Learning Answer-Entailing Structures for Standardised Tests

Mrinmaya Sachan
Machine Learining Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
mrinmays@cs.cmu.edu

December 14, 2016

Abstract

Background Recently, researchers have proposed standardised tests used for elementary learning as "drivers for progress in AI" (Clark & Etzioni, 2016). These tests are widely accessible, easily measurable, and support incremental progress. Yet solving them would require significant advances in language understanding and world modelling. Hence, these tests can serve as perfect test-beds for building the next generation of knowledge driven AI applications.

Aim In this paper, we aim to solve two of these standardized tests – namely, reading comprehensions and elementary science question answering. Both these tests require a significant level of language understanding, background knowledge modelling and reasoning. Given a corpus of passages and multiple-choice questions based on them for the reading comprehension task or a corpus of multiple-choice elementary science questions and some instructional material (textbooks, study guides, etc) for the science QA task, we build a system that learns to solve novel reading comprehensions and elementary science tests.

Data We used freely available datasets for both tasks: (a) MCTest-500 dataset, a freely available set of 500 passages (split into 300 train, 50 dev and 150 test) and associated questions (Richardson et al., 2013), and (b) 2500 8^{th} grade science questions (split into 1500 train, 500 dev and 500 test) released by the *Allen Institute of AI* as a part of their *Aristo* project.

Methods We posit that there is a hidden (latent) structure that explains the relation between the question, correct answer, and the texts. We call this the *answer-entailing structure*; given the structure, the correctness of the answer is evident. Since the structure is latent, it must be inferred. We present a unified max-margin framework that learns to find these hidden structures (given a corpus of question-answer pairs), and uses what it learns to answer novel questions. We extend this framework to incorporate multi-task learning on the different sub-tasks that are required to solve these standardized tests.

Results Evaluation on the publicly available datasets described above shows that our framework of solving standardized tests via latent answer-entailing structures outperforms various IR and neural-network baselines and achieves the state-of-the-art on both these tasks. While these structures are useful proxies for the semantics required to solve these tasks, they have their own limitations. So, we also analyzed the strengths and weaknesses of these structures on the 20 subtasks for machine comprehension proposed in Weston et. al. (2015).

Conclusion The strategy of learning latent answer-entailing structures works well in practice for standardized tests as it transforms the difficult question answering tasks into a more familiar structure learning problem. These structures are cheap proxies to the understanding and reasoning required for these tasks. However, this technique has its own limitations. We described the benefits and limitations of this approach with a more fine-grained analysis.

1 Introduction

Standardized tests have often been proposed as "drivers for progress in AI" (Clark & Etzioni, 2016). These include reading comprehension tests (Richardson et al., 2013), science question answering (Clark, 2015), algebra word problems (Kushman et al., 2014), geometry problems (Seo et al., 2014, 2015), etc. These tests are widely accessible, easily comprehensible, clearly measurable, and offer a graduated progression from simple tasks to those requiring deep understanding of the world. This makes them perfect testbeds for research in knowledge-driven AI.

In this paper, we propose a latent structure learning approach for two standardized tests – reading comprehension tests (Richardson et al., 2013) and science question answering (Clark, 2015). Reading comprehension tests evaluate a machine's understanding by posing a series of reading comprehension questions and associated passages, where the answer to each question can be found only in its associated passage. Despite significant recent interest (Burges, 2013; Weston et al., 2014, 2015), the reading comprehension task remains unsolved. On the other hand, the science question answering task (Clark, 2015) evaluates the system's ability to answer multiple-choice elementary science questions given access to the necessary textbooks and other instructional materials. These science tests are challenging because a wide variety of background knowledge and reasoning is required to answer them. Despite some recent interest (Khot et al., 2015; Li & Clark, 2015; Clark et al., 2016), the science question answering task too is far from being solved.

Our approach learns latent *answer-entailing structures* that can help us answer the questions. The answer-entailing structures in our model are closely related to the inference procedure used in various models for machine translation (Blunsom & Cohn, 2006), textual entailment (MacCartney et al., 2008), paraphrase (Yao et al., 2013), question answering (Yih et al., 2013), etc. and correspond to the best (latent) alignment between a hypothesis (formed from the question and a candidate answer) with appropriate snippets in the texts that are required to answer the question. Examples of answer-entailing structures for the two tasks is given in Figure 1 and Figure 2, respectively.

There are some key differences between the answer-entailing structures considered here and the alignment structures considered in previous works in question answering (QA). First, we can align multiple sentences in the texts (passage or textbooks) to the hypothesis. The sentences in the texts considered for alignment are not restricted to occur contiguously in the texts. To allow such a dis-contiguous alignment, we make use of the document structure; in particular, we take help from rhetorical structure theory (Mann & Thompson, 1988) and event and entity coreference links across sentences. For the science QA task, our model has some additional key novelties: we incorporate the student's curriculum hierarchy (i.e. the book, chapter, section bifurcation) into the latent structure. This helps us jointly learn the retrieval and answer selection modules of a question answering system. Retrieval and answer selection are usually designed as isolated or loosely connected components in QA systems (Ferrucci, 2012) leading to loss in performance – our approach mitigates this. Modern textbooks typically provide a set of review questions after each section to help students understand the material better. We also make use of these review problems to further improve our model. These review problems have value as part of the latent structure is known for these questions. Finally, we also utilize domain-specific knowledge sources such as study guides, science dictionaries or semi-structured knowledge tables within our model.

Modelling the inference procedure via answer-entailing structures is a crude yet effective and computationally inexpensive proxy to model the semantics needed for the problem. Learning these latent structures can also be beneficial as they can assist a human in verifying the correctness of the answer, eliminating the need to read lengthy texts.

The overall model is trained in a max-margin fashion using a latent structural SVM (Yu & Joachims, 2009, LSSVM) where the answer-entailing structures are latent. We also extend our LSSVM to multi-task settings using a top-level question-type classification. Many QA systems include a question classification component (Li & Roth, 2002; Zhang & Lee, 2003), which typically divides the questions into semantic

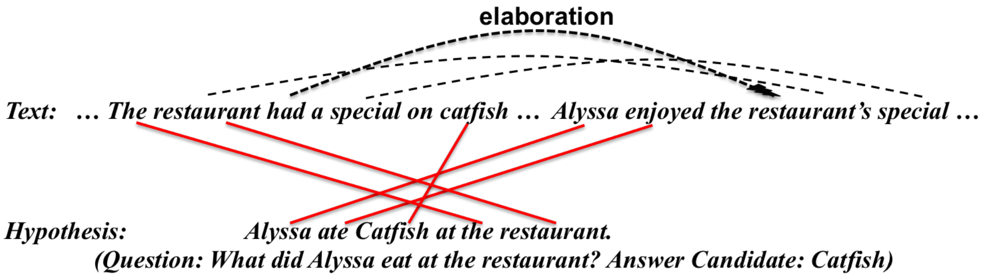
elaboration
Text: … The restaurant had a special on catfish … Alyssa enjoyed the restaurant's special …
Hypothesis: Alyssa ate Catfish at the restaurant.
(Question: What did Alyssa eat at the restaurant? Answer Candidate: Catfish)

Figure 1: An example *answer-entailing structure* from the MCTest500 dataset. The question and answer candidate are combined to generate a hypothesis sentence. Then latent alignments are found between the hypothesis and the appropriate snippets in the text. The solid red lines show the word alignments from the hypothesis words to the passage words, the dashed black lines show auxiliary co-reference links in the text and the labelled dotted black arrows show the RST relation (elaboration) between the two sentences. Note that the two sentences do not have to be contiguous sentences in the text. We provide some more examples of *answer-entailing structures* in the supplementary.

