Question Answering in Natural Language Narratives Using Symbolic Probabilistic Reasoning

Hannaneh Hajishirzi and Erik T. Mueller

hannaneh.hajishirzi@disneyresearch.com. etm@us.ibm.com
Disney Research, IBM

Abstract

We present a framework to represent and reason about narratives. We build a symbolic probabilistic representation of the temporal sequence of world states and events implied by a narrative using statistical approaches. We show that the combination of this representation together with domain knowledge and symbolic probabilistic reasoning algorithms enables understanding of a narrative and answering semantic questions whose responses are not contained in the narrative. In our experiments, we show the power of our framework (vs. traditional approaches) in answering semantic questions for two domains of RoboCup soccer commentaries and early reader children stories focused on spatial contexts.

1 Introduction

Semantic understanding of narratives and answering questions about them are important problems in natural language processing (Hobbs *et al.* 1993). These are fundamental to question answering systems, help desk systems, dialogue generators, and robot command interfaces. Most current question answering systems (e.g., (Rilo & Thelen 2000; Poon & Domingos 2009)) are designed to answer queries whose responses can be located in the text. For instance, in Figure 1, the question "who picked off the ball?" can be answered by syntactically processing the text. However, there are many questions whose responses are not explicitly mentioned in the text, and responding to them requires semantic understanding of the text. For example, the question "Who has the possession of the ball at every step?" may require understanding real events that happen in the game and following those events to track the possession of the ball.

There is a growing interest in semantic parsing of texts by mapping them to a sequence of meaningful events (Bejan 2008; Branavan *et al.* 2009; Vogel & Jurafsky 2010; Chen, Kim, & Mooney 2010; Hajishirzi *et al.* 2011; Liang, Jordan, & Klein 2009). This mapping allows deeper understanding of the narratives, but does not succeed in answering semantic questions without using advanced reasoning techniques. In this paper we present a framework to answer these semantic questions whose responses cannot be located in the text.

We show that a combination of probabilistic symbolic representation of narratives together with domain knowledge and inference algorithms enables understanding narratives and answering semantic questions about them. We represent a narrative as a sequence of sentences since the coherence among sentences provides a strong bias to understand the narrative. We model every sentence in the narrative with a probability distribution over different event types whose semantic descriptions are derived from the domain knowledge.

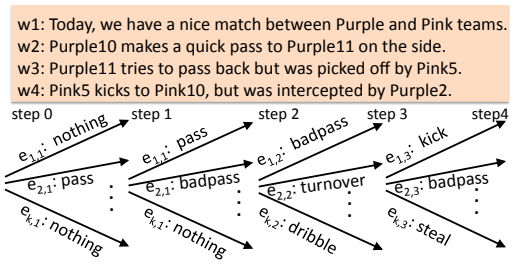

Figure 1: A part of soccer commentary and the corresponding event rankings.

Our question answering approach has three steps. The first step is to collect the domain knowledge in the form of a few event type descriptions and entity correlations. We use a symbolic language to represent the domain knowledge and narrative elements. The second step is to interpret every sentence by computing the likelihood of corresponding events and mapping every sentence to the event type which is most likely. The final step is to use a probabilistic reasoning algorithm to answer a query. Our reasoning algorithm builds a forest of narrative interpretations, answers a query in each interpretation, and integrates the responses. Our experiments demonstrate the power of our framework to answer semantic questions in the two domains of children's stories and soccer commentaries.

Related Work: There have been different approaches for semantic understanding of narratives, which are not focused on answering questions. (Branavan *et al.* 2009; Vogel & Jurafsky 2010; Chen, Kim, & Mooney 2010; Hajishirzi *et al.* 2011) use reinforcement learning or EM-like approaches to map sentences to event sequences in the domains of game instructions, help instructions, giving directions, and RoboCup soccer commentaries. These approaches take advantage of the properties of a special domain and interpret the sentences with high accuracy. In more general domains, (Bejan 2008) introduce an approach to infer events from a text, but do not take advantage of the coherence of natural language text, which provides a strong bias

to understand the text. (Hobbs *et al.* 1993) do not model uncertainty, an essential part of narrative understanding.

