Machine Comprehension with Discourse Relations

Karthik Narasimhan
CSAIL, MIT
karthikn@csail.mit.edu

Regina Barzilay
CSAIL, MIT
regina@csail.mit.edu

Abstract

This paper proposes a novel approach for incorporating discourse information into machine comprehension applications. Traditionally, such information is computed using off-the-shelf discourse analyzers. This design provides limited opportunities for guiding the discourse parser based on the requirements of the target task. In contrast, our model induces relations between sentences while optimizing a task-specific objective. This approach enables the model to benefit from discourse information without relying on explicit annotations of discourse structure during training. The model jointly identifies relevant sentences, establishes relations between them and predicts an answer. We implement this idea in a discriminative framework with hidden variables that capture relevant sentences and relations unobserved during training. Our experiments demonstrate that the discourse aware model outperforms state-of-the-art machine comprehension systems.[1]

1 Introduction

The task of machine comprehension concerns the automatic extraction of answers from a given passage. Often, the relevant information required to answer a question is distributed across multiple sentences. Understanding the relation(s) between these sentences is key to finding the correct answer. Consider the example in fig. 1. To answer the question about *why Sally put on her shoes* , we need to infer that *She put on her shoes* and *She went outside to walk* are connected by a causality relation.

> Sally liked going outside. She put on her shoes. She went outside to walk. [...] Missy the cat meowed to Sally. Sally waved to Missy the cat. [...] Sally hears her name. ”Sally, Sally, come home”, Sally's mom calls out. Sally runs home to her Mom. Sally liked going outside.
>
> Why did Sally put on her shoes?
> A) To wave to Missy the cat
> B) To hear her name
> C) *Because she wanted to go outside*
> D) To come home

Figure 1: Sample story excerpt from a passage in the MCTest dataset.[2] Correct answer is in italics.

Prior work has demonstrated the value of discourse relations in related applications such as question answering (Jansen et al., 2014). Traditionally, however, these approaches rely on outputs from off-the-shelf discourse analyzers, using them as features for target applications. Such pipeline designs provide limited opportunities for guiding the discourse parser based on the requirements of the end task. Given a wide spectrum of discourse frameworks (Mann and Thompson, 1988; Prasad et al., 2008; Wolf and Gibson, 2005), it is not clear a priori what the optimal set of discourse annotations is for the task. Moreover, a generic discourse parser may introduce additional errors due to the mismatch between its training corpus and a dataset used in an application. In fact, the largest discourse treebanks are based on newspaper corpora (Prasad et al., 2008; Carlson et al., 2002), which differ significantly in style from text used in machine comprehension corpora (Richardson et al., 2013).

In this paper, we propose a novel approach for incorporating discourse structure into machine

[1] Code and data are available at http://people.csail.mit.edu/karthikn/mcdr.

[2] http://research.microsoft.com/en-us/um/redmond/projects/mctest/

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1253–1262,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

comprehension applications. Rather than using a standalone parser that is trained on external supervised data to annotate discourse relations, the model induces relations between sentences while optimizing a task-specific objective. This design biases the model to learn relations at a granularity optimized for the machine comprehension task. In contrast to a generic discourse analyzer, our method can also utilize additional information available in the machine comprehension context. For instance, question types provide valuable cues for determining discourse relations, and thus can facilitate learning.

We implement these ideas in a discriminative log-linear model with hidden variables. The model jointly identifies relevant sentences, establishes relations between them and predicts an answer. Since the same set of sentences can give rise to multiple questions, we do not limit the model to a single discourse relation, but rather model a distribution over possible relations. During training, we only have access to questions and gold answers. Since relevant sentences and their relations are not known, we model them as hidden variables. To guide the model towards linguistically plausible discourse relations, we add a few seed markers that are typical of each relation. The model predicts relations not only based on the sentences, but also incorporates information about the question. By decomposing the dependencies between model components, we can effectively train the model using a standard gradient descent approach.

We evaluate our model using a recently released machine comprehension dataset (Richardson et al., 2013). In this corpus, roughly half of the questions rely on multiple sentences in the passage to generate the correct answer. For baselines, we use the best published results on this dataset. Our results demonstrate that our relation-aware model outperforms the individual baselines by up to 5.7% and rivals the performance of a state-of-the-art combination system. Moreover, we show that the discourse relations it predicts for sentence pairs exhibit considerable overlap with relations identified by human annotators.