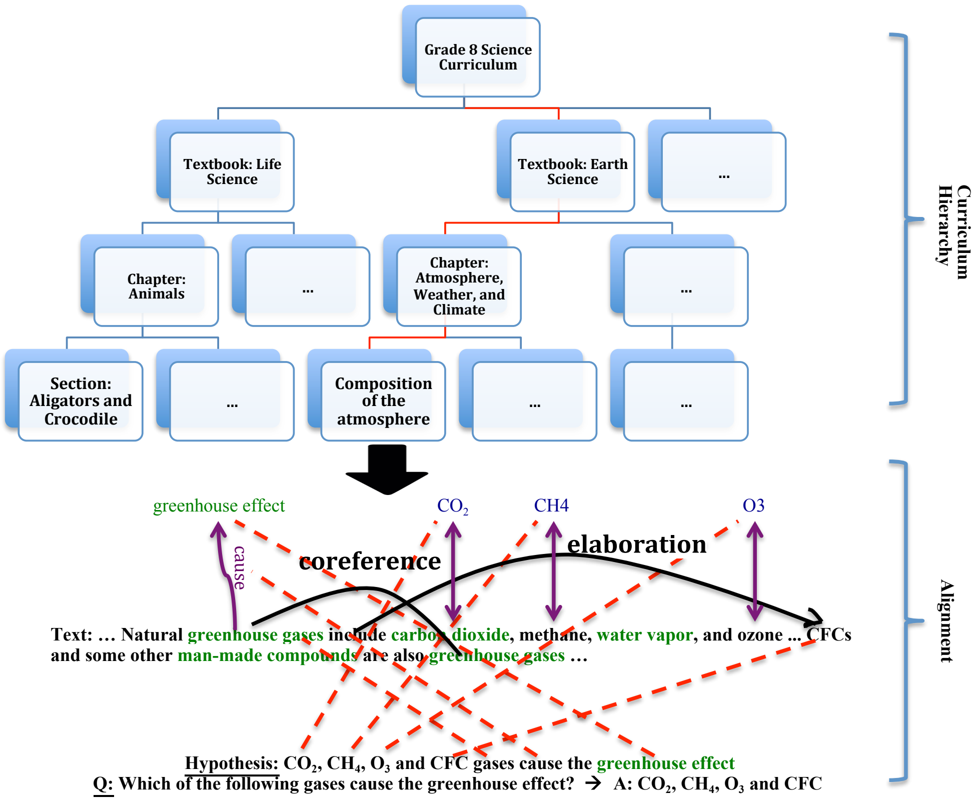
Grade 8 Science Curriculum
Textbook: Life Science
Textbook: Earth Science
...
Chapter: Animals
...
Chapter: Atmosphere, Weather, and Climate
...
Section: Aligators and Crocodile
...
Composition of the atmosphere
...
...
Curriculum Hierarchy
greenhouse effect
cause
CO$_2$
CH4
O3
coreference
elaboration
Text: … Natural greenhouse gases include carbon dioxide, methane, water vapor, and ozone … CFCs and some other man-made compounds are also greenhouse gases …
Hypothesis: CO$_2$, CH$_4$, O$_3$ and CFC gases cause the greenhouse effect
Q: Which of the following gases cause the greenhouse effect? → A: CO$_2$, CH$_4$, O$_3$ and CFC
Alignment

Figure 2: An example *answer-entailing structure* from the science question answering dataset. The answer-entailing structure consists of selecting a particular textbook from the curriculum, picking a chapter in the textbook, picking a section in the chapter, picking sentences in the section and then aligning words/mwe's in the hypothesis (formed by combining the question and an answer candidate) to words/mwe's in the picked sentences or some related "knowledge" appropriately chosen from additional knowledge stores. In this case, the relation (greenhouse gases, cause, greenhouse effect) and the equivalences (e.g. carbon dioxide = CO_2) – shown in violet – are hypothesized using external knowledge resources. The dashed red lines show the word/mwe alignments from the hypothesis to the sentences (some word/mwe are not aligned, in which case the alignments are not shown), the solid black lines show coreference links in the text and the RST relation (elaboration) between the two sentences. The picked sentences do not have to be contiguous sentences in the text. All mwe's are shown in green.

categories based on the type of the question or answers expected. This helps the system impose some constraints on the plausible answers. Standardized tests too can benefit from such a pre-classification step, not only to constrain plausible answers, but also to allow the system to use different processing strategies for each category. By using the multi-task setting, our learner is able to exploit the commonality among tasks where possible, while having the flexibility to learn task-specific parameters where needed. To the best of our knowledge, this is the first use of multi-task learning in a structured prediction model for QA.

We validate for our model on two real-world datasets – (a) MCTest (Richardson et al., 2013), and (b) 8^{th} grade science QA dataset (Clark et al., 2016), and achieve superior performance vs. a number of IR and neural network baselines on both. These works are already published (Sachan et al., 2015, 2016). Extensions that use the abstract meaning representation (Banarescu et al., 2013) and curriculum learning (Bengio et al., 2009) to improve these models were later published (Sachan & Xing, 2016b, 2016a) but not is described in this article.

2 Related Work

The field of QA is quite rich. Traditionally, there has been a large body of research that has focused on short factoid questions such as "Who is the president of the United States?". Factoid question answering contains questions about some factual knowledge which can usually be answered by querying the web or existing knowledge tables. Since it is impossible to review all the factoid QA research in this document, we point the interested reader to the TREC[1] and CLEF[2] evaluations. Recently, there has been a resurgence of non-factoid QA in the form of reading comprehensions (QA4MRE evaluations[3], MCTest (Richardson et al., 2013) and bAbI (Weston et al., 2015) datasets are notable examples). Non Factoid QA focuses on answering questions that are not fact based. They require solutions that can "understand" the content rather than using IR style solutions or solutions that use the redundancy of the web to answer questions. This is one of the main reasons for the growing interest in Non Factoid QA.

Recently, researchers have proposed standardised tests as drivers for progress in AI (Clark & Etzioni, 2016). Some example standardised tests are reading comprehensions (Richardson et al., 2013), algebra word problems (Kushman et al., 2014), geometry problems (Seo et al., 2014), entrance exam tests (Fujita et al., 2014; Arai & Matsuzaki, 2014), etc. These tests are usually in the form of question-answers and focus on elementary school learning. Our work focuses on reading comprehensions and 8^{th} grade science QA which are both standardised tests.

In this paper, we present a strategy for learning answer-entailing structures that help us perform inference over much longer texts by treating this as a structured input-output problem. The approach of treating a problem as one of mapping structured inputs to structured outputs is common across many NLP applications. Examples include word or phrase alignment for bitexts in MT (Blunsom & Cohn, 2006), text-hypothesis alignment in RTE (Sammons et al., 2009; MacCartney et al., 2008; Yao et al., 2013; Sultan et al., 2014), question-answer alignment in QA (Berant et al., 2013; Yih et al., 2013; Yao & Van Durme, 2014), etc. All of these approaches align local parts of the input to local parts of the output. In this work, we extended the word alignment formalism to align multiple sentences in the text to the hypothesis. We also incorporated the document structure (rhetorical structures (Mann & Thompson, 1988)) and co-reference information to help us perform inference over longer documents.

QA has had a long history of using pipeline models that extract a limited number of high-level features from induced representations of question-answer pairs, and then build a classifier using some labelled corpora. On the other hand, we learn these structures and the model for answering standardized test questions

[1] http://trec.nist.gov/

[2] http://nlp.uned.es/clef-qa/

[3] http://nlp.uned.es/clef-qa/repository/qa4mre.php

jointly through a unified max-margin framework. We note that there exist some recent models such as Yih et. al. (2013) that do model QA by automatically defining some kind of alignment between the question and answer snippets and use a similar structured input-output model. However, they are limited to single sentence matching to determine answers.