There have been semantic question answering systems in the literature. (Narayanan & Harabagiu 2004) use reasoning about actions for question answering. However, their sentence representation and reasoning algorithms are different from ours. (Poon & Domingos 2009) introduce a powerful framework for semantic parsing and answering questions about biomedical entities mentioned in the text. Still, these approaches cannot answer semantic questions whose responses cannot be found in the text.

2 Narratives

A narrative in our system is a text (written in English) that consists of a sequence of sentences in natural language. These sentences describe the temporal evolution of events that happen in real world. Specifically, a narrative is a sequence of length T of sentences $\langle w_1, w_2, \ldots, w_T \rangle$. In this paper we are focused on sports commentaries (RoboCup soccer commentaries) and children's stories (early reader stories with a focus on spatial contexts).

Sentences: Every sentence in the narrative either represents a fact about the world (e.g., "Not many birds fly at night.", "Offside has been called on the Pink team.") or an incremental update of the overall knowledge about the world (e.g., "Then the whip-poor-will flew away." or "Pink9 tries to kick to Pink10, but was defended by Purple3."). Because of the nature of natural language, sentences might not explicitly mention the name of the real events that occur in the world. For instance the above soccer sentence could be interpreted as *passing* between players, *unsuccessful passing* between players, a player *kicking* the ball, or a player *defending* the other player.

State of the world: The state of the world (e.g., the soccer game or the story scene) changes over time. For example, after the sentences "The pink goalie kicks off to Pink2." and "They went back home.", (partial) descriptions of the current state of the world would be 'Pink2 has possession of the ball' and 'their current location is home', respectively.

Meaning Representation Language: We use a symbolic language (similar to (Hajishirzi *et al.* 2011)) to model the narratives and prior knowledge in a structured way. This language allows us to capture general domain knowledge about event types. Moreover, it can be extended to model infinite domains using natural language processing tools. In each domain, our meaning representation language consists of domain entities, variables, state predicates, and event types.

For instance in the RoboCup domain, the language consists of a finite set of constants (e.g., $\texttt{Pink}_1$, $\texttt{Purple}_1$), variables (e.g., $\texttt{player}_1, \texttt{player}_2$), state predicates (e.g., $\texttt{holding}(\texttt{player}, \texttt{ball})$, $\texttt{atCorner}$, $\texttt{atPenalty}$), and event types (e.g., $pass(player_1, player_2)$, $kick(player_1)$, $steal(player_1)$). In the stories domain, the language consists of constants (e.g., $\texttt{Bus}$, $\texttt{Home}$, $\texttt{Mall}$), variables (e.g., $\texttt{person}$, $\texttt{object}$, $\texttt{location}$), state predicates (e.g., $\texttt{atloc}(\texttt{person}, \texttt{loc})$, $\texttt{in}(\texttt{object}, \texttt{place})$), and event types (e.g., $go(person, loc_1, loc_2)$, $walk(person, loc_1, loc_2)$).

Event types (e.g., $kick(player_1)$) and state predicates (e.g., $\texttt{holding}(\texttt{player}, \texttt{ball})$) are syntactically represented with a name together with a list of arguments. For event types and state predicates, these arguments are all variables rather than constants. A ground event is an instantiation of an event type, meaning that the variables are replaced with constants. Similarly, a ground predicate is an instantiation of a state predicate. For example, $walk(person, loc_1, loc_2)$ is an event type, whereas $walk(Camila, Home, Office)$ is a ground event.

A (complete) state s in this framework is a full assignment of $\{true, false\}$ to all possible groundings of the predicates in F. However, at any particular time step, it is generally the case that the values of many state predicates are unknown. A belief state is a set of (complete) states that hold in a particular time step. It can also be interpreted as a conjunction of those ground predicates whose truth values are known.

3 Question Answering about Narratives

A query about the narrative inquires about a belief state (conjunction of literals) in our meaning representation language. Our question answering framework consists of three steps: collecting prior knowledge, mapping sentences to event forests, and probabilistic inference on the event forests using the prior knowledge.

3.1 Collecting Domain Knowledge

Domain knowledge provides useful information for answering semantic questions. Our framework takes advantage of two types of domain knowledge: description of few domain events and correlations among domain entities.