2 Related Work

Machine Comprehension Following traditional methods in question answering, most approaches to machine comprehension focus on analyzing the connection between the question, candidate answer and the document. For instance, Richardson et al. (2013) show that using word overlap alone provides a good starting point for the task. Using textual entailment output (Stern and Dagan, 2011) and embedding-based representations (Iyyer et al., 2014) further improves the result. Even though these methods operate at a paragraph level, they do not model relations between sentences. For instance, in their work on factoid question answering using recursive neural networks, Iyyer et al. (2014) average the sentence vectors element-wise when considering more than one sentence.

A notable exception is the approach proposed by Berant et al. (2014). Their approach builds on a semantic representation that encodes a number of inter-event relations, such as *cause* and *enable*. These relations straddle the boundary between discourse and semantic connections, since most of them are specific to the domain of interest. These relations are identified in a supervised fashion using a significant amount of manual annotations. In contrast, we are interested in extracting discourse relations with minimal additional annotation, relying primarily on the available question-answer pairs. As a result, we look at a smaller set of basic relations that can be learned without explicit annotations.

Discourse analysis for Question Answering Prior work has established the value of domain-independent discourse relations in question answering applications (Verberne et al., 2007; Jansen et al., 2014; Chai and Jin, 2004). For instance, Verberne et al. (2007) propose an answer extraction technique that treats question topics and answers as siblings in a Rhetorical Structure Theory (RST) tree, significantly improving performance on *why*-questions. Chai and Jin (2004) argue that incorporating discourse processing can significantly help context question answering, a task in which subsequent questions may refer to entities or concepts in previous questions. Jansen et al. (2014) utilize discourse information to improve reranking of human-written answers for non-factoid questions. They experiment with both shallow discourse markers and deep representations based on RST parsers to rerank answers for *how* and *why*-type questions[3].

While the above approaches vary greatly in

[3] They use data from Yahoo! Answers and a Biology textbook.

terms of their design, they incorporate discourse information in a similar fashion, adding it as features to a supervised model. The discourse information is typically computed using discourse parsers based on frameworks like RST (Feng and Hirst, 2014) or PDTB (Lin et al., 2014), trained using supervised data. In contrast, our goal is to learn discourse relations driven by the task objective. The set of these relations does not capture the richness of discourse representations considered in traditional discourse theories (Mann and Thompson, 1988; Prasad et al., 2008). However, we learn them without explicit annotations of discourse structure, and demonstrate that they improve model performance.

3 Task Description and Approach

We focus on the task of machine comprehension, which involves answering questions based on a passage of text. Concretely, let us consider a passage $p_i = \{\mathcal{Z}_i, \mathcal{Q}_i\}$ to consist of a set of sentences $\mathcal{Z}_i = \{z_{in}\}$ and a set of questions $\mathcal{Q}_i = \{q_{ij}\}$, with each question also having a set of answer choices $\mathcal{A}_{ij} = \{a_{ijk}\}$. We denote the correct answer choice for a question q_{ij} as a^*_{ij}. Given a set of training passages $\mathcal{P}_{train}$ with questions annotated with the correct answer choice, the task is to be able to answer questions accurately in a different set of passages $\mathcal{P}_{test}$.

Figure 1 shows an example of a passage, along with a question and answer choices. The only (weak) source of supervision available is the correct answer choice for each question in training. We do not use any extra annotations during training. We propose joint probabilistic models to address this task, that can learn to identify single or multiple relevant sentences given a question, establish a relation between them and score the answer choices.

We explore three different discriminative models, ranging from a simple one that answers questions using a single sentence in the passage, to one that infers relations between multiple sentences to score answer choices. We defer the description of the features used in our models to section 3.1.

Model 1 In our first model, we assume that each question can be answered using a single sentence from the passage. Treating the sentence as a hidden variable, we define a joint model for a sentence $z \in \mathcal{Z}$ and an answer choice $a \in \mathcal{A}_j$, given a question q_j.

$$P(a, z \mid q_j) \; = P(z \mid q_j) \cdot P(a \mid z, q_j) \quad (1)$$

We define the joint probability as a product of two distributions. The first is the conditional distribution of sentences in the paragraph given the question. This is to help identify the right sentence required to answer the question. The second component models the conditional probability of an answer given the question q and a sentence z. For both component probabilities, we use distributions from the exponential family with features and associated weights:

$$P(z \mid q) \propto e^{\theta_1 \cdot \phi_1(q,z)}$$
$$P(a \mid z, q) \propto e^{\theta_2 \cdot \phi_2(q,a,z)}$$

where ϕs are the feature functions and θs are the corresponding weight vectors.

