of the phrase can likely avoid this issue.

Hard entailment: In 10% of the examples, the model needed subtle reasoning to be able to infer the true label. For example, the premise "*Viruses have Nucleic Acids and Proteins but lack other features of living cells.*" can be used to conclude "*Nucleic acids are found in all living cells and viruses.*" even though it is not explicitly stated.

[15]In one example, the hypothesis tokens had noisy HTML characters and in the second example, the annotator labeled assumed domain knowledge that was not stated in the premise.

[16]Open IE: (The energy content of foods; is expressed; T:often)

Conclusion

We present a new natural dataset for textual entailment, SCITAIL, derived directly from an end task, namely that of Science question answering. We show that this is a challenging dataset for current state-of-the-art models. We propose a new neural entailment architecture that can use any graph-based syntactic/semantic structure from the hypothesis. This additional use of structure results in 5% improvement on this dataset. We hope that this model sets a strong baseline for achieving further gains on SCITAIL in the near future, helping the field make progress in reasoning with complex, natural language. Exploring other possible syntactic representations for the hypothesis and comparing against newly developed approaches for using structure (Chen et al. 2017) remain interesting directions for future work, as does the translation of improvements on this entailment sub-task to more effective question-answering systems for the Science domain.

Acknowledgments

The authors would like to thank Dongyeop Kang, Oyvind Tafjord, Luke Zettlemoyer for valuable discussions and help throughout the project.

Appendix: Complete list of edge labels

Prepositions := {"with", "at", "from", "into", "during", "including", "until", "against", "among", "throughout", "despite", "towards", "upon", "concerning", "of", "to", "in", "for", "on", "by", "about", "like", "through", "over", "before", "between", "after", "since", "without", "under", "within", "along", "following", "across", "behind", "beyond", "plus", "except", "but", "up", "out", "around", "down", "off", "above", "near"}

Special Open IE tags := "L:", "T:"

Argument edges := "subj", "subj-obj", "obj"

References

Androutsopoulos, I., and Malakasiotis, P. 2010. A survey of paraphrasing and textual entailment methods. *J. Artif. Intell. Res.* 38:135–187.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Banko, M.; Cafarella, M. J.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction from the web. In *IJCAI*.

Bentivogli, L.; Clark, P.; Dagan, I.; and Giampiccolo, D. 2010. The Sixth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for natural language inference. In *ACL*.

Clark, P.; Etzioni, O.; Khot, T.; Sabharwal, A.; Tafjord, O.; Turney, P.; and Khashabi, D. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*.

Dagan, I.; Roth, D.; Sammons, M.; and Zanzotto, F. M. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies* 6(4):1–220.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL Recognising Textual Entailment Challenge. In *MLCW*.

de Marneffe, M.-C., and Manning, C. D. 2008. The Stanford typed dependencies representation.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2017. AllenNLP: A deep semantic natural language processing platform. Technical report.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *AISTATS*.

Guthrie, D.; Allison, B.; Liu, W.; Guthrie, L.; and Wilks, Y. 2006. A closer look at skip-gram modelling. In *LREC*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 98:1735–80.

Ji, H., and Grishman, R. 2011. Knowledge base population: Successful approaches and challenges. In *ACL*.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. S. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Khashabi, D.; Khot, T.; Sabharwal, A.; Clark, P.; Etzioni, O.; and Roth, D. 2016. Question answering via integer programming over semi-structured knowledge. In *IJCAI*.

Khashabi, D.; Khot, T.; Sabharwal, A.; and Roth, D. 2017. Learning what is essential in questions. In *CoNLL*, 80–89.

Khot, T.; Balasubramanian, N.; Gribkoff, E.; Sabharwal, A.; Clark, P.; and Etzioni, O. 2015. Exploring Markov logic networks for question answering. In *EMNLP*.

Khot, T.; Sabharwal, A.; and Clark, P. 2017. Answering complex questions using open information extraction. In *ACL*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Magnini, B.; Zanoli, R.; Dagan, I.; Eichler, K.; Neumann, G.; Noh, T.-G.; Padó, S.; Stern, A.; and Levy, O. 2014. The Excitement Open Platform for textual inferences. In *ACL*.

Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; and Zamparelli, R. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*.

Miwa, M., and Bansal, M. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *ACL*.

Mou, L.; Men, R.; Li, G.; Xu, Y.; Zhang, L.; Yan, R.; and Jin, Z. 2016. Natural language inference by tree-based convolution and heuristic matching. In *ACL*.

Parikh, A. P.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A decomposable attention model for natural language inference. In *EMNLP*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Richardson, M.; Burges, C. J. C.; and Renshaw, E. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Welbl, J.; Liu, N. F.; and Gardner, M. 2017. Crowdsourcing multiple choice science questions. In *Workshop on Noisy User-generated Text*.

Zhao, K.; Huang, L.; and Ma, M. 2016. Textual entailment with structured attentions and composition. In *COLING*.