Learning Answer-Entailing Structures for Machine Comprehension

Mrinmaya Sachan[1*] **Avinava Dubey**[1*] **Eric P. Xing**[1] **Matthew Richardson**[2]
[1]Carnegie Mellon University [2]Microsoft Research
[1]{mrinmays, akdubey, epxing}@cs.cmu.edu [2]mattri@microsoft.com

Abstract

Understanding open-domain text is one of the primary challenges in NLP. Machine comprehension evaluates the system's ability to understand text through a series of question-answering tasks on short pieces of text such that the correct answer can be found only in the given text. For this task, we posit that there is a hidden (latent) structure that explains the relation between the question, correct answer, and text. 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 machine comprehension questions on novel texts. We extend this framework to incorporate multi-task learning on the different subtasks that are required to perform machine comprehension. Evaluation on a publicly available dataset shows that our framework outperforms various IR and neural-network baselines, achieving an overall accuracy of 67.8% (vs. 59.9%, the best previously-published result.)

1 Introduction

Developing an ability to understand natural language is a long-standing goal in NLP and holds the promise of revolutionizing the way in which people interact with machines and retrieve information (e.g., for scientific endeavor). To evaluate this ability, Richardson et al. (2013) proposed the task of machine comprehension (MCTest), along with a dataset for evaluation. Machine comprehension evaluates a machine's understanding by posing a series of reading comprehension questions and associated texts, where the answer to each question can be found only in its associated text. Solutions typically focus on some semantic interpretation of the text, possibly with some form of probabilistic or logical inference, in order to answer the questions. Despite significant recent interest (Burges, 2013; Weston et al., 2014; Weston et al., 2015), the problem remains unsolved.

In this paper, we propose an approach for machine comprehension. Our approach learns latent *answer-entailing structures* that can help us answer questions about a text. The answer-entailing structures in our model are closely related to the inference procedure often used in various models for MT (Blunsom and Cohn, 2006), RTE (MacCartney et al., 2008), paraphrase (Yao et al., 2013b), QA (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 text that are required to answer the question. An example of such an answer-entailing structure is given in Figure 1. The key difference between the answer-entailing structures considered here and the alignment structures considered in previous works is that we can align multiple sentences in the text to the hypothesis. The sentences in the text considered for alignment are not restricted to occur contiguously in the text. To allow such a discontiguous alignment, we make use of the document structure; in particular, we take help from rhetorical structure theory (Mann and Thompson, 1988) and event and entity coreference links across sentences. 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 bene-

*Work started while the first two authors were interns at Microsoft Research, Redmond.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 239–249, Beijing, China, July 26-31, 2015.

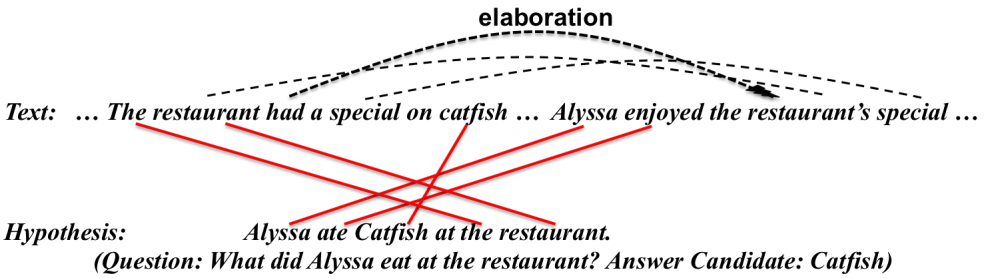

Figure 1: The *answer-entailing structure* for an example from MCTest500 dataset. The question and answer candidate are combined to generate a hypothesis sentence. Then latent alignments are found between the hypothesis and the appropriate snippets in the text. The solid red lines show the word alignments 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.

ficial as they can assist a human in verifying the correctness of the answer, eliminating the need to read a lengthy document.

The overall model is trained in a max-margin fashion using a latent structural SVM (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 and Roth, 2002; Zhang and Lee, 2003), which typically divides the questions into semantic categories based on the type of the question or answers expected. This helps the system impose some constraints on the plausible answers. Machine comprehension 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. Recently, Weston et al. (2015) defined a set of 20 sub-tasks in the machine comprehension setting, each referring to a specific aspect of language understanding and reasoning required to build a machine comprehension system. They include fact chaining, negation, temporal and spatial reasoning, simple induction, deduction and many more. We use this set to learn to classify questions into the various machine comprehension subtasks, and show that this task classification further improves our performance on MCTest. 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 provide experimental validation for our model on a real-world dataset (Richardson et al., 2013) and achieve superior performance vs. a number of IR and neural network baselines.