We cast the learning problem as estimation of the parameter weights to maximize the likelihood of the correct answers in the training data. We consider soft assignments to z and marginalize over all its values to get the likelihood of an answer choice:

$$P(a_{jk} \mid q_j) \; = \sum_n P(a_{jk}, z_n | q_j) \quad (3)$$

This results in the following regularized likelihood objective to maximize:

$$\begin{aligned} &L_1(\theta; \mathcal{P}_{train}) \\ &\quad = \log \sum_{i=1}^{|\mathcal{P}_{train}|} \sum_{j=1}^{|\mathcal{Q}_i|} P(a^*_{ij} \mid q_{ij}) - \lambda ||\theta||^2 \end{aligned} \quad (4)$$

Model 2 We now propose a model for the multi-sentence case where we make use of more than a single relevant sentence pertaining to a question. Considering that a majority of the questions in the dataset can be answered using two sentences, we restrict ourselves to sentence pairs for purposes of computational tractability. We define the new joint model as:

$$\begin{aligned} P(a, z_1, z_2 \mid q) \; &= P(z_1 \mid q) \cdot P(z_2 \mid z_1, q) \\ &\cdot P(a \mid z_1, z_2, q) \end{aligned} \quad (5)$$

where the new components are also exponential-family distributions:

$$P(z_2 \mid z_1, q) \propto e^{\theta_3 \cdot \phi_3(q, z_1, z_2)}$$
$$P(a \mid z_1, z_2, q) \propto e^{\theta_2 \cdot \phi_2(q,a,z_1,z_2)}$$

Here, we have three components: the conditional probability of a sentence z_1 given q, of a second sentence z_2 given q and z_1, and of the answer a given q and the sentences.[4] Ideally, we would be able to consider all possible pairs of sentences in a given paragraph. However, to reduce computation costs in practice, we use a sentence window k and consider only sentences that are at most k away from each other.[5] We hence maximize:

$$L_2(\theta; \mathcal{P}_{train}) = \tag{7}$$
$$\log \sum_{i=1,j=1,m=1}^{|\mathcal{P}_{train}|,|\mathcal{Q}_i|,|\mathcal{Z}_i|} \sum_{n \in [m-k, m+k]} P(a^*_{ij}, z_{im}, z_{in} \mid q_{ij})$$
$$- \lambda ||\theta||^2$$

Model 3 In our next model, we aim to capture important relations between sentences. This model has two novel aspects. First, we consider a distribution over relations between sentence pairs as opposed to a single relation. Second, we utilize the cues from the question as context to resolve ambiguities in sentences pairs with multiple plausible relations.

We add in a hidden variable $r \in \mathcal{R}$ to represent the relation type. We incorporate features that tie in the question type with the relation type, and that connect the type of relation to the lexical and syntactic similarities between sentences. Our relation set $\mathcal{R}$ consists of the following relations:

- *Causal* : Causes of events or reasons for facts.
- *Temporal* : Time-ordering of events
- *Explanation* : Predominantly dealing with *how*-type questions.
- *Other* : A relation other than the above[6]

We can now modify the joint probability from (5) by adding in relation type r to get:

$$\begin{aligned} P(a, r, z_1, z_2 \mid q) &= P(z_1 \mid q) \cdot P(r \mid q) \\ &\cdot P(z_2 \mid z_1, r, q) \cdot P(a \mid z_1, z_2, r, q) \end{aligned} \tag{8}$$

where

$$P(r \mid q) \propto e^{\theta_4 \cdot \phi_4(q,r)} \tag{9a}$$

$$P(z_2 \mid z_1, r, q) \propto e^{\theta_3 \cdot \phi_3(q,r,z_1,z_2)} \tag{9b}$$

$$P(a \mid z_1, z_2, r, q) \propto e^{\theta_2 \cdot \phi_2(q,r,a,z_1,z_2)} \tag{9c}$$