Another advantage of our approach is its simple and elegant extension to multi-task settings as a way to combine the retrieval and alignment model. There has been a rich vein of work in multi-task learning for SVMs in the ML community. Evgeniou and Pontil (2004) proposed a multi-task SVM formulation assuming that the multi-task predictor $\mathbf{w}$ factorizes as the sum of a shared and a task-specific component. We used the same idea to propose a multi-task variant of Latent Structured SVMs. This allows us to use the single task SVM in the multi-task setting with a different feature mapping. This is much simpler than other competing approaches such as Zhu et. al. (2011) proposed in the literature for multi-task LSSVM.

3 Problem Definition

Standardized tests typically require the system to answer questions based on some background knowledge (passage, science textbooks, external resources, etc.). For each question $q_i \in Q$, let $A_i = \{a_{i1}, \ldots, a_{im}\}$ be the set of candidate answers to the question. Let a_i^* be the correct answer. The candidate answers may be pre-defined, as is the case in multiple-choice QA, or may be undefined but easy to extract with a high degree of confidence (e.g., by using a pre-existing system). We assume that each question has exactly one correct answer. We want to learn a function $f : (q, \mathcal{K}) \rightarrow a$ that, given a question q_i and background knowledge $\mathcal{K}$, outputs an answer $\hat{a}_i \in A_i$. The background knowledge is task and solution dependant. In the reading comprehension task, the background knowledge is the passage corresponding to the question itself. In the science question answering task, the science textbooks and the instructional materials form the background knowledge.[4]

4 The Solution

4.1 Modeling Standardized Tests as a Textual Entailment Problem

A key idea in this paper is that we can cast the task of solving the given standardized test as a textual entailment problem by converting each question-answer candidate pair $(q_i, a_{i,j})$ into a hypothesis statement h_{ij}. For example, the question "What did Alyssa eat at the restaurant?" and answer candidate "Catfish" in Figure 1 can be combined to achieve a hypothesis "Alyssa ate Catfish at the restaurant". We use a set of question matching/rewriting rules to achieve this transformation. These rules match the question into one of a large set of pre-defined templates and apply a unique transformation to the question and answer candidate to achieve the hypothesis statement. Hence, for each question q_i, the task reduces to picking the hypothesis $\hat{h}_i$ that has the highest likelihood of being entailed by the text among the set of hypotheses $\mathbf{h}_i = \{h_{i1}, \ldots, h_{im}\}$ generated for that question. Let $h_i^* \in \mathbf{h}_i$ be the correct hypothesis. Now let us define the latent answer-entailing structures in more detail.

4.2 Latent Answer-Entailing Structures

The latent answer-entailing structures help the model in providing evidence for the correct hypothesis. We consider the quality of one-to-one word alignment from the hypothesis to snippets in the texts as a proxy for the evidence. Hypothesis words are aligned to a unique word in the text or an empty word. For example,

[4] Other linguistic resources (taggers, parsers, chunkers, etc.) are also used to generate features. Hence, in spirit, they can also be considered as background knowledge.

in the reading comprehension example (Figure 1), all words but “at” are aligned to a word in the text. The word “at” can be assumed to be aligned to an empty word and it has no effect on the model. Learning these alignment edges typically helps a model decompose the input and output into semantic constituents and determine which constituents should be compared to each other. These alignments can then be used to generate more effective features.

The alignment depends on two things: (a) snippets in the text to be aligned to the hypothesis and (b) word alignment from the hypothesis to the snippets.

Snippet Selection: Determining the snippets in the text to be aligned to the hypothesis is a crucial step. For the reading comprehension task, we explore two variants of the snippets in the text to be aligned to the hypothesis:

1. *Sentence:* The simplest variant is to find a single sentence in the text that best aligns to the hypothesis. This is the structure considered in a majority of previous works in RTE (MacCartney et al., 2008) and QA (Yih et al., 2013) as they only reason on single sentence length texts.

2. *Subset:* Here we find a subset of sentences from the text (instead of just one sentence) that best aligns with the hypothesis.

For the science question answering task, the snippet selection also takes the curriculum structure and other instructional material into account. In this case, the snippet from the curriculum to be aligned to the hypothesis is determined by walking down the curriculum hierarchy and then picking a set of sentences from the section chosen. Then, a subset of relevant external knowledge in the form of triples and equivalences (called knowledge bits) is selected from our reservoir of external knowledge (science dictionaries, cheat sheets, semi-structured tables, etc). Finally, words in the hypothesis are aligned to words in the snippet or knowledge bits. Since, a lot of scientific concepts (e.g. carbon dioxide) are multi-word expressions, we detected multi-word expressions using jMWE (Kulkarni & Finlayson, 2011). Then, we align words/multi-word expressions in the hypothesis to words/multi-word expressions in the chosen snippet for science question answering. See Figure 2 for an example. The choice of the snippets composed with the word alignment is the resulting hidden structure called the answer-entailing structure.

4.3 The Structured Prediction Model

A natural solution is to treat answer selection as a structured prediction problem of ranking the hypotheses $\mathbf{h}_i$ such that the correct hypothesis is at the top of this ranking.We learn a scoring function $S_{\mathbf{w}}(h, \mathbf{z})$ with parameter $\mathbf{w}$ such that the score of the correct hypothesis h_i^* and the corresponding best latent structure $\mathbf{z}_i^*$ is higher than the score of the other hypotheses and their corresponding best latent structures. In fact, in a max-margin fashion, we want that $S_{\mathbf{w}}(h_i^*, \mathbf{z}_i^*) > S(h_{ij}, \mathbf{z}_{ij}) + 1 - \xi_i$ for all $h_j \in \mathbf{h} \setminus h^*$ for some slack ξ_i. Writing the relaxed max margin formulation:

$$\min_{||\mathbf{w}||} \quad \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_i \left(\max_{\mathbf{z}_{ij}, h_{ij} \in \mathbf{h}_i \setminus h_i^*} S_{\mathbf{w}}(h_{ij}, \mathbf{z}_{ij}) + \Delta(h_i^*, h_{ij}) - S_{\mathbf{w}}(h_i^*, \mathbf{z}_i^*) \right) \tag{1}$$

We use 0-1 cost, i.e. $\Delta(h_i^*, h_{ij}) = \mathbb{1}(h_i^* \neq h_{ij})$ If the scoring function is convex then this objective is in concave-convex form and hence can be solved by the concave-convex programming procedure (CCCP) (Yuille & Rangarajan, 2003). We assume the scoring function to be linear:$S_{\mathbf{w}}(h, \mathbf{z}) = \mathbf{w}^T \psi(h, \mathbf{z})$. Here, $\psi(h, \mathbf{z})$ is a feature map discussed later. The CCCP algorithm essentially alternates between solving for $\mathbf{z}_i^*$, $\mathbf{z}_{ij}\ \forall j$ s.t. $h_{ij} \in \mathbf{h}_i \setminus h_i^*$ and $\mathbf{w}$ to achieve a local minima. In the absence of information regarding the latent structure $\mathbf{z}$ we pick the structure that gives the best score for a given hypothesis i.e. $\arg\max_z S_{\mathbf{w}}(h, z)$. The complete procedure is given in Algorithm 1.