2 The Problem

Machine comprehension requires us to answer questions based on unstructured text. We treat this as selecting the best answer from a set of candidate answers. The candidate answers may be pre-defined, as is the case in multiple-choice question answering, or may be undefined but restricted (e.g., to yes, no, or any noun phrase in the text).

Machine Comprehension as Textual Entailment: Let for each question $q_i \in Q$, $\mathbf{t}_i$ be the unstructured text and $A_i = \{a_{i1}, \ldots, a_{im}\}$ be the set of candidate answers to the question. We cast the machine comprehension task as a textual entailment task 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 the question matching/rewriting rules described in Cucerzan and Agichtein (2005) to achieve this transformation. For each question q_i, the machine comprehension 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.

3 Latent Answer-Entailing Structures

The latent answer-entailing structures help the model in providing evidence for the correct hy-

pothesis. We consider the quality of a one-to-one word alignment from a hypothesis to snippets in the text as a proxy for the evidence. Hypothesis words are aligned to a unique text word in the text or an empty word. For example, in 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 structures 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. We explore three variants of the snippets in the text to be aligned to the hypothesis. The choice of these snippets composed with the word alignment is the resulting hidden structure called an answer-entailing structure.

1. Sentence Alignment: 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 Alignment: Here we find a subset of sentences from the text (instead of just one sentence) that best aligns with the hypothesis.

3. Subset+ Alignment: This is the same as above except that the best subset is an ordered set.

4 Method

A natural solution is to treat MCTest as a structured prediction problem of ranking the hypotheses $\mathbf{h}_i$ such that the correct hypothesis is at the top of this ranking. This induces a constraint on the ranking structure that the correct hypothesis is ranked above the other competing hypotheses. For each text $\mathbf{t}_i$ and hypotheses set $\mathbf{h}_i$, let $\mathcal{Y}_i$ be the set of possible orderings of the hypotheses. Let $\mathbf{y}_i^* \in \mathcal{Y}_i$ be a correct ranking (such that the correct hypothesis is at the top of this ranking). Let the set of possible answer-entailing structures for each text hypothesis pair $(\mathbf{t}_i, \mathbf{h}_i)$ be denoted by $\mathcal{Z}_i$. For each text $\mathbf{t}_i$, with hypotheses set $\mathbf{h}_i$, an ordering of the hypotheses $\mathbf{y} \in \mathcal{Y}_i$, and hidden structure $\mathbf{z} \in \mathcal{Z}_i$, we define a scoring function $Score_{\mathbf{w}}(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y})$ parameterized by a weight vector $\mathbf{w}$ such that we have the prediction rule: $(\widehat{\mathbf{y}}_i, \widehat{\mathbf{z}}_i) = \arg\max_{\mathbf{y}\in\mathcal{Y}_i, \mathbf{z}\in\mathcal{Z}_i} Score_{\mathbf{w}}(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y})$. The learning task is to find $\mathbf{w}$ such that the predicted ordering $\widehat{\mathbf{y}}_i$ is close to the optimal ordering $\mathbf{y}_i^*$. Mathematically this can be written as $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \Delta(\mathbf{y}_i^*, \mathbf{z}_i^*, \widehat{\mathbf{y}}_i, \widehat{\mathbf{z}}_i)$ where $\mathbf{z}_i^* = \arg\max_{\mathbf{z}\in\mathcal{Z}_i} Score_{\mathbf{w}}(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}_i^*)$ and Δ is the loss function between the predicted and the actual ranking and latent structure. We simplify the loss function and assume it to be independent of the hidden structure $(\Delta(\mathbf{y}_i^*, \mathbf{z}_i^*, \widehat{\mathbf{y}}_i, \widehat{\mathbf{z}}_i) = \Delta(\mathbf{y}_i^*, \widehat{\mathbf{y}}_i))$ and use a linear scoring function: $Score_{\mathbf{w}}(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) = \mathbf{w}^T\phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y})$ where ϕ is a feature map dependent on the text $\mathbf{t}_i$, the hypothesis set $\mathbf{h}_i$, an ordering of answers $\mathbf{y}$ and a hidden structure $\mathbf{z}$. We use a convex upper bound of the loss function (Yu and Joachims, 2009) to rewrite the objective:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 - C\sum_i \mathbf{w}^T\phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}_i^*, \mathbf{y}_i^*) \quad (1)$$
$$+C\sum_{i=1}^{n} \max_{\mathbf{y}\in\mathcal{Y}_i, \mathbf{z}\in\mathcal{Z}_i} \{\mathbf{w}^T\phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) + \Delta(\mathbf{y}_i^*, \mathbf{y})\}$$