Event Descriptions Semantically, an event type either describes a belief state (e.g., $corner$) or deterministically maps a belief state to a new belief state (e.g., $pass(player_1, player_2)$, $putdown(person, object)$). The semantics of an event type is described with STRIPS (Fikes & Nilsson 1971) preconditions and effects. $\langle e(\vec{x}), \texttt{Precond}(\tilde{\texttt{x}}), \texttt{Effect}(\tilde{\texttt{x}}) \rangle$ is the description of the event $e(\vec{x})$ where $\texttt{Precond}(\tilde{\texttt{x}})$ and $\texttt{Effect}(\tilde{\texttt{x}})$ are conjunctions of state predicates or their negations. We use the frame assumption that the truth value of a ground predicate stays the same unless it is changed by an event. For instance, $\langle pass(player_1, player_2), \texttt{holding}(\texttt{player}_1), \texttt{holding}(\texttt{player}_2) \rangle$ describes that the event $pass$ changes the ball possession from $player_1$ to $player_2$ and $\langle run(l_2), \texttt{empty}(\texttt{l}_2), \texttt{atloc}(\texttt{l}_2) \wedge \texttt{tired} \rangle$ describes the event run that the agent runs to the empty location l_2 and becomes tired.

We use our symbolic language to describe *event types* rather than *ground events*. Describing event types stands in contrast to describing all possible state-to-state transitions. Hereinafter, for simplicity, we use the terms 'event' instead of 'ground event', and 'state' instead of 'belief state'.

These event descriptions can be constructed manually or can be extracted from VerbNet (Schuler 2005), a comprehensive verb lexicon that contains semantic information such as preconditions, effects, and arguments of about 5000 verbs. Our language includes a noise event called $Nothing$ which has no preconditions and no effects, and hence does not alter the state of the world. Events in the domain knowledge together with a noise event can cover all the remaining verbs in the domain.

Objects						Locations	
Animal		Ball		Bed		street	
wild	.64	game	.43	bedroom	.43	vehicle	.40
water	.16	hand	.33	hotel	.29	pavement	.28
forest	.05	park	.09	hospital	.16	curb	.25
cage	.05	basket	.08	store	.12	bus stop	.7

Table 1: Scores $score(l|o)$ and $P_0(o|l)$ as part of domain knowledge captured using OMCS.

Entity Correlations In general domains, like stories, adding basic knowledge considered as "common sense" would help in better understanding the narratives. Different sources of commonsense knowledge are currently available. Here, we use Open Mind Common Sense (OMCS) (Singh 2002), a database of commonsense knowledge collected through sentences from ordinary people.

For spatial contexts, we are interested in the knowledge about correlations among objects and locations. For this purpose, we use the `findin` file in OMCS, which contains a set of statements of the form, "you often find (object o) (in location l)". These statements range from stronger correlations like "(bed) in a (bedroom) " to weaker correlations like "(bed) in a (store)". In addition, many useful correlations like "(bed) in a (hospital)" are missing.

To rank these correlations we assign a score to each pair of an object o and a location l by statistically computing the frequency of the co-occurrence of the object o and the location l in a large corpus of natural language stories. The corpus that we use is "American fictions" downloaded from the Project Gutenberg.[1]. Intuitively, the object o and the location l would appear relatively close to each other in the corpus if they are correlated.

In OMCS, for every object o (1600 objects) and every location l (2675 locations) we compute the scores $score(o|l)$ and $score(l|o)$ by calculating the frequencies of occurrences of o and l, and co-occurrence of o and l in the corpus i.e., $score(o|l) = \frac{\#(o,l)}{\#l}$ and $score(l|o) = \frac{\#(o,l)}{\#o}$.

We then rank the OMCS correlations for a specific object o using the derived scores of all correlations among any location and the object o. Similarly, we rank the OMCS correlations for every location l. Since we compute the correlations among all the objects and locations we can add missing correlations if their scores are high. Table 1 displays the top four scores for some objects and locations. Our approach adds the correlation $(bed, hospital)$ to OMCS since its score is among the top four scores $score(l|o = bed)$.