Algorithm 1 Alternate Minimization for LSSVM

```
Initialise $\mathbf{w}$
$C_i = \emptyset \quad \forall i = 1 \ldots n$
repeat
    for $i = 1, \ldots, n$ do
        $\mathbf{z}_i^* = \arg\max_{\mathbf{z}} S_{\mathbf{w}}(h_i^*, \mathbf{z})$
        for $h_{ij} \in \mathbf{h}_i \setminus h_i^*$ do
            $\mathbf{z}_{ij} = \arg\max_{\mathbf{z}} S_{\mathbf{w}}(h_{ij}, \mathbf{z})$
        $h_i^t, \mathbf{z}_i^t = \arg\max_{h_{ij} \neq h_i^*, \mathbf{z}_{ij}} S_{\mathbf{w}}(h_{ij}, \mathbf{z}_{ij})$
        $C_i = C_i \cup (\{h_i^t, z_i^t\} \cap (S_{\mathbf{w}}(h_{ij}, \mathbf{z}_{ij}) > S_{\mathbf{w}}(h_i^*, \mathbf{z}_i^*) - 1))$
    Solve QP:
        $\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + \sum_i \xi_i$
        s.t. $S_{\mathbf{w}}(h_i^*, \mathbf{z}_i^*) > S_{\mathbf{w}}(h, \mathbf{z}) + 1 - \xi_i \;\; \forall \{h, z\} \in C_i \qquad \forall i = 1 \ldots n$
until Convergence
```

4.4 Inference

In both standardized tests (reading comprehension and science question-answering), we need to search in an exponential space. Hence, we use a greedy procedure, namely beam search with a fixed beam size (5) for inference. That is, in each step, we only expand the five most promising substructure candidates so far given by the current score. We infer the snippet and alignments (for the reading comprehension task) and the textbook, chapter, section, snippet and alignments (for the science question answering task) one by one in this order.

Knowledge selection for science QA: For science QA, we select top 5 knowledge bits (triples, equivalences, etc.) from the knowledge resources that could be relevant for this question-answer during the inference procedure too. This is done heuristically by picking knowledge bits that explain parts of the hypothesis that are not explained by the chosen snippets.

4.5 Multi-task Latent Structured Learning

Standardized tests are usually complex, and often require us to interpret questions, the kind of answers they seek as well as the kinds of inference required to solve them. Many approaches in question answering (Moldovan et al., 2003; Ferrucci, 2012) solve this by having a top-level classifier that categorizes the complex task into a variety of sub-tasks. The sub-tasks can correspond to various categories of questions that can be asked or various facets of text understanding that are required to do well on the standardized test. It is well known that learning a sub-task together with other related sub-tasks leads to a better solution for each sub-task. Hence, we consider learning classifications of the sub-tasks and then using multi-task learning.

We extend our LSSVM to multi-task settings. Let S be the number of sub-tasks. We assume that the predictor $\mathbf{w}$ for each subtask s is partitioned into two parts: a parameter $\mathbf{w}_0$ that is globally shared across all subtasks and a parameter $\mathbf{v}_s$ that is locally used to account for the variations within the particular subtask: $\mathbf{w} = \mathbf{w}_0 + \mathbf{v}_s$. Mathematically we define the scoring function for the hypothesis set $\mathbf{h}_i$ of the sub-task s as:

$Score_{\mathbf{w}_0,\mathbf{v},s}(\mathbf{h}_i,\mathbf{z}) = (\mathbf{w}_0+\mathbf{v}_s)^T\phi(\mathbf{h}_i,\mathbf{z})$. The objective in this case can be written as:

$$\begin{aligned} \min_{\mathbf{w}_0,\mathbf{v}} \quad & \lambda_2\|\mathbf{w}_0\|^2 + \frac{\lambda_1}{S}\sum_{s=1}^{S}\|\mathbf{v}_s\|^2 + \\ & \sum_{s=1}^{S}\sum_{i=1}^{n}\left(\max_{h_{ij}\in\mathbf{h}_i,\mathbf{z}_{ij}\in\mathcal{Z}_i}\{(\mathbf{w}_0+\mathbf{v}_s)^T\phi(h_{ij},\mathbf{z}_{ij})+\Delta(\mathbf{h}_i^*,h_{ij})\} - C(\mathbf{w}_0+\mathbf{v}_s)^T\phi(h_i^*,\mathbf{z}_i^*)\right) \end{aligned} \tag{2}$$

Now, we extend a trick that Evgeniou and Pontil (2004) used on linear SVM to reformulate this problem into an objective that looks like (eq 1). Such reformulation will help in using algorithm 1 to solve the multi-task problem as well. Lets define a new feature map Φ_s, one for each sub-task s using the old feature map ϕ as:

$$\Phi_s(\mathbf{h}_i,\mathbf{z},\mathbf{y}) = (\frac{\phi(\mathbf{h}_i,\mathbf{z},\mathbf{y})}{\mu}, \underbrace{\mathbf{0},\ldots,\mathbf{0}}_{s-1}, \phi(\mathbf{h}_i,\mathbf{z},\mathbf{y}), \underbrace{\mathbf{0},\ldots,\mathbf{0}}_{S-s})$$

where $\mu = \frac{S\lambda_2}{\lambda_1}$ and the $\mathbf{0}$ denotes the zero vector of the same size as ϕ. Also define our new predictor as $\mathbf{w} = (\sqrt{\mu}\mathbf{w}_0, \mathbf{v}_1,\ldots,\mathbf{v}_S)$. Using this formulation we can show that $\mathbf{w}^T\Phi_s(\mathbf{h}_i,\mathbf{z}) = (\mathbf{w}_0+\mathbf{v}_s)^T\phi(\mathbf{h}_i,\mathbf{z})$ and $\|\mathbf{w}\|^2 = \sum_s\|\mathbf{v}_s\|^2 + \mu\|\mathbf{w}_0\|^2$. Hence, if we now define the objective (eq. 1) but use the new feature map and $\mathbf{w}$ then we will get back our multi-task objective (eq. 2). Thus, we can use the same setup as before for multi-task learning after appropriately changing the feature map. We will explore a few definitions of sub-tasks in our experiments.

4.6 Features

Recall that our features had the form $\psi(h,z)$ where the hypothesis h was itself formed from a question q and answer candidate a. Our feature vector $\psi(h,\mathbf{z})$ decomposes into a number of parts, where each part corresponds to a part of the answer-entailing structure. The answer-entailing structure for science question answering decomposes into five parts (namely textbook selection, chapter selection, section selection, snippet selection and alignment selection). Hence, the feature vector for the science question answering task decomposes into five parts. However, for the reading comprehension task, the text corresponding to the question is given. The answer-entailing structure for the reading comprehension test only has two parts (snippet selection and alignment selection). Hence, the feature vector for the reading comprehension task decomposes into two parts, and only the last two feature parts described below apply for the reading comprehension task.

For the textbook selection part, we index all the textbooks and score the top retrieved textbook by querying the hypothesis statement. We use tf-idf and BM25 scorers resulting in two features. Then, we find the jaccard similarity of bigrams and trigrams in the hypothesis and the textbook to get two more features for the first part. Similarly, for the chapter selection part we index all the textbook chapters and compute the tf-idf, BM25 and bigram, trigram features. For the section selection part we index all the sections and compute the same features.

The snippet selection part has features based on the text snippet part of the answer-entailing structure. Here we do a deeper linguistic analysis and include features for matching local neighborhoods in the snippet and the hypothesis: features for matching bigrams, trigrams, dependencies, semantic roles, predicate-argument structure as well as the global syntactic structure: a tree kernel for matching dependency parse trees of entire sentences (Srivastava & Hovy, 2013). If a text snippet contains the answer to the question, it should intuitively be similar to the question as well as to the answer. Hence, we also add features that are the element-wise product of features for the text-question match and text-answer match. In addition to features for the exact word/phrase match of the snippet and the hypothesis, we also add features using two paraphrase databases: ParaPara (Chan, Callison-Burch, & Van Durme, 2011) and DIRT (Lin & Pantel, 2001). These databases contain paraphrase rules of the form $\text{string}_1 \rightarrow \text{string}_2$. ParaPara has rules like "imprisoned" → "sent to jail", "huge amount of" → "large quantity of", etc. extracted through bilingual pivoting and DIRT database contains rules like "A is the author of B" → "A wrote B", "A caused B" → "B is triggered by A", etc. extracted using the distributional hypothesis (Harris, 1954). Whenever we have a substring in the text snippet that can be transformed into another using any of these two databases, we keep match features for the substring with a higher score (according to the current $\mathbf{w}$) and ignore the other substring. Finally, we also have features corresponding to the RST (Mann & Thompson, 1988) and coreference links to enable inference across sentences. RST tells us that sentences with discourse relations are related to each other and can help us answer certain kinds of questions (Jansen et al., 2014). For example, the "cause" relation between sentences in the text can often give cues that can help us answer "why" or "how" questions. Hence, we add additional features - conjunction of the rhetorical structure label from a RST parser and the question word - to our feature vector. Similarly, the entity and event co-reference relations allow us to reason about repeating entities or events. We replace an entity/event mention with their first mentions if that results into a greater score.