This problem can be solved using Concave-Convex Programming (Yuille and Rangarajan, 2003) with the cutting plane algorithm for structural SVM (Finley and Joachims, 2008). We use phi partial order (Joachims, 2006; Dubey et al., 2009) which has been used in previous structural ranking literature to incorporate ranking structure in the feature vector ϕ:

$$\phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) = \sum_{j:h_{ij}\neq h_i^*} c_j(\mathbf{y})(\psi(\mathbf{t}_i, h_i^*, z_i^*) - \psi(\mathbf{t}_i, h_{ij}, z_j)) \quad (2)$$

where, $c_j(\mathbf{y}) = 1$ if h_i^* is above $\mathbf{h}_{ij}$ in the ranking $\mathbf{y}$ else -1. We use pair preference (Chakrabarti et al., 2008) as the ranking loss $\Delta(\mathbf{y}_i^*, \mathbf{y})$. Here, ψ is the feature vector defined for a text, hypothesis and answer-entailing structure.

Solution: We substitute the feature map definition (2) into Equation 1, leading to our LSSVM formulation. We consider the optimization as an alternating minimization problem where we alternate between getting the best z_{ij} and ψ for each text-hypothesis pair given $\mathbf{w}$ (inference) and then solving for the weights $\mathbf{w}$ given ψ to obtain an optimal ordering of the hypothesis (learning). The step for solving for the weights is similar to rankSVM

(Joachims, 2002). Algorithm 1 describes our overall procedure Here, we use beam search for inferring the latent structure z_{ij} in step 3. Also, note that in step 3, when the answer-entailing structures are "Subset" or "Subset+", we can always get a higher score by considering a larger subset of sentences. To discourage this, we add a penalty on the score proportional to the size of the subset.

Algorithm 1 Alternate Minimization for LSSVM

1: Initialize $\mathbf{w}$
2: **repeat**
3: $z_{ij} = \arg\max_z \mathbf{w}^T \psi(\mathbf{t}_i, h_{ij}, z)\ \forall i, j$
4: Compute ψ for each i, j
5: $\mathcal{C}_i = \emptyset\ \forall i$
6: **repeat**
7: **for** $i = 1, \ldots, n$ **do**
8: $r(\mathbf{y}) = \mathbf{w}^T \phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) + \Delta(\mathbf{y}_i^*, \mathbf{y}) - \mathbf{w}^T \phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}_i^*, \mathbf{y}_i^*)$
9: $\widehat{\mathbf{y}}_i = \arg\max_{\mathbf{y} \in \mathcal{Y}_i} r(\mathbf{y})$
10: $\xi_i = \max\{0, \max_{\mathbf{y} \in \mathcal{U}_i} r(\mathbf{y})\}$
11: **if** $r(\widehat{\mathbf{y}}_i) > \xi_i + \epsilon$ **then**
12: $\mathcal{C}_i = \mathcal{C}_i \cup \widehat{\mathbf{y}}_i$

$$Solve: \quad \min_{\mathbf{w}, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_i \xi_i$$

$$\forall i, \forall \mathbf{y} \in \mathcal{C}_i : \mathbf{w}^T \phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}_i^*, \mathbf{y}_i^*) \geq \mathbf{w}^T \phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) + \Delta(\mathbf{y}_i^*, \mathbf{y}) - \xi_i$$