The extra component $P(r \mid q)$ is the conditional distribution of the relation type r depending on the question. This is to encourage the model to learn, for instance, that *why*-questions correspond to the *causal* relation. We also add in extra features to $P(z_2 \mid z_1, r)$, that help select a sentence pair conditioned on a relation. The likelihood objective to maximize is:

$$L_3(\theta; \mathcal{P}_{train}) \tag{10}$$
$$= \log \sum_{i,j,m,r \in \mathcal{R}} \sum_{n \in [m-k, m+k]} P(a^*_{ij}, z_{im}, z_{in}, r \mid q_{ij})$$
$$- \lambda ||\theta||^2$$

We maximize the likelihood objectives using LBFGS-B (Byrd et al., 1995). We compute the gradients required using Automatic Differentiation (Corliss, 2002).

To predict an answer for a test question q_j, we simply marginalize over all the hidden variables and choose the answer that maximizes $P(a_{jk} \mid q_j)$:

$$\hat{a}_j = \underset{k}{argmax}\ P(a_{jk} | q_j)$$

3.1 Features

We use a variety of lexical and syntactic features in our model. We employ the Stanford CoreNLP tool (Manning et al., 2014) to pre-process the data. Other than commonly used features in Q&A systems such as unigram and bigram matches, part-of-speech tags, syntactic features, we also add in features specific to our model.

We first define some terms used in our description. *Entities* are coreference-resolved nouns or pronouns. *Actions* refer to verbs other than auxiliary ones such as *is, are, was* and *were*. An *entity graph* is a graph between entities present in a sentence. We create an entity graph by collapsing nodes in the dependency graph and storing the intermediate nodes between any two entity nodes in the edge between the nodes. We refer to the words in a question q as q-words and similarly to words in an answer a as a-words and those in a sentence z as z-words. Figure 2 shows an example of an entity graph constructed from the dependency graph of a sentence.

We divide the features into 4 sets (ϕ_{1-4}), corresponding to each component probability in (8). Types 1 and 2 are inspired by prior work in question classification/answering (Blunsom et al., 2006; Jansen et al., 2014). Feature types 3 and 4 are specific to our models, primarily dealing with relation types.

[4] Since this component replaces the second component in model 1, we use the same subscript 2 for its feature set ϕ.

[5] Including the case where $z_1 = z_2$.

[6] This includes the no-relation cases

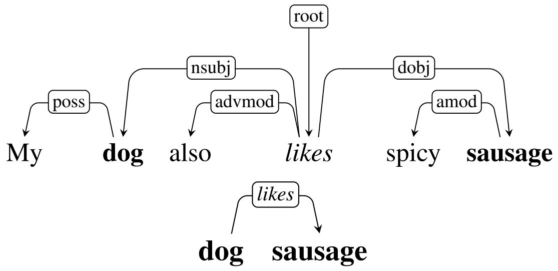

Figure 2: **Top**: Dependency graph, **Bottom**: Entity graph for an example sentence. Entities are in bold, actions are in italics.

Relation	Word list
Causal	because, why, due, so
Temporal	when, between, soon, before, after, during, then, finally, now, nowadays, first
Explanation	how, by, using

Table 1: Seed marker words for relations used by the model.

Split	MC160		MC500	
	Passages	Questions	Passages	Questions
Train	70	280	300	1200
Dev	30	120	50	200
Test	60	240	150	600

Table 2: Dataset Statistics

Type 1 (ϕ_1) These features are primarily intended to help the model select the most relevant sentence from the passage for a question. We add commonly used features such as unigram and bigram matches, syntactic root match, entity and action matches, missed entities/actions (in q but absent in z) and fractional coverage of q-words in z. In addition, we use matches between the edges of the entity graph of q and z. We also have second-order features that are a cross of each feature mentioned above with the question word (*how, what, when*, etc.).

Type 2 (ϕ_2) Features in ϕ_2 capture interactions between the answer a, question q and sentence(s) (z_1, z_2 in models 2,3 or z in model 1). For the first-order features, we use ones similar to those in ϕ_1 for lexical, syntactic, entity and action matches/misses between a and z. In addition, we add in a *neighbor match* feature, which checks for matches between the neighborhood of a word from a that occurs in z, and q-words. Another feature we employ is the *joint match* between z-words and the union of a-words and q-words. Finally, we add in a sliding window (*SW*) feature, computing its value as in Richardson et al. (2013).