For the alignment part, we induce features based on word level similarity of aligned words: (a) surface-form match (Edit-distance), and (b) semantic word match (cosine similarity using SENNA word vectors (Collobert et al., 2011) and "Antonymy" 'Class-Inclusion' or 'Is-A' relations using Wordnet). Distributional vectors for multi-word expressions are obtained by adding the vector representations of comprising words (Mitchell & Lapata, 2008). To account for the hypothesized knowledge bits for the science question answering task, whenever we have the case that a word/multi-word expression in the hypothesis can be aligned to a word/multi-word expression in a hypothesized knowledge bit to produce a greater score, then we keep the features for the alignment with the knowledge bit instead.

4.7 Negation

We empirically found that one key limitation in our formulation is its inability to handle negation (both in questions and text). Negation is especially hurtful to our model as it not only results in poor performance on questions that require us to reason with negated facts, it provides our model with a wrong signal (facts usually align well with their negated versions). We use a simple heuristic to overcome the negation problem. We detect negation (either in the hypothesis or a sentence in the text snippet aligned with it) using a small set of manually defined rules that test for presence of words such as "not", "n't", etc. Then, we flip the partial order - i.e. the correct hypothesis is now ranked below the other competing hypotheses for this question. For inference at test time, we also invert the prediction rule i.e. we predict the hypothesis (answer) that has the least score under the model.

5 Experiments

5.1 Datasets

We use two datasets for our evaluation:

Reading Comprehension: First is the MCTest-500 dataset (Richardson et al., 2013)[5], a freely available set of 500 stories (split into 300 train, 50 dev and 150 test) and associated questions. The passages are fictional stories so the answers can be found only in the story itself. The stories and questions are carefully limited, thereby minimizing the world knowledge required for this task. Yet, the task is challenging for most modern NLP systems. Each passage in MCTest has four multiple choice questions, each with four answer

[5] http://research.microsoft.com/mct

choices. Each question has only one correct answer. Furthermore, questions are also annotated with 'single' and 'multiple' labels. The questions annotated 'single' only require one sentence in the story to answer them. For 'multiple' questions it should not be possible to find the answer to the question in any individual sentence of the passage. In a sense, the 'multiple' questions are harder than the 'single' questions as they typically require complex lexical analysis, some inference and some form of limited reasoning.

Science Question Answering: The second dataset is a freely available set of 8^{th} grade science questions released by the *Allen Institute of AI* as part of their *Aristo* project[6]. The dataset comprises of 2500 questions. Each question has 4 answer candidates, of which exactly one is correct. We used questions 1-1500 for training, questions 1500-2000 for development and questions 2000-2500 for testing. We also used publicly available 8^{th} grade science textbooks available through `http://www.ck12.org`. The science curriculum consists of seven textbooks on Physics, Chemistry, Biology, Earth Science and Life Science. Each textbook on an average has 18 chapters, and each chapter in turn is divided into 12 sections on an average. Each section, on an average, is followed by 3-4 multiple choice review questions (total 1369 review questions) to enhance student learning. These review problems have value as part of the answer-entailing structure (textbook, chapter and section) is known for these problems. We will use these to further boost our results. We also collected a number of domain specific science dictionaries[7], study guides[8], flash cards[9] and semi-structured tables (Simple English Wiktionary[10] and Aristo Tablestore[11]) available online and created triples and equivalences used as external knowledge.

5.2 Baselines

We have a number of baselines: (1) The first three baselines are inspired from Richardson et. al. (2013). The first baseline (called *SW*) uses a sliding window and matches a bag of words constructed from the question and hypothesized answer to the text. (2) Since *SW* ignores long range dependencies, the second baseline (called *SW+D*) accounts for intra-word distances as well. As far as we know, *SW+D* is the best previously published result on MCTest[12]. (3) The third baseline (called *RTE*) uses textual entailment to answer MCTest questions. For this baseline, MCTest is again re-casted as an RTE task by converting each question-answer pair into a statement (using Cucerzan et. al. (2005)) and then selecting the answer whose statement has the highest likelihood of being entailed by the story[13]. (4) The fourth baseline (called *LSTM*) is taken from Weston et. al. (2015). The baseline uses LSTMs (Hochreiter & Schmidhuber, 1997) to accomplish the task. LSTMs have recently achieved state-of-the-art results in a variety of tasks due to their ability to model long-term context information as opposed to other neural networks based techniques. (5) The fifth baseline (called *QANTA*)[14] is taken from Iiyer et. al. (2014). *QANTA* too uses a recursive neural network for question answering. (6) The sixth baseline (called *Jacana*) uses an off-the shelf aligner (Yao et al., 2013) to align sentences in the passage with the hypothesis. Then it selects the answer that produces the alignment with

[6] `https://www.kaggle.com/c/the-allen-ai-science-challenge/download/training\_set.tsv.zip`

[7] `http://www.harcourtschool.com/glossary/science/intro.html`, and `http://sci2.esa.int/glossary/`

[8] `http://www.depedbataan.com/resources/20/gr_8_teaching_guide_in_science.pdf`, and `http://www.mapleschools.com/docs/293_11_30_20078th\%20Grade\%20Science\%20Study\%20Guide\%201.pdf`

[9] `https://quizlet.com/`

[10] `https://simple.wiktionary.org/wiki/Main_Page`

[11] `http://allenai.org/content/data/AristoTablestore-Nov2015Snapshot.zip`

[12] We also construct two additional baselines (*LSTM* and *QANTA*) for comparison in this paper both of which achieve superior performance to *SW+D*.

[13] The BIUTEE system (Stern & Dagan, 2012) available under the Excitement Open Platform `http://hltfbk.github.io/Excitement-Open-Platform/` was used for recognizing textual entailment.

[14] `http://cs.umd.edu/~miyyer/qblearn/`

maximum score.

For the science question answering task, we have some additional baselines taken from Clark (2016): (7) The *Lucene* baseline scores each answer candidate a_i by searching for the combination of the question q and answer candidate a_i in a lucene-based search engine and returns the highest scoring answer candidate. (8) The *PMI* baseline similarly scores each answer candidate a_i by computing the point-wise mutual information to measure the strength of the association between parts of the question-answer candidate combine and parts of the CK12 curriculum. (9) Finally, to test if our science question answering approach indeed benefits from jointly learning the retrieval and the answer selection modules, our final baseline *Lucene+LSSVM Alignment* retrieves the top section by querying $q + a_i$ in *Lucene* and then learns the remaining answer-entailment structure (alignment part of the answer-entailing structure in Figure 2) using a LSSVM.

5.3 Incorporating partially known structures

Now, we describe how review questions can be incorporated for the task of science QA. As described earlier, modern textbooks often provide review problems at the end of each section. These review problems help students review and better understand the material. Inspired by this, we make use of these review problems along with the standardized test problems to improve our model. These review problems have value as part of the answer-entailing structure (textbook, chapter and section) is known for these problems. Hence, we now have two sets of questions $\mathcal{Q}_{kaggle}$ and $\mathcal{Q}_{review}$. For the hypotheses derived from the review questions, the answer entailing structure is partially known. In this case, we use the formulation (equation 1) except that the max over $\mathbf{z}$ for the review questions is only taken over the unknown part of the latent structure. Our results show that jointly training (JT) our model with these review problems leads to further improvements.