13: **until** no change in any $\mathcal{C}_i$
14: **until** Convergence

Multi-task Latent Structured Learning: Machine comprehension is a complex task which often requires us to interpret questions, the kind of answers they seek as well as the kinds of inference required to solve them. Many approaches in QA (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 at machine comprehension in its entirety.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 each subtasks and a parameter $\mathbf{v}_s$ that is locally used to provide for the variations within the particular subtask: $\mathbf{w} = \mathbf{w}_0 + \mathbf{v}_s$. Mathematically we define the scoring function for text $\mathbf{t}_i$, hypothesis set $\mathbf{h}_i$ of the subtask s to be $Score_{\mathbf{w}_0, \mathbf{v}, s}(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) = (\mathbf{w}_0 + \mathbf{v}_s)^T \phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y})$. The objective in this case is

$$\min_{\mathbf{w}_0, \mathbf{v}} \lambda_2 \|\mathbf{w}_0\|^2 + \frac{\lambda_1}{S} \sum_{s=1}^{S} \|\mathbf{v}_s\|^2 \tag{3}$$
$$\sum_{s=1}^{S} \sum_{i=1}^{n} \max_{\mathbf{y} \in \mathcal{Y}_i, \mathbf{z} \in \mathcal{Z}_i} \{(\mathbf{w}_0 + \mathbf{v}_s)^T \phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) + \Delta(\mathbf{y}_i^*, \mathbf{y})\} - C \sum_i (\mathbf{w}_0 + \mathbf{v}_s)^T \phi(\mathbf{t}, \mathbf{h}_i, \mathbf{z}_i^*, \mathbf{y}_i^*)$$

Now, we extend a trick that Evgeniou and Pontil (2004) used on linear SVM to reformulate this problem into an objective that looks like (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:

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

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 $\mathbf{w}^T \Phi_s(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y}) = (\mathbf{w}_0 + \mathbf{v}_s)^T \phi(\mathbf{t}_i, \mathbf{h}_i, \mathbf{z}, \mathbf{y})$ and $\|\mathbf{w}\|^2 = \sum_s \|\mathbf{v}_s\|^2 + \mu\|\mathbf{w}_0\|^2$. Hence, if we now define the objective (1) but use the new feature map and $\mathbf{w}$ then we will get back our multi-task objective (3). 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.

Features: Recall that our features had the form $\psi(\mathbf{t}, h, z)$ where the hypothesis h was itself formed from a question q and answer candidate a. Given an answer-entailing structure z, we induce the following features based on word level similarity of aligned words: (a) Limited word-level surface-form matching and (b) Semantic word form matching: Word similarity for synonymy using SENNA word vectors (Collobert et al., 2011),

"Antonymy" 'Class-Inclusion' or 'Is-A' relations using Wordnet (Fellbaum, 1998). We compute additional features of the aforementioned kinds to match named entities and events. We also add features for matching local neighborhood in the aligned structure: features for matching bigrams, trigrams, dependencies, semantic roles, predicate-argument structure as well as features for matching global structure: a tree kernel for matching syntactic representations of entire sentences using Srivastava and Hovy (2013). The local and global features can use the RST and coreference links enabling inference across sentences. For instance, in the example shown in figure 1, the coreference link connecting the two "restaurant" words brings the snippets "Alyssa enjoyed the" and "had a special on catfish" closer making these features more effective. The answer-entailing structures should be intuitively similar to the question but also the answer. Hence, we add features that are the product of features for the text-question match and text-answer match.

String edit Features: In addition to looking for features on exact word/phrase match, we also add features using two paraphrase databases ParaPara (Chan et al., 2011) and DIRT (Lin and Pantel, 2001). The ParaPara database contains strings of the form $\text{string}_1 \rightarrow \text{string}_2$ like "total lack of" $\rightarrow$ "lack of", "is one of" $\rightarrow$ "among", etc. Similarly, the DIRT database contains paraphrases of the form "If **X** decreases **Y** then **X** reduces **Y**", "If **X** causes **Y** then **X** affects **Y**", etc. Whenever we have a substring in the text can be transformed into another using these two databases, we keep match features for the substring with a higher score (according to $\mathbf{w}$) and ignore the other substring.

The sentences with discourse relations are related to each other by means of substitution, ellipsis, conjunction and lexical cohesion, etc (Mann and Thompson, 1988) and can help us answer certain kinds of questions (Jansen et al., 2014). As an 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 RST label and the question word - to our feature vector. Similarly, the entity and event co-reference relations can allows the system to reason about repeating entities or events through all the sentences they get mentioned in. Thus, we add additional features of the aforementioned types by replacing entity mentions with their first mentions.