Type 3 (ϕ_3) The next set of features are specific to only models 2 and 3, used to connect sentences z_1 and z_2 (and a relation r in model 3 only). We use features like the inter-sentence distance and the presence of relation-specific markers in the sentences. We also cross the latter features with *entity* and *action* matches between z_1 and z_2. For the relation-specific words for each relation (except *Other*), we use words (see Table 1) derived mainly from Marcu (1997)'s list of discourse markers.

Type 4 (ϕ_4) The final set of features (used only in model 3) are present to help the model learn connections between the words in the question and the relation type r. Specifically, we check if the interrogative word in the question matches the class represented by r. For instance, the word *why* matches the *Causal* relation.

For the match-type features of all four types, we use the match count as the feature value if the count is non-zero. If the count is zero, we instead set a corresponding *zero* feature[7] to 1.

4 Experimental setup

Data and Setup We run our experiments on a recently compiled dataset for machine comprehension: MCTest (Richardson et al., 2013). The data consists of two distinct sets: MC160 and MC500, which are of different sizes. Table 2 gives details on the data splits for each dataset. Each passage has 4 questions, with 4 answer choices each. The questions are also annotated into 2 types: *single*, if the question can be answered using a single sentence in the passage, or *multi* otherwise. We do not use the type information in our learning; we only use it for categorizing accuracy during evaluation. We report final results on all our models trained with $\lambda = 0.1$, tuned using the Dev sets.

Evaluation We report accuracy scores for each model averaged over the questions in the test data. For each question, the system gains 1 point if it scores the correct answer highest and 0 otherwise. In case of ties, we use an inverse weighting

[7] For each match feature, like *Entity-Match*, we have a corresponding *zero* feature, *Entity-Match-Zero*

Model	MC160						MC500					
	dev			test			dev			test		
	Single	Multi	All	Single	Multi	All	Single	Multi	All	Single	Multi	All
SWD	67.92	50.74	58.33	75.89	60.15	67.5	63.95	54.38	58.5	63.23	57.62	60.16
RTE	64.15	53.73	58.33	57.14	59.37	58.33	58.13	47.36	52	70.22	42.37	55
RTE+SWD	71.69	59.6	65	76.78	62.5	69.16	73.25	57.89	64.5	68.01	59.45	63.33
Model 1	**78.45**	**60.57**	**68.47**	**83.25**	60.35	71.04†	**74.41**	57.01	64.5†	**70.58**	57.77	63.58†
Model 2	74.68	60.07	66.52	81.47	64.25	72.29†	73.25	**61.4**	**66.5***†	66.17	**59.9**	62.75†
M2 + RST	72.79	58.58	64.86	79.68	61.91	70.20†	72.09	57.89	64.0†	66.54	59.29	62.58
Model 3	72.79	60.07	65.69	82.36	**65.23**	**73.23***†	72.09	60.52	65.5*†	68.38	**59.9**	**63.75**†

Table 3: Accuracy (%) of the different baselines (in italics) and our models. **Single**: questions requiring single sentence to answer; **Multi**: questions requiring multiple sentences to answer. Sentence window (k) = 4 for models 2 and 3. Best scores are shown in bold. Statistical significance (shown only for *All* columns) of $p < 0.05$ using two-tailed paired t-test: *vs *SWD*, †vs *RTE*.

scheme to assign partial credit. So, if three answers (including the correct one) tie for the highest score, the system gains 1/3 points.

Baselines We use the systems proposed by Richardson et al. (2013) as our baselines. These systems have the best reported scores on this dataset. The first baseline, *SWD*, uses a sliding window to count matches between the passage words and the words in the answer. This is then combined with a score representing the average distance between answer and question words in the passage. The second baseline, *RTE*, uses a *textual entailment* recognizer (Stern and Dagan, 2011) to determine if the answer (turned into a statement along with the question) is entailed by the passage. The third system, *RTE+SWD*, is a weighted combination of the first two baselines and achieves the highest accuracy on the dataset.