3.2 Interpreting Sentences

A sentence interpretation maps a sentence to an event whose description is provided in the domain knowledge. A natural language sentence can be interpreted in different ways since the events might not be explicitly mentioned in the sentence. We assign a score to all possible events associated with a sentence and a context. The score, represented as $P(e_i|w, s)$, encodes the likelihood that an event e_i is an interpretation of a sentence w in the context s. For example in "John went to work," when the work is far, he is more likely to *drive* than to *walk*.

Figure 1 shows a part of a RoboCup narrative and a possible ranking of events based on their scores. In the soccer domain, we use the approach of (Hajishirzi *et al.* 2011) to compute the likelihood $P(e_i|w, s)$ for all ground events e_i and every sentence w and the state s of the soccer game.

In general domains (e.g., stories), it is not feasible to compute the likelihood of all the events for every sentence and every state. We use WordNet (Fellbaum 1998) to choose the possible events that can be interpretations of a sentence. We use arguments of a sentence to represent the context of the story (the world state). We then compute event likelihoods corresponding to a sentence by statistical analysis of the co-occurrence of the event together with the sentence arguments in natural language.

For every sentence w, we run a semantic role labeller (Punyakanok, Roth, & Yih 2008) to find the verb $verb$ and the roles of the verb arguments $args$, i.e., $w = \langle verb, args\rangle$. We take *hyponyms* of a verb from WordNet to form the possible event interpretations of a verb. In linguistics, a hyponym is a word that shares a *type-of* relationship with its *hypernym*; i.e., they share general properties but differ in particular points. For example, *run*, *walk*, and *drive* are all hyponyms of *go* (their *hypernym*).[2]

Now, for every sentence $w = \langle verb, args\rangle$ we aim to compute the likelihood $P(e_i|w, s)$ that the verb can be replaced by its hyponym e_i given the sentence context s (arguments $args$). We derive $P(e_i|w, s)$ by multiplying the likelihoods $P(e_i|n_j)$ of co-occurrence of the hyponym e_i and the nouns n_j in the sentence arguments $args = \langle n_1, \ldots n_k\rangle$ in the corpus: $P(e_i|w, s) \propto \prod_{n_j \in args} P(n_j|e_i) = \prod_{n_j \in args} \#(n_j, e_i)/\#e_i$. For that, we use a database (Lee 1999) consisting of (noun, verb, frequency) tuples extracted for 1000 most common nouns.

There are many cases in which the words n_j, e_i, or the pair (n_j, e_i) do not exist in the corpus. If the noun n_j does not exist in the corpus, we generalize the noun n_j by replacing it with its hypernym and computing the frequency of the new pair $(hypernym(n_j), e_i)$. For example, we replace the term *lady* in the $args$ with its hypernym *woman* since the latter appears in the corpus, but the earlier does not.

A more complicated problem is that neither the hyponym e_i nor the noun n_j appears in the corpus. In that case, we find similar nouns $Sims(n_i)$ to the noun n_j in the corpus. We then compute the weighted sum over the frequencies $\#(sim, e_i)$ where sim is one of the similar nouns to n_j. The weights are derived by computing the semantic distance $Dist(sim, n_j)$ (Lee 1999) between each similar noun $sim \in Sims(n_j)$ and n_j. Therefore,

$$P(e_i|w = \langle verb, args\rangle, s) \quad (1)$$
$$= \frac{1}{Z} \prod_{n_j \in args} \sum_{sim \in Sims(n_j)} Dist(sim, n_j) \frac{\#(sim, e_i)}{\#sim}$$

[1] See http://www.gutenberg.org/

[2] Please notice the problem of interpreting sentences is different from word sense disambiguation. We assume that the sense of the verb is already known (*go* means to change location), but the hyponym (how the agent changes the location e.g., *walk, run, drive*) is unknown.