Subset+ Features: We add an additional set of features which match the first sentence in the ordered set to the question and the last sentence in the ordered set to the answer. This helps in the case when a certain portion of the text is targeted by the question but then it must be used in combination with another sentence to answer the question. For instance, in Figure 1, sentence 2 mentions the target of the question but the answer can only be given when in combination with sentence 1.

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 to 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 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

Datasets: We use two datasets for our evaluation. **(1)** First is the MCTest-500 dataset [1], a freely available set of 500 stories (split into 300 train, 50 dev and 150 test) and associated questions (Richardson et al., 2013). The stories are fictional 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 story in MCTest has four multiple choice questions, each with four answer 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

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

harder than the 'single' questions as they typically require complex lexical analysis, some inference and some form of limited reasoning. Cucerzan-converted questions can also be downloaded from the MCTest website.

(2) The second dataset is a synthetic dataset released under the *bAbI project*[2] (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. [3]

Baselines: We have five 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 this 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 this task.[4] (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 and Agichtein (2005)) and then selecting the answer whose statement has the highest likelihood of being entailed by the story. [5] (4) The fourth baseline (called *LSTM*) is taken from Weston et al. (2015). The baseline uses LSTMs (Hochreiter and 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*)[6] is taken from Iyyer et al. (2014). *QANTA* too uses a recursive neural network for question answering.

Task Classification for MultiTask Learning: We consider three alternative task classifications for our 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[7] 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.

Results: We compare multiple variants of our LSSVM[8] where we consider a variety of answer-entailing structures and our modification for negation and multi-task LSSVM, where we consider three kinds of task classification strategies against the baselines on the *MCTest* dataset. We consider two evaluation metrics: accuracy (proportion of questions correctly answered) and $NDCG_4$

[2]https://research.facebook.com/researchers/1543934539189348

[3]Note that the *bAbI* dataset is artificial and not meant for open-domain machine comprehension. It is a toy dataset generated from a simulated world. Due to its restrictive nature, we do not use it directly in evaluating our method vs. other open-domain machine comprehension methods. However, it provides benefit in identifying interesting subtasks of machine comprehension. As will be seen, we are able to leverage the dataset both to improve our multi-task learning algorithm, as well as to analyze the strengths and weaknesses of our model.

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

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

[6]http://cs.umd.edu/ miyyer/qblearn/

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

[8]We tune the SVM regularization parameter C and the penalty factor on the subset size on the development set. We use a beam of size 5 in our experiments. We use Stanford CoreNLP and the HILDA parser (Feng and Hirst, 2014) for linguistic preprocessing.