5 Results

Comprehension accuracy Table 3 shows that our relation-aware model 3 outperforms individual baselines on both test sets. On the MC160 test set, the model achieves the best performance of 73.23% accuracy, outperforming the *SWD* baseline by 5.7% and the *RTE+SWD* combination by 4.07%. The major gains of model 3, which utilizes inter-sentential relations, over model 1 can be seen in the accuracy of *multi* type questions with a jump of almost 5% absolute in accuracy (statistically significant with $p < 0.05$). On the MC500 test set, we again find that model 3, with a score of 63.75%, provides a gain of 3.5% over *SWD* and is comparable to the performance of *RTE+SWD* (63.33%)

The importance of utilizing multiple relevant sentences to score answers is evident from the higher scores of models 2 and 3 on *multi* type questions in both test sets. However, model 1, which retrieves only a single relevant sentence for each question, achieves the best scores on the *single* type questions up to 83.25% on MC160 test. One reason for this could be the larger search space for model 3 over pairs of sentences compared to just single sentences for model 1.

Table 4 shows the variation of our model's accuracy with the question type. We see that the model deals well with *what, where* and *why* type questions in MC500, achieving almost 67-69% accuracy.[8] The major errors (in MC500) seem to come from the *how*-questions, where the model's accuracy is low (48%). In MC160, the accuracy is even higher for *what*-questions (almost 80%). On the other hand, the model does slightly worse on *why*-questions, with only 60% accuracy.

RST augmented model Further, we experiment with adding in relations extracted by a publicly available RST parser (Feng and Hirst, 2012). The parser extracts a tree with the passage sentences as its leaves and relations as interior nodes in the tree. From this tree, we compute the relation between a pair of sentences as their lowest common ancestor. If one of the sentences is broken down into clauses, we use them all to gather multiple relations. We add in features that combine the RST-predicted relation with the interrogation word of the question, and with entity and action matches between sentence pairs.

We can see from Table 3 that adding in RST features to model 2 (M2+RST) does not give the

[8] Note that *what*-questions may also require causal/temporal/explanation relations to answer.

Question Type	MC160		MC500	
	Dev	Test	Dev	Test
how	50.00 (10)	71.42 (21)	54.54 (11)	48.83 (43)
what	64.40 (59)	79.36 (126)	63.15 (114)	67.19 (317)
where	30.76 (13)	91.66 (12)	82.60 (23)	68.96 (58)
which	75.00 (4)	33.33 (6)	25.00 (4)	48.00 (25)
who	70.50 (17)	67.85 (28)	62.50 (16)	59.74 (77)
why	85.71 (14)	59.45 (37)	65.38 (26)	69.35 (62)
when	100.0 (2)	80.00 (5)	100.0 (4)	62.50 (8)
whose	-	-	-	66.67 (3)
(other)	100.0 (1)	40.00 (5)	50.00 (2)	14.28 (7)

Table 4: Accuracy (%) of model 3 by question type for question in MC160 and MC500 dev and test sets. Numbers in parentheses indicate the number of questions of each type.

same performance as model 3. In fact, the model performs slightly worse than model 2, which does not utilize inter-sentential relations. Our analysis of the RST trees reveals that for a vast majority of sentence pairs (77%), the RST algorithm predicts the *elaboration* relation which does not provide an informative distinction.

5.1 Analysis

To gain further insight into the workings of our model, we perform several analyses on model 3 using human judgements. We annotate 240 questions from the test set of MC160 with the most relevant sentences[9] in the passage for each question. In addition, if they chose more than a single relevant sentence, we also asked the annotators to mark the most appropriate relation (from our set of relations used in model 3) between the sentence pairs.[10] We find that 146 question annotations contain a single relevant sentence and 94 contain multiple sentences.[11] We obtain 103 sentence pairs with annotated relations.

Annotation statistics We select a random subset of 134 questions from this data to annotate twice and compute inter-annotator agreement. The second annotator agreed completely with the sentence predictions of the first annotator in 76.11% cases and both annotators agreed on at least one sentence in 94.77% of the questions. The agreement on relations annotated over common sentence pairs is 68.6%, with $\kappa = 0.462$. We find that out of the 103 annotated sentence pairs, 67 are next to each other in the passage while 27 are at a distance of two and 9 pairs are at a distance of three or more.