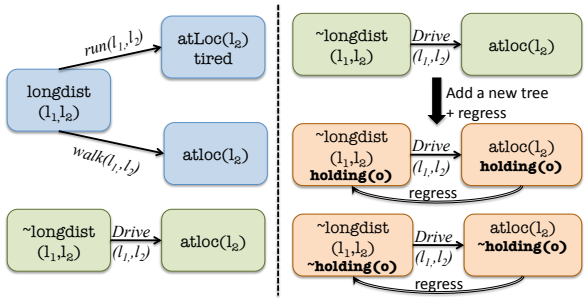

Figure 2: (*left*) Interpretation forest for $go(l_1, l_2)$ (*right*) Adding a new tree in the forest with observation of `holding(o)` at one branch at step 1.

where $Z = \sum_{sim \in Sims(n_j)} Dist(sim, n_j)$ and $\#sim$ is derived by summing over the frequencies of all the pairs (sim, v) in the corpus i.e., $\#sim = \sum_{v \in Verbs} \#(sim, v)$.

3.3 Symbolic Probabilistic Reasoning Algorithm

The reasoning algorithm takes as input a narrative $\langle w_1 \dots w_T \rangle$ and event likelihoods $P(e_i|w_t, s_{t-1})$ for each sentence w_t and the state s_{t-1} and answers a query q about the narrative. The query q inquires about the correctness of a conjunction of state predicates in our meaning representation language. Any reasoning algorithm about a symbolic probabilistic framework (e.g., (Iocchi *et al.* 2004; Reiter 2001; Hajishirzi & Amir 2007)) can be adopted to answer the query.

One approach is to (approximately) find the most likely interpretation (event sequence $\langle e_1, \dots, e_T \rangle$) corresponding to the narrative $\langle w_1 \dots w_T \rangle$ and evaluate the query in that event sequence. A simple approximation (Branavan *et al.* 2009; Chen, Kim, & Mooney 2010) is to select the most likely event corresponding to every sentence (i.e., $e_t = \arg\max_{e_i} P(e_i|w_t, s_{t-1})$. While simple, this method cannot take advantage of the coherency among the world states before and after a sentence. (Hajishirzi *et al.* 2011) use a dynamic programming approach to approximate the most likely event sequence. The utility of selecting an event e_t is based on the score of the event and the coherency of the event with the previously selected events in the sequence.

An alternative strategy is to consider all the possible interpretations of the narrative instead of a single interpretation and compute the likelihood $P(q|\langle w_1 \dots w_T \rangle)$ of the query given all the possible interpretations. For short narratives and narratives with a small branching factor, this probability can be computed exactly. For exact computation, we first build an *interpretation forest* (see Figure 2) that maintains and grows all feasible interpretations of the narrative in an online fashion. We then answer a query by integrating the results from each interpretation in the forest.

The nodes of the interpretation forest are the belief states and the edges are the events. We use belief states rather than complete states to decrease the size of the forest. To keep the consistency and provide the ability to answer queries about the past and the current world states, when a new sentence appears, we update the acquired information forward (using *Progress*) and then backward (using *Regress*).

Progress(s_{t-1}, e) takes as input an event e and the current state s_{t-1} and returns the updated state s_t if the `Precond(e)` is consistent with s_{t-1} by applying the event description e. *Regress*(s_t, e, s_{t-1}) takes as input an event e and current and previous states of the narrative and updates s_{t-1} using the preconditions of e. In this setting we specify the interpretation forest properties as follows:

Definition 1 (Interpretation Forest). An interpretation forest (with belief states as nodes and events as edges) has the following properties:

1. The labels of roots partition the set of all world states.
2. The sum of probabilities of the edges e_i leading from a node s is 1; i.e., $\sum_i P(e_i|s) = 1$.
3. For an edge e_t between nodes s_{t-1} and s_t: $s_t = Progress(s_{t-1}, e_t)$, $s_{t-1} = Regress(s_t, e_t, s_{t-1})$.
4. If a state predicate f appears in a node s, it appears at all descendants of the node unless there is an edge e leading to s and f is in `Effect(e)`.

To build the interpretation forest for the narrative, we first set the roots as belief states that are *crucial* for interpreting the first sentence w_0. The crucial state predicate is selected according to the event descriptions in the domain knowledge. For example, (Figure 2 (*left*)) the state predicate `longdist` is crucial for the the verb *go* since the preconditions of *walk* and *run* include $\neg$`longdist`, and the precondition of *drive* includes `longdist`. We then grow each interpretation by adding new edges (event $e_{i,t}$) corresponding to the sentence w_t. The new nodes s_t are derived by progressing the previous state s_{t-1} with the event $e_{i,t}$. If a new state predicate appear to be known at a branch of the forest, the algorithm replicates the current node into several nodes and *regresses* the information back to the root to maintain the properties of the forest. Figure 2 (*right*) shows regressing the observation `holding(o)` back to time step 0.