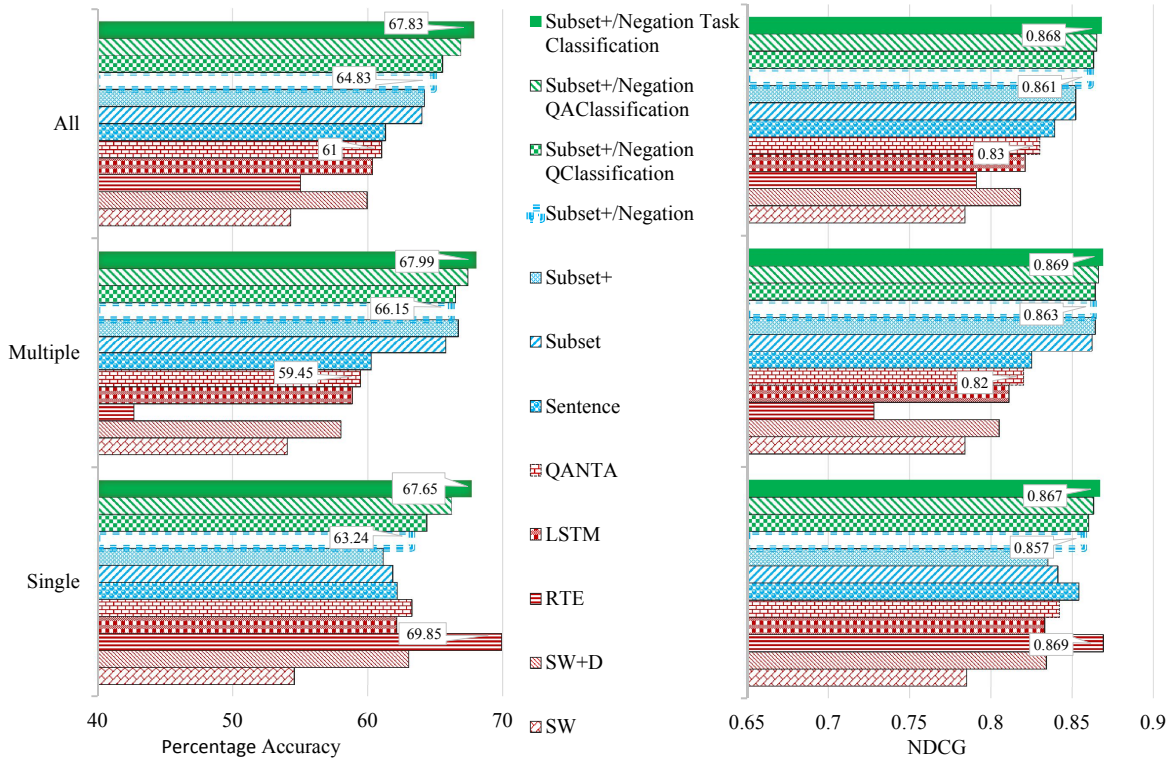

Figure 2: Comparison of variations of our method against several baselines on the MCTest-500 dataset. The figure shows two statistics, accuracy (on the left) and NDCG_4 (on the right) on the test set of MCTest-500. All differences between the baselines and LSSVMs, the improvement due to negation and the improvements due to multi-task learning are significant ($p < 0.01$) using the two-tailed paired T-test. The exact numbers are available in the supplementary.

(Järvelin and Kekäläinen, 2002). Unlike classification accuracy which evaluates if the prediction is correct or not, NDCG_4, being a measure of ranking quality, evaluates the position of the correct answer in our predicted ranking.

Figure 2 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 indicating that of-the-shelf RTE systems cannot perform inference across large texts.

Figure 2 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 SSVM. 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. [9] The multi-task method (TaskClassification) which uses the Weston style categorization does better

[9]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*.

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.

6 Strengths and Weaknesses

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 for machine comprehension. To do so, we evaluate our methods on various tasks in the *bAbI* dataset.For the *bAbI* dataset, we add additional features inspired from the "task" distinction to handle specific "tasks".

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 structure, we lack the ability to deeply reason about facts or numbers. This is an important challenge for future work.

7 Related Work

The field of QA is quite rich. Most QA evaluations such as TREC have typically focused on short factoid questions. The solutions proposed have ranged from various IR based approaches (Mittal and Mittal, 2011) that treat this as a problem of retrieval from existing knowledge bases and perform some shallow inference to NLP approaches that learn a similarity between the question and a set of candidate answers (Yih et al., 2013). A majority of these approaches do not focus on doing any deeper inference. However, the task of machine comprehension requires an ability to perform inference over paragraph length texts to seek the answer. This is challenging for most IR and NLP techniques. In this paper, we presented a strategy for learning answer-entailing structures that helped 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 and Cohn, 2006), text-hypothesis alignment in RTE (Sammons et al., 2009; MacCartney et al., 2008; Yao et al., 2013a; Sultan et al., 2014), question-answer alignment in QA (Berant et al., 2013; Yih et al., 2013; Yao and Van Durme, 2014), etc. Again 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 and Thompson, 1988)) and co-reference 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 built a classifier using some labelled corpora. On the other hand we learnt these structures and performed machine comprehension 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 answers.

Another advantage of our approach is its simple and elegant extension to multi-task settings. 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.

8 Conclusion

In this paper, we addressed the problem of machine comprehension which tests language understanding through multiple choice question answering tasks. We posed the task as an extension to RTE. Then, we proposed a solution by learning latent alignment structures between texts and the hypotheses in the equivalent RTE setting. The task requires 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. We also discuss strengths and limitations of our model in a more fine-grained analysis. In the future, we plan to use logic-like semantic representations of texts, questions and answers and explore approaches to perform structured inference over richer semantic representations.

Acknowledgments

The authors would like to thank the anonymous reviewers, along with Sujay Jauhar and Snigdha Chaturvedi for their valuable comments and suggestions to improve the quality of the paper.

References

[Berant et al.2013] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544. ACL.

[Blunsom and Cohn2006] Phil Blunsom and Trevor Cohn. 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*, pages 65–72. Association for Computational Linguistics.