It has been well documented that identifying discourse relations without explicit markers is significantly harder than with markers (Pitler et al., 2008; Lin et al., 2009; Park and Cardie, 2012). We compute statistics on the presence of discourse markers anywhere in the manually picked sentence(s) for each question. We find that only 33.89% of these pairs have a relevant discourse marker present in either sentence. We consider a discourse marker as relevant if it occurs in our marker list for the annotated relation. Further, if we only consider markers occurring at the beginning or end of the sentences, this number drops to 9.23% of sentences. Since we consider relations between sentence pairs, most explicit markers that could help identify these relations would occur at an extremity of either sentence. We point out that these numbers are an over-estimation since many of the markers occur in syntactic roles as opposed to discourse in the sentences (ex. *so* in *This is so good* compared to *So, he decided to ...*). These statistics reflect the difficulty of the problem since operating over implicit relations is much harder.

Sentence Retrieval We analyze our models' ability to predict relevant sentences given only the question. For each question, we order the pairs scored by a model in descending order of their probability according to $P(z_1, z_2 \mid q)$ and compare them to the annotated pairs, reporting recall at various thresholds.

This is a stringent evaluation primarily due to

[9] The annotators are native English speakers.

[10] If there were more than *two* relevant sentences, we asked them to mark relations between all pairs. This was a very rare occurrence though.

[11] We found that some of the *multiple* questions did not require multiple sentences to answer and conversely, some *single* questions required more than one sentence to answer.

Question Type (#)	**R @ 1**				**R @ 2**				**R @ 5**			
	Freq	M1	M2	M3	Freq	M1	M2	M3	Freq	M1	M2	M3
how (21)	9.67	29.03	32.25	35.48	9.67	32.25	45.16	48.38	19.35	51.61	58.06	64.51
what (126)	3.64	37.85	35.59	35.02	9.37	39.54	47.45	45.76	21.81	53.67	63.84	63.27
where (12)	13.63	50.00	50.00	56.25	13.63	50.00	68.75	62.5	45.45	68.75	75.00	81.25
which (6)	0.0	21.42	21.42	14.28	9.09	21.42	21.42	14.28	9.09	21.42	35.71	42.85
who (28)	2.12	45.45	45.45	42.42	4.25	45.45	60.60	57.57	23.40	63.63	72.72	75.75
why (37)	9.09	34.32	31.34	34.32	2.27	35.82	40.29	40.29	38.63	44.77	53.73	52.23
when (5)	0.0	57.14	57.14	57.14	0.0	57.14	71.42	71.42	25.00	85.71	100.0	85.71
(other) (5)	0.0	42.85	28.57	42.85	0.0	42.85	57.14	57.14	100.0	71.42	71.42	71.42
Single (146)	6.84	56.16	53.42	51.36	11.64	58.21	66.43	63.69	33.56	72.60	80.13	80.82
Multi (94)	3.88	24.27	23.30	25.72	9.70	25.24	34.46	33.98	29.61	39.32	50.00	50.48
Overall (240)	5.11	**37.5**	35.79	36.36	10.51	38.92	**47.72**	46.30	31.25	53.12	62.5	**63.06**

Table 5: Recall (%) of relevant sentence(s) in the ranking by models 1, 2 and 3 compared with a match-frequency baseline (Freq) at various thresholds, for different question types in MC160. Question frequencies are in parentheses. Bold numbers represent best scores.

two reasons. First, we do not use the candidate answers in selecting relevant sentences. Second, on the machine comprehension task, the model predicts answers by *marginalizing* over the sentences/sentence pairs. Hence, the model can score answers correctly even if the relevant sentence(s) are not at the top of its sentence distribution calculated here. We compute the distribution over sentence pairs as:

$$P(z_1, z_2|q) = \sum_{r \in \mathcal{R}} P(z_1 \mid q) \cdot P(r \mid q) \cdot P(z_2|z_1, r, q)$$

For comparison, we add in a baseline (Freq) that orders sentences using the sum of unigram and bigram matches with the question (in descending frequency).

Table 5 shows that our models perform significantly better than the Freq baseline over all question types. For the *single*-question case, we observe that model 3 ranks the annotated sentence at the top of its distribution around 51% of the time and 80% of the time in the top 5. For *multi*-sentences, these recall numbers drop to around 25% (@1) and 50% (@5). We also observe that models 2 and 3 perform better than model 1 on the *multi*-sentence cases. The similar sentence recall of models 2 and 3 also point to the fact that the gains from model 3 on comprehension accuracy are due to its ability to utilize relations between the sentences.