5.4 Task Classification for MultiTask Learning

Reading Comprehensions: We consider three alternative task classifications for our reading comprehension experiments. First, we look at question classification. We use a simple question classification based on the question word (what, why, what, etc.). We call this QClassification. Next, we also use a question/answer classification[15] from Li and Roth (2002). This classifies questions into different semantic classes based on the possible semantic types of the answers sought. We call this QAClassification. Finally, we also learn a classifier for the 20 tasks in the Machine Comprehension gamut described in Weston et. al. (2015). The classification algorithm (called TaskClassification) was built on the *bAbI* training set. It is essentially a Naive-Bayes classifier and uses only simple unigram and bigram features for the question and answer. The tasks typically correspond to different strategies when looking for an answer in the machine comprehension setting. In our experiments we will see that learning these strategies is better than learning the question answer classification which is in turn better than learning the question classification.

Science QA: We explore two simple question classification schemes. The first classification scheme classifies questions based on the question word (what, why, etc.). We call this *Qword* classification. The second scheme is based on the type of the question asked and classifies questions into three coarser categories: (a) questions without context, (b) questions with context and (c) negation questions. This classification is based on the observation that many questions lay down some context and then ask a science concept based on this context. However, other questions are framed without any context and directly ask for the science concept itself. Then there is a smaller, yet, important subset of questions that involve negation that also needs to be handled separately. Table 1 gives examples of this classification. We call this classification *Qtype* classification[16].

[15] http://cogcomp.cs.illinois.edu/Data/QA/QC/

[16] We wrote a set of question matching rules (similar to the rules used to convert question answer pairs to hypotheses) to achieve this classification

Question Category	Example
Without context	Which example describes a learned behavior in a dog?
With context	When athletes begin to exercise, their heart rates and respiration rates increase. At what level of organization does the human body coordinate these functions?
Negation Questions	A teacher builds a model of a hydrogen atom. A red golf ball is used for a proton, and a green golf ball is used for an electron. Which is not accurate concerning the model?

Table 1: Example questions for *Qtype* classification

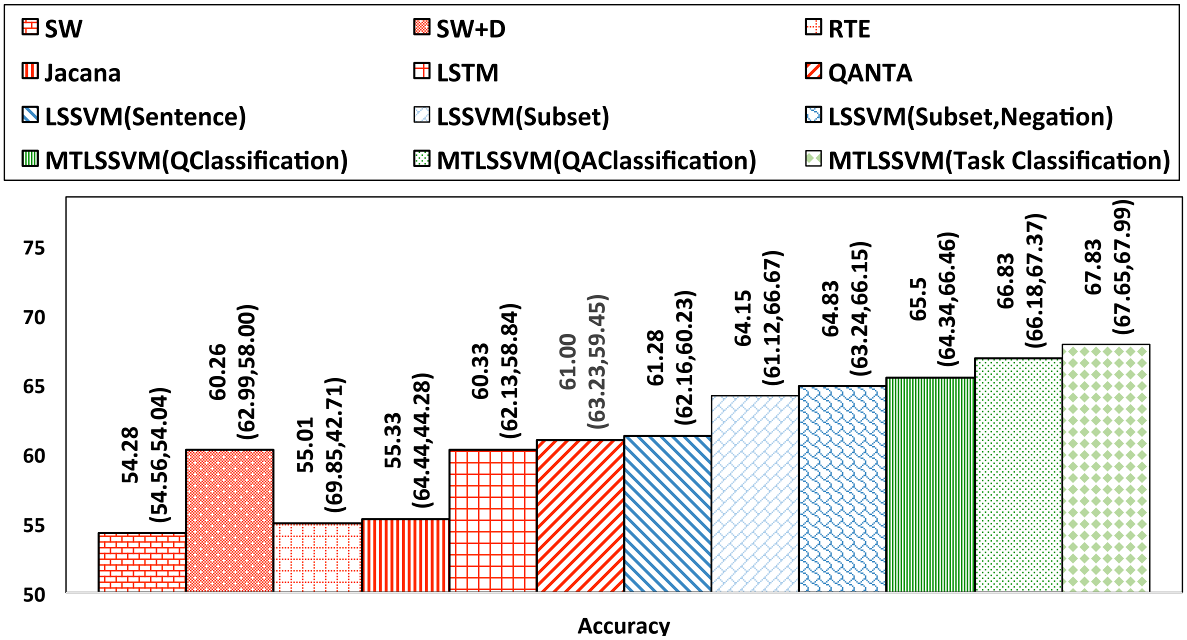

Figure 3: Variations of our method vs several baselines on the MCTest dataset. The labels corresponding to the result bars show test accuracy and the test accuracy on 'single' and 'multiple' sentences written in parentheses, respectively. Differences between the baselines and LSSVMs, the improvement due to negation, and the improvements due to multi-task learning are significant ($p < 0.05$) using the two-tailed paired T-test.

5.5 Results

We compare variants of our method[17] where we consider our modification for negation or not and multi-task LSSVMs.

Reading Comprehension: Figure 3 describes the comparison on *MCTest*. We can observe that all the LSSVM models have a better performance than all the five baselines (including LSTMs and RNNs which are state-of-the-art for many other NLP tasks) on both metrics. Very interestingly, LSSVMs have a considerable improvement over the baselines for "multiple" questions. We posit that this is because of our answer-entailing structure alignment strategy which is a weak proxy to the deep semantic inference procedure required for machine comprehension. The RTE baseline achieves the best performance on the "single" questions. This is perhaps because the RTE community has almost entirely focused on single sentence text hypothesis pairs for a long time. However, RTE fares pretty poorly on the "multiple" questions

[17] We tune the SVM regularization parameter C on the development set. We use Stanford CoreNLP, the HILDA parser (Feng & Hirst, 2014), and jMWE (Kulkarni & Finlayson, 2011) for linguistic preprocessing

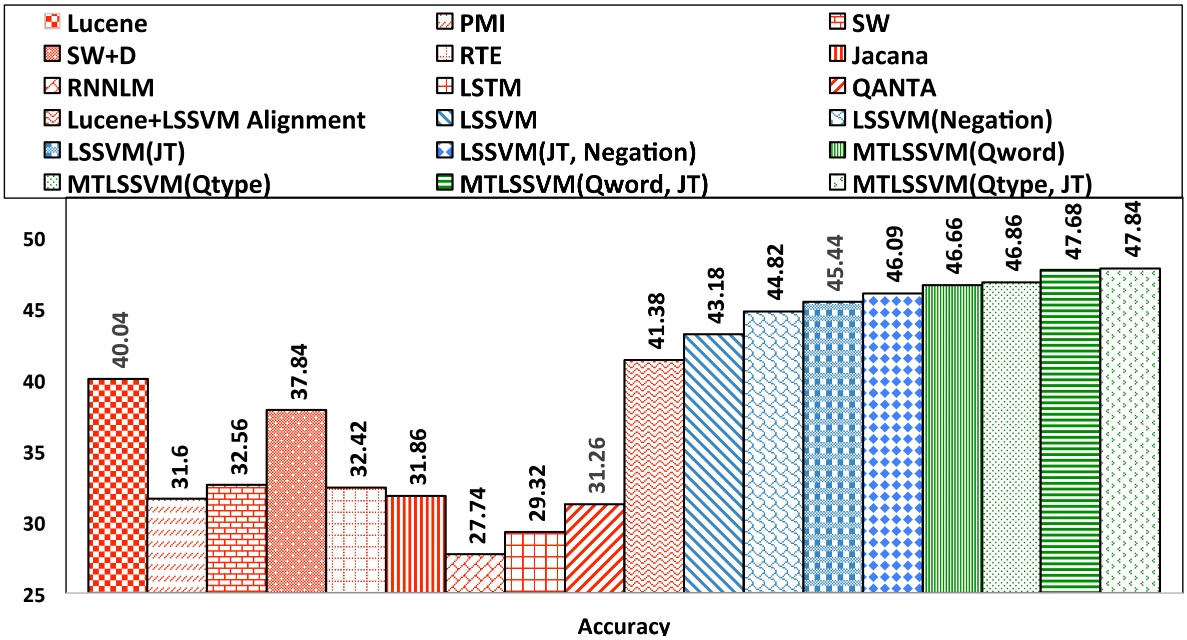

Figure 4: Variations of our method vs several baselines on the Science QA dataset. Differences between the baselines and LSSVMs, the improvement due to negation, the improvements due to multi-task learning and joint-learning are significant ($p < 0.05$) using the two-tailed paired T-test.

indicating that of-the-shelf RTE systems cannot perform inference across large texts.