[Burges2013] Christopher JC Burges. 2013. Towards the machine comprehension of text: An essay. Technical report, Microsoft Research Technical Report MSR-TR-2013-125, 2013, pdf.

[Chakrabarti et al.2008] Soumen Chakrabarti, Rajiv Khanna, Uma Sawant, and Chiru Bhattacharyya. 2008. Structured learning for non-smooth ranking losses. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 88–96.

[Chan et al.2011] Tsz Ping Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking bilingually extracted paraphrases using monolingual distributional similarity. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–42.

[Collobert et al.2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

[Cucerzan and Agichtein2005] S. Cucerzan and E. Agichtein. 2005. Factoid question answering over unstructured and structured content on the web. In *Proceedings of TREC 2005*.

[Dubey et al.2009] Avinava Dubey, Jinesh Machchhar, Chiranjib Bhattacharyya, and Soumen Chakrabarti. 2009. Conditional models for non-smooth ranking loss functions. In *ICDM*, pages 129–138.

[Evgeniou and Pontil2004] Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117.

[Fellbaum1998] Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

[Feng and Hirst2014] Vanessa Wei Feng and Graeme Hirst. 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)*, pages 511–521.

[Ferrucci2012] David A Ferrucci. 2012. Introduction to this is watson. *IBM Journal of Research and Development*, 56(3.4):1–1.

[Finley and Joachims2008] T. Finley and T. Joachims. 2008. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, pages 304–311.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Iyyer et al.2014] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Empirical Methods in Natural Language Processing*.

[Jansen et al.2014] Peter Jansen, Mihai Surdeanu, and Peter Clark. 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)*, pages 977–986.

[Järvelin and Kekäläinen2002] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

[Joachims2002] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

[Joachims2006] T. Joachims. 2006. Training linear SVMs in linear time. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD)*, pages 217–226.

[Li and Roth2002] Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7.

[Lin and Pantel2001] Dekang Lin and Patrick Pantel. 2001. Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328.

[MacCartney et al.2008] Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the conference on empirical methods in natural language processing*, pages 802–811.

[Mann and Thompson1988] William C Mann and Sandra A Thompson. 1988. {Rhetorical Structure Theory: Toward a functional theory of text organisation}. *Text*, 3(8):234–281.

[Mittal and Mittal2011] Sparsh Mittal and Ankush Mittal. 2011. Versatile question answering systems: seeing in synthesis. *International Journal of Intelligent Information and Database Systems*, 5(2):119–142.

[Moldovan et al.2003] Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21(2):133–154.

[Richardson et al.2013] Matthew Richardson, J.C. Christopher Burges, and Erin Renshaw. 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*, pages 193–203.

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

[Srivastava and Hovy2013] Shashank Srivastava and Dirk Hovy. 2013. A walk-based semantically enriched tree kernel over distributed word representations. In *Empirical Methods in Natural Language Processing*, pages 1411–1416.

[Stern and Dagan2012] Asher Stern and Ido Dagan. 2012. Biutee: A modular open-source system for recognizing textual entailment. In *Proceedings of the ACL 2012 System Demonstrations*, pages 73–78.

[Sultan et al.2014] Arafat Md Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association of Computational Linguistics – Volume 2, Issue 1*, pages 219–230.

[Weston et al.2014] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

[Weston et al.2015] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks.

[Yao and Van Durme2014] Xuchen Yao and Benjamin Van Durme. 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)*, pages 956–966. Association for Computational Linguistics.

[Yao et al.2013a] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013a. A lightweight and high performance monolingual word aligner. In *ACL (2)*, pages 702–707.

[Yao et al.2013b] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013b. Semi-markov phrase-based monolingual alignment. In *Proceedings of EMNLP*.

[Yih et al.2013] Wentau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

[Yu and Joachims2009] Chun-Nam Yu and T. Joachims. 2009. Learning structural svms with latent variables. In *International Conference on Machine Learning (ICML)*.

[Yuille and Rangarajan2003] A. L. Yuille and Anand Rangarajan. 2003. The concave-convex procedure. *Neural Comput.*

[Zhang and Lee2003] Dell Zhang and Wee Sun Lee. 2003. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32. ACM.

[Zhu et al.2011] Jun Zhu, Ning Chen, and Eric P Xing. 2011. Infinite latent svm for classification and multi-task learning. In *Advances in neural information processing systems*, pages 1620–1628.