We observe that *where, when* and *who* questions have the highest recalls. This is likely because these questions often have characteristic words occurring in the sentences (such as *here, there, after, before, him, her*). In contrast, questions asking *how, which* and *why* have lower recalls since they often involve reasoning over multiple sentences. *What*-questions are somewhere in between since their complexity varies from question to question.

Relation Retrieval We examine how well our model can predict relations between given sentence pairs. For each annotated pair of sentences, we calculate the relation distribution and compute the relation recall at various thresholds of the ranking by probability. The relation distribution is computed as:

$$P(r|z_1, z_2, q) = \frac{P(r \mid q) \cdot P(z_2|z_1, r, q)}{\sum_{r' \in \mathcal{R}} P(r' \mid q) \cdot P(z_2|z_1, r', q)}$$

From table 6, we observe that our model's top prediction matches the manual annotations (overall) 51% of the time. The model predicts *causal* and *other* relations more accurately than the other two.

Relation (#)	**R @ 1**	**R @ 2**
Causal (32)	56.25	75.00
Temporal (11)	27.27	54.54
Explanation (6)	16.66	33.33
Other (54)	57.40	64.81
Overall	51.45	65.04

Table 6: Recall of annotated relations at various thresholds in the ordered relation distribution predicted by model 3. Relation frequencies are in parentheses.

6 Conclusions

In this paper, we propose a new approach for incorporating discourse information into machine comprehension applications. The key idea is to implant discourse analysis into a joint model for comprehension. Our results demonstrate that the discourse-aware model outperforms state-of-the-art standalone systems, and rivals the performance of a system combination. We also find that features derived from an off-the-shelf parser do not improve performance of the model. Our analysis also demonstrates that the model accuracy varies significantly according to the question type. Finally, we show that the predicted discourse relations exhibit considerable overlap with relations identified by human annotators.

Acknowledgements

We thank Lyla Fischer, Rushi Ganmukhi, Hrishikesh Joshi and Deepak Narayanan for helping with annotating the MC160 test set. We also thank the anonymous ACL reviewers and members of MIT's NLP group for their insightful comments and suggestions. This research is developed in a collaboration of MIT with the Arabic Language Technologies (ALT) group at Qatar Computing Research Institute (QCRI) within the Interactive sYstems for Answer Search (IYAS) project.

References

[Berant et al.2014] Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar, October. Association for Computational Linguistics.

[Blunsom et al.2006] Phil Blunsom, Krystle Kocik, and James R Curran. 2006. Question classification with log-linear models. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 615–616. ACM.

[Byrd et al.1995] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

[Carlson et al.2002] Lynn Carlson, Mary Ellen Okurowski, Daniel Marcu, Linguistic Data Consortium, et al. 2002. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania.

[Chai and Jin2004] Joyce Y Chai and Rong Jin. 2004. Discourse structure for context question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL*, volume 2004, pages 23–30. Citeseer.

[Corliss2002] George Corliss. 2002. *Automatic differentiation of algorithms: from simulation to optimization*, volume 1. Springer Science & Business Media.

[Feng and Hirst2012] Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 60–68. Association for Computational Linguistics.

[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, Baltimore, Maryland, June. Association for Computational Linguistics.

[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, Baltimore, Maryland, June. Association for Computational Linguistics.

[Lin et al.2009] Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 343–351. Association for Computational Linguistics.

[Lin et al.2014] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, pages 1–34.

[Mann and Thompson1988] William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

[Manning et al.2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

[Marcu1997] Daniel Marcu. 1997. *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*. Ph.D. thesis, University of Toronto.

[Park and Cardie2012] Joonsuk Park and Claire Cardie. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 108–112. Association for Computational Linguistics.

[Pitler et al.2008] Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind K Joshi. 2008. Easily identifiable discourse relations.

[Prasad et al.2008] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

[Richardson et al.2013] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, pages 193–203.

[Stern and Dagan2011] Asher Stern and Ido Dagan. 2011. A confidence model for syntactically-motivated entailment proofs. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 455–462, Hissar, Bulgaria, September. RANLP 2011 Organising Committee.

[Verberne et al.2007] Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–736. ACM.

[Wolf and Gibson2005] Florian Wolf and Edward Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–287.