Figure 3 also compares the performance of LSSVM variants when various answer-entailing structures are considered. Here we observe a clear benefit of using the alignment to the best subset structure over alignment to best sentence structure. We furthermore see improvements when the best subset alignment structure is augmented with the subset+ features. We can observe that the negation heuristic also helps, especially for "single" questions (majority of negation cases in the *MCTest* dataset are for the "single" questions).

It is also interesting to see that the multi-task learners show a substantial boost over the single task LSSVM. Also, it can be observed that the multi-task learner greatly benefits if we can learn a separation between the various strategies needed to learn an overarching list of subtasks required to solve the machine comprehension task. [18] The multi-task method (TaskClassification) which uses the Weston style categorization does better than the multi-task method (QAClassification) that learns the question answer classification. QAClassification in turn performs better than multi-task method (QClassification) that learns the question classification only.

Science QA: We consider both kinds of task classification strategies and joint training (JT).

Figure 4 shows the results. First, we can immediately observe that all the LSSVM models have a better performance than all the baselines. We also found an improvement when we handle negation using the heuristic described above[19]. MTLSSVMs showed a boost over single task LSSVM. *Qtype* classification scheme was found to work better than *Qword* classification which simply classifies questions based on the question word. The multi-task learner could benefit even more if we can learn a better separation between the various strategies needed to answer science questions. We found that joint training with review questions

[18] Note that this is despite the fact that the classifier in not learned on the *MCTest* dataset but the *bAbI* detaset! This hints at the fact that the task classification proposed in Weston et. al. (2015) is more general and broadly also makes sense for other machine comprehension settings such as *MCTest*.

[19] We found that the accuracy over test questions tagged by our heuristic as negation questions went up from 33.64 percent to 42.52 percent and the accuracy over test questions not tagged as negation did not decrease significantly

helped improve accuracy as well.

5.6 Strengths and Weaknesses of the Latent Structures:

A good question to be asked is how good is structure alignment as a proxy to the semantics of the problem? In this section, we attempt to tease out the strengths and limitations of such a structure alignment approach. To do so, we evaluate our methods on various reading comprehension tasks in the *bAbI* dataset.

The *bAbI* dataset is a synthetic dataset released under the *bAbI project*[20] (Weston et al., 2015). The dataset presents a set of 20 'tasks', each testing a different aspect of text understanding and reasoning in the QA setting, and hence can be used to test and compare capabilities of learning models in a fine-grained manner. For each 'task', 1000 questions are used for training and 1000 for testing. The 'tasks' refer to question categories such as questions requiring reasoning over single/two/three supporting facts or two/three arg. relations, yes/no questions, counting questions, etc. Candidate answers are not provided but the answers are typically constrained to a small set: either yes or no or entities already appearing in the text, etc. We write simple rules to convert the question and answer candidate pairs to hypotheses. For the *bAbI* dataset, we add additional features inspired from the "task" distinction to handle specific "tasks".

Table 2 shows the results of various LSSVMmodels on the *bAbI* datasets for each sub-task. In our experiments, we observed a similar general pattern of improvement of LSSVM over the baselines as well as the improvement due to multi-task learning. Again task classification helped the multi-task learner the most and the QA classification helped more than the QClassification. It is interesting here to look at the performance within the sub-tasks. Negation improved the performance for three sub-tasks, namely, the tasks of modelling "yes/no questions", "simple negations" and "indefinite knowledge" (the "Indefinite Knowledge" sub-task tests the ability to model statements that describe possibilities rather than certainties). Each of these sub-tasks contain a significant number of negation cases. Our models do especially well on questions requiring reasoning over one and two supporting facts, two arg. relations, indefinite knowledge, basic and compound coreference and conjunction. Our models achieve lower accuracy better than the baselines on two sub-tasks, namely "path finding" and "agent motivations". Our model along with the baselines do not do too well on the "counting" sub-task, although we get slightly better scores. The "counting" sub-task (which asks about the number of objects with a certain property) requires the inference to have an ability to perform simple counting operations. The "path finding" sub-task requires the inference to reason about the spatial path between locations (e.g. Pittsburgh is located on the west of New York). The "agents motivations" sub-task asks questions such as 'why an agent performs a certain action'. As inference is cheaply modelled via alignment structures, we lack the ability to deeply reason about facts or numbers. This is an important challenge for future work.

5.7 Feature Ablation

As described before, our feature set for the science QA experiment comprises of five parts, where each part corresponds to a part of the answer-entailing structure – textbook ($\mathbf{z}_1$), chapter ($\mathbf{z}_2$), section ($\mathbf{z}_3$), snippets ($\mathbf{z}_4$), and alignment ($\mathbf{z}_5$). It is interesting to know the relative importance of these parts in our model. Hence, we perform feature ablation on our best performing model - *MTLSSVM(QWord, JT)* where we remove the five feature parts one by one and measure the loss in accuracy. Figure 5 shows that the choice of section and alignment are important components of our model. Yet, all components are important and removing any of them will result in a loss of accuracy. Finally, in order to understand the value of external knowledge resources (K), we removed the component that induces and aligns the hypothesis with knowledge bits. This results in significant loss in performance, estabishing the efficacy of adding in external knowledge via our approach.

[20] `https://research.facebook.com/researchers/1543934539189348`

	Baslines				LSSVM					
								MultiTask		
Tasks	**SW**	**RTE**	**LSTM**	**QANTA**	**Sentence**	**Subset**	**Subset+Negation**	**QClassification**	**QAClassification**	**TaskClassification**
Single Supporting Fact	36	98	50	89	100	100	100	100	100	100
Two Supporting Facts	2	79	20	69	60	92	91	93	93	94
Three Supporting Facts	7	46	20	42	52	86	84	86	87	88
Two Arg. Relations	50	54	61	68	89	91	90	92	93	93
Three Arg. Relations	20	31	70	63	84	89	88	91	90	91
Yes/No Questions	49	48	48	54	58	58	78	81	84	85
Counting	52	11	49	55	61	63	61	65	64	64
Lists/Sets	42	34	45	47	55	73	71	77	80	82
Simple Negation	62	56	64	72	63	64	76	79	80	81
Indefinite Knowledge	45	43	44	68	74	78	87	88	91	92
Basic Coreference	25	31	72	80	91	96	96	97	97	98
Conjunction	9	59	74	86	94	91	90	95	96	97
Compound Coreference	26	72	94	95	86	89	88	93	93	94
Time Reasoning	19	68	27	43	65	70	68	71	74	76
Basic Deduction	20	49	21	72	76	78	76	80	81	82
Basic Induction	43	53	23	55	57	61	58	61	63	64
Positional Reasoning	46	66	51	55	81	88	88	90	91	90
Size Reasoning	52	77	52	63	78	84	83	85	87	89
Path Finding	0	11	8	45	9	9	9	11	11	11
Agents Motivations	76	91	91	93	66	70	68	69	69	70
Mean Performance	34	54	49	66	70	77	78	79	81	82

Table 2: Comparison of accuracies on the variations of our method against several baselines on 20 Tasks of the bAbI dataset. All integer differences are significant ($p < 0.05$) using the two-tailed paired T-test.