Finally, we compute $P(q|\langle w_1 \dots w_T \rangle)$ by integrating the likelihoods of all the interpretations that satisfy the query. An interpretation of the narrative, a sequence of states and events, is a path from a root to a leaf in the interpretation forest. Every interpretation $p = \langle s_0, e_1, s_1, \dots, e_T, s_T \rangle$ satisfies the query q if the state s_T models the query q (i.e., $s_T \models q$). The likelihood of an interpretation p is computed as $P(p) = P(s_0) \prod_t P(e_t|w_t, s_{t-1})$ where $P(s_0)$ is computed from the prior distribution. Finally, the probability of the query is computed by marginalizing over the probabilities of the query given every interpretation p^i in the forest.

$$P(q|\langle w_1 \dots w_T \rangle) = \sum_i P(p^i = \langle s_0^i e_1^i \dots e_T^i s_T^i \rangle) P(q|p^i)$$
$$= \sum_{i, s_T^i \models q} P(s_0^i) \prod_t P(e_t^i|w_t, s_{t-1}^i)$$

4 Experiments: Reading Comprehension

We demonstrate the effectiveness of our approach to answer semantic questions for the two domains of RoboCup soccer commentaries and early reader children's stories.

4.1 RoboCup Soccer Commentaries

The RoboCup soccer commentaries dataset (Chen, Kim, & Mooney 2010) is based on commentaries of four champi-

		1	2	3	4	Avg.
Soccer	Query I: Who has the possession of the ball?					
	Our approach	**.76**	**.70**	**.80**	**.74**	**.75**
	Baseline	.27	.27	.26	.24	.26
	Query II: Did a correct pass happen at this step?					
	Our approach	**.95**	**.93**	**.99**	**.98**	**.96**
	Baseline	.82	.50	.59	.57	.64
BadBat	Bad Bat: Where is person x at this step?					
	Baseline1		Baseline2		Our approach	
	0.39		0.56		**0.83**	

Table 2: Results of answering questions about (*top*) four RoboCup commentaries and (*bottom*) Bad Bat texts using our approach vs. baselines.

onship games of the RoboCup simulation league that took place from 2001 to 2004. Each game is associated with a sequence of human comments in English (about 2000 comments in total). The meaning representation language includes 16 events (actions with the ball), 10 state predicates (the state of the ball), and 24 constants (player names). We build event descriptions and compute the event likelihoods using the approaches in (Hajishirzi *et al.* 2011).

Our semantic queries inquire about a game property at every time step. To answer semantic questions, we find the most likely interpretation, update the game state with the selected event, and check if the query is valid in the game state. To compute the accuracy, we report the fraction of time steps for which the algorithm responds correctly.

We compare the accuracy of our algorithm with a baseline that returns a part of the text that shares similar features to the query. Our baseline is a rough implementation of a traditional reading comprehension system (Rilo & Thelen 2000). We report the results of both approaches for each game on two questions in Table 2 (*top*). Unlike our approach, the feature matching baseline can neither infer the difference between *successful* and *unsuccessful* passes nor reason about the position of the ball at every time step.

4.2 Children Stories

We apply our framework to a less structured domain, early reader children's stories (good night stories[3] and Bad Bat[4]) focused on spatial contexts. The collection includes 10 stories, with the total of about 200 sentences. We chose early reader stories since they do not require too much syntactical preprocessing. In this domain, the meaning representation language consists of 25 constants, 10 state predicates, and 15 events (verbs extracted from 1000 most common nouns). We build the domain knowledge by manually describing events and extracting object-location correlations (Section 3.1).