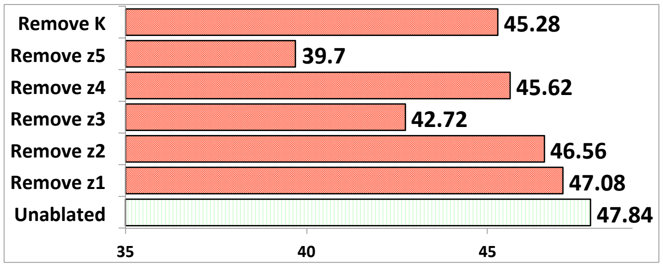

Figure 5: Feature ablation on the *MTLSSVM(Qword, JT) model*. We remove feature parts one by one and see what impact does each feature part removal has on the accuracy of the model.

6 Conclusion

In this paper, we proposed a solution for standardised tests (reading comprehensions and science question answering) which test a system's ability to understand language through a series of multiple choice question answering tasks. We posed the tests as an extension to RTE and developed a technique that learns latent alignment structures between given texts (passage, textbooks, etc.) and the hypotheses in the equivalent RTE setting. These tests require solving a variety of sub-tasks so we extended our technique to a multi-task setting. Our technique showed empirical improvements over various IR and neural network baselines. The latent structures while effective are cheap proxies to the reasoning and language understanding required for this task and have their own limitations. In the future, we plan to explore approaches to perform structured inference over richer semantic representations.

References

Arai, N. H., & Matsuzaki, T. (2014). The impact of ai on education–can a robot get into the university of tokyo?. In *Proc. ICCE*, pp. 1034–1042.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., & Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48. ACM.

Berant, J., Chou, A., Frostig, R., & Liang, P. (2013). Semantic parsing on freebase from question-answer pairs.. In *EMNLP*, pp. 1533–1544. ACL.

Blunsom, P., & Cohn, T. (2006). Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 65–72. Association for Computational Linguistics.

Burges, C. J. (2013). Towards the machine comprehension of text: An essay. Tech. rep., Microsoft Research Technical Report MSR-TR-2013-125, 2013, pdf.

Chan, T. P., Callison-Burch, C., & Van Durme, B. (2011). Reranking bilingually extracted paraphrases using monolingual distributional similarity. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pp. 33–42.

Clark, P. (2015). Elementary School Science and Math Tests as a Driver for AI:Take the Aristo Challenge!. In *Proceedings of IAAI*.

Clark, P., & Etzioni, O. (2016). My computer is an honor student - but how intelligent is it? standardized tests as a measure of ai. In *Proceedings of AI Magazine*.

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

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, *12*, 2493–2537.

Cucerzan, S., & Agichtein, E. (2005). Factoid question answering over unstructured and structured content on the web. In *Proceedings of TREC 2005*.

Evgeniou, T., & Pontil, M. (2004). Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 109–117.

Feng, V. W., & Hirst, G. (2014). A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 511–521.

Ferrucci, D. A. (2012). Introduction to this is watson. *IBM Journal of Research and Development*, *56*(3.4), 1–1.

Fujita, A., Kameda, A., Kawazoe, A., & Miyao, Y. (2014). Overview of todai robot project and evaluation framework of its nlp-based problem solving. *World History*, *36*, 36.

Harris, Z. (1954). Distributional structure. *Word*, *10*(23), 146–162.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Iyyer, M., Boyd-Graber, J., Claudino, L., Socher, R., & Daumé III, H. (2014). A neural network for factoid question answering over paragraphs. In *Empirical Methods in Natural Language Processing*.

Jansen, P., Surdeanu, M., & Clark, P. (2014). Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 977–986.

Khot, T., Balasubramanian, N., Gribkoff, E., Sabharwal, A., Clark, P., & Etzioni, O. (2015). Markov logic networks for natural language question answering. *arXiv preprint arXiv:1507.03045*.

Kulkarni, N., & Finlayson, M. A. (2011). jmwe: A java toolkit for detecting multi-word expressions. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pp. 122–124. Association for Computational Linguistics.

Kushman, N., Artzi, Y., Zettlemoyer, L., & Barzilay, R. (2014). Learning to automatically solve algebra word problems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Li, X., & Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1–7.

Li, Y., & Clark, P. (2015). Answering elementary science questions by constructing coherent scenes using background knowledge. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Lin, D., & Pantel, P. (2001). Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 323–328.

MacCartney, B., Galley, M., & Manning, C. D. (2008). A phrase-based alignment model for natural language inference. In *Proceedings of the conference on empirical methods in natural language processing*, pp. 802–811.

Mann, W. C., & Thompson, S. A. (1988). {Rhetorical Structure Theory: Toward a functional theory of text organisation}. *Text*, *3*(8), 234–281.

Mitchell, J., & Lapata, M. (2008). Vector-based models of semantic composition. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pp. 236–244.

Moldovan, D., Paşca, M., Harabagiu, S., & Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, *21*(2), 133–154.

Richardson, M., Burges, J. C., & Renshaw, E. (2013). Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 193–203.

Sachan, M., Dubey, A., Xing, E. P., & Richardson, M. (2015). Learning answer-entailing structures for machine comprehension. In *Proceedings of ACL*.

Sachan, M., Dubey, K., & Xing, E. P. (2016). Science question answering using instructional materials. In *Proceedings of ACL*.

Sachan, M., & Xing, E. P. (2016a). Easy questions first? A case study on curriculum learning for question answering. In *Proceedings of ACL*.

Sachan, M., & Xing, E. P. (2016b). Machine comprehension using rich semantic representations. In *Proceedings of ACL*.

Sammons, M., Vydiswaran, V., Vieira, T., Johri, N., Chang, M., Goldwasser, D., Srikumar, V., Kundu, G., Tu, Y., Small, K., Rule, J., Do, Q., & Roth, D. (2009). Relation alignment for textual entailment recognition. In *TAC*.

Seo, M. J., Hajishirzi, H., Farhadi, A., & Etzioni, O. (2014). Diagram understanding in geometry questions. In *Proceedings of AAAI*.

Seo, M. J., Hajishirzi, H., Farhadi, A., Etzioni, O., & Malcolm, C. (2015). Solving geometry problems: combining text and diagram interpretation. In *Proceedings of EMNLP*.

Srivastava, S., & Hovy, D. (2013). A walk-based semantically enriched tree kernel over distributed word representations. In *Empirical Methods in Natural Language Processing*, pp. 1411–1416.

Stern, A., & Dagan, I. (2012). Biutee: A modular open-source system for recognizing textual entailment. In *Proceedings of the ACL 2012 System Demonstrations*, pp. 73–78.

Sultan, A. M., Bethard, S., & Sumner, T. (2014). Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association of Computational Linguistics – Volume 2, Issue 1*, 219–230.

Weston, J., Bordes, A., Chopra, S., & Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Weston, J., Chopra, S., & Bordes, A. (2014). Memory networks. *arXiv preprint arXiv:1410.3916*.

Yao, X., Durme, B. V., Callison-Burch, C., & Clark, P. (2013). A lightweight and high performance monolingual word aligner.. In *ACL (2)*, pp. 702–707.

Yao, X., & Van Durme, B. (2014). Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 956–966. Association for Computational Linguistics.

Yao, X., Van Durme, B., Callison-Burch, C., & Clark, P. (2013). Semi-markov phrase-based monolingual alignment. In *Proceedings of EMNLP*.

Yih, W., Chang, M.-W., Meek, C., & Pastusiak, A. (2013). Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Yu, C.-N., & Joachims, T. (2009). Learning structural svms with latent variables. In *International Conference on Machine Learning (ICML)*.

Yuille, A. L., & Rangarajan, A. (2003). The concave-convex procedure. *Neural Comput.*

Zhang, D., & Lee, W. S. (2003). Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 26–32. ACM.

Zhu, J., Chen, N., & Xing, E. P. (2011). Infinite latent svm for classification and multi-task learning. In *Advances in neural information processing systems*, pp. 1620–1628.