In the Bad Bat domain, the queries are "where" questions inquiring about the location of entities in the stories (i.e., `Camila`, `BadBat`, `Molly`, and `Hall`) at different time steps. In Table 2 (*bottom*) we report the fraction of time steps for which the location is inferred correctly using our approach and two baselines. (Baseline1) is a feature matching approach that looks for a sentence that contains the entity

[3]http://www.goodnightstories.com/

[4]http://www.stuartstories.com/badbat/

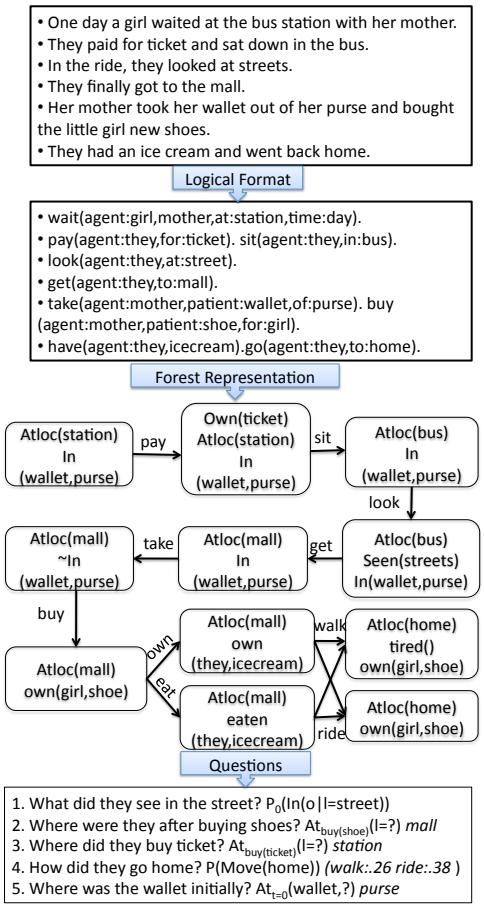

Figure 3: (a) sample story (b) symbolic representation corresponding to the story (c) interpretation forest (d) semantic questions about the story and responses.

together with a location. (Baseline2) performs more intelligent feature matching and chooses the closest location that appeared in the story if a sentence including the entity and a location cannot be found. As the results suggest, syntactical analysis (even Baseline2) cannot infer the location given a sentence like "Camila left the room." Only by reasoning (backing up the information) can our algorithm infer that `Camila` was in the room earlier in the story.

We now show the application of every step of our framework in a sample story (Figure 3(a)).

Sentence Interpretation We use WordNet together with our statistical analysis (Section 3.2) to compute the event likelihoods for every sentence (see Table 3 for hyponym likelihoods of three verbs given a specific argument). We verify the computed event ranks by testing in two semantic corpora, SemCor (Miller *et al.* 1993) (about 15000 sentences) and SensEval-2 (Mihalcea & Moldovan 2002) (about 100 sentences). These corpora are semantically annotated with reference to the WordNet lexical database. Unfortunately, no annotation is available to test the sentence interpretations according to the verb hyponyms. For evaluation, we change the test corpora as follows: we replace every verb of the

	verb, args					
	memorize, pattern		make, tea		go, home	
event,p	study	.35	cook	.42	drive	.38
	review	.22	make	.37	walk	.28
	absorb	.18	throw	.16	run	.36
	memorize	.17	dip	.05		

Table 3: $P(e|\langle v, args\rangle)$ of the hyponyms e for the sentences $w = \langle verb, args\rangle$ in the SemCor corpus and stories domain.

	SemCor Corpus		SensEval Corpus	
	Baseline	Ours	Baseline	Ours
top 1	0.25	**0.43**	0.14	**0.24**
top 2	0.6	**0.74**	0.43	**0.52**
top 3	0.7	**0.78**	0.59	**0.66**
top 4	0.8	**0.89**	0.68	**0.75**
top 5	0.89	**0.92**	0.77	**0.82**

Table 4: Rank of the original verb of the sentence in top k returned events for a sentence in SemCor and SensEval Corpora.

sentence with a more general verb (its hypernym). In the derived corpus, we compute the likelihood of all the hyponyms of the new verb using our statistical analysis (Section 3.2). For evaluation, we examine if the original verb is among the top k events reported for the new verb. The baseline ranks the events (hyponyms) according to their frequencies in WordNet independent of the arguments of the sentence. Our approach consistenly performs better (Table 4).

Question answering: Now that the event likelihoods are available, we generate the interpretation forest (see Figure 3(c)) and then enumerate all the interpretations to answer the questions. Here, we show different categories of questions that cannot be answered using feature matching question answering techniques. First category of questions (questions 1, 2, 3, and 5) inquire about the object locations that are not explicitly mentioned in the text. Our method infers the corresponding locations using the interpretation forest and the entity correlations in the domain knowledge. The second category of questions (question 4) inquire about the likelihood of the actual event that happened. This can be infered using our sentence interpretation technique.

5 Conclusions and Future Work

We introduced a framework to answer questions whose responses cannot be located in the texta and that require causal reasoning and domain knowledge to answer. To this end, we first collect domain knowledge in terms of event descriptions and entity correlations. We then derive different interpretations of a narrative by computing the likelihood of every event corresponding to a sentence. We then apply a probabilistic reasoning algorithm and answer questions. Our experiments demonstrate the power of this framework for answering questions about RoboCup soccer commentaries and children's stories. The advantage of our system is a flexible structure in which each part can be replaced with a more powerful or domain-specific algorithm. Our framework currently can only handle simple sentences since it does not include complex and rich NLP tools.

Our immediate future work is to augment our system with rich NLP tools and apply it to the Remedia corpus for reading comprehension. Also, we plan to go beyond verbs and generalize our method to understanding other parts of the sentence (Poon & Domingos 2009).

Acknowledgements

We thank Eyal Amir and Julia Hockenmaier for their insightful comments on this paper. We would also like to thank the anonymous reviewers for their helpful comments on improving the current paper.

References

Bejan, C. 2008. Unsupervised discovery of event scenarios from texts. In *FLAIRS*.

Branavan, S.; Chen, H.; Zettlemoyer, L.; and Barzilay, R. 2009. Reinforcement learning for mapping instructions to actions. In *ACL-IJCNLP*, 82–90.

Chen, D.; Kim, J.; and Mooney, R. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *JAIR* 37:397–435.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Fikes, R., and Nilsson, N. J. 1971. STRIPS: a new approach to the application of theorem proving to problem solving. *AIJ* 2:189–208.

Hajishirzi, H., and Amir, E. 2007. Stochastic filtering in a probabilistic action model. In *AAAI*.

Hajishirzi, H.; Hockenmaier, J.; Mueller, E.; and Amir, E. 2011. Reasoning about robocup soccer narratives. In *UAI*.

Hobbs, J. R.; Stickel, M. E.; Appelt, D. E.; and Martin, P. 1993. Interpretation as abduction. *Artificial Intelligence* 63:69–142.

Iocchi, L.; Lukasiewicz, T.; Nardi, D.; and Rosati, R. 2004. Reasoning about actions with sensing under qualitative and probabilistic uncertainty. *ECAI*.

Lee, L. 1999. Measures of distributional similarity. In *ACL*.

Liang, P.; Jordan, M. I.; and Klein, D. 2009. Learning semantic correspondences with less supervision. In *ACL-IJCNLP*.

Mihalcea, R., and Moldovan, D. 2002. Pattern learning and active feature selection for word sense disambiguation. In *ACL workshop*, 127130.

Miller, G.; Leacock, C.; Randee, T.; and Bunker, R. 1993. A semantic concordance. In *DARPA workshop on HLT*, 303–308.

Narayanan, S., and Harabagiu, S. 2004. Question answering based on semantic structures. In *Coling*, 693–701.

Poon, H., and Domingos, P. 2009. Unsupervised semantic parsing. In *EMNLP*.

Punyakanok, V.; Roth, D.; and Yih, W. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics* 34(2).

Reiter, R. 2001. *Knowledge In Action*. MIT Press.

Rilo, E., and Thelen, M. 2000. A rule-based question answering system for reading comprehension tests. In *In ANLP/NAACL Workshop*.

Schuler, K. K. 2005. *VerbNet: a broad-coverage, comprehensive verb lexicon*. Ph.D. Dissertation.

Singh, P. 2002. The public acquisition of commonsense knowledge. In *AAAI Spring Symposimum*.

Vogel, A., and Jurafsky, D. 2010. Learning to follow navigational directions. In *ACL*.