Understanding Complex Multi-sentence Entity seeking Questions

Danish Contractor[1,2*], **Barun Patra**[1†], **Mausam**[1], **Parag Singla**[1]
[1]Indian Institute of Technology, New Delhi [2]IBM Research AI, New Delhi
dcontrac@in.ibm.com,barunpatra95@gmail.com, mausam,parags@cse.iitd.ac.in

Abstract

We present the novel task of understanding multi-sentence *entity-seeking* questions (MSEQs) i.e, questions that may be expressed in multiple sentences, and that expect one or more entities as an answer. We formulate the problem of understanding MSEQs as a semantic labeling task over an open representation that makes minimal assumptions about schema or ontology specific semantic vocabulary. At the core of our model, we use a BiDiLSTM (bi-directional LSTM) CRF and to overcome the challenges of operating with low training data, we supplement it by using hand-designed features, as well as hard and soft constraints spanning multiple sentences. We find that this results in a 6-7pt gain over a vanilla BiDiLSTM CRF. We demonstrate the strengths of our work using the novel task of answering real-world entity-seeking questions from the tourism domain. The use of our labels helps answer 53% more questions with 42 % more accuracy as compared to baselines.

Introduction

We introduce the novel task of understanding multi-sentence questions. Specifically, we focus our attention on multi-sentence *entity-seeking* questions (MSEQs) i.e., questions that expect one or more entities as answer. Such questions are commonly found in online forums, blog posts, discussion boards etc and come from a variety of domains including tourism, books and consumer products.

Figure 1 shows an example MSEQ from a tourism forum, where the user is interested in finding a hotel that satisfies some constraints and preferences; an *answer* to this question is thus the name of a hotel (entity) which needs to satisfy some properties such as being a 'budget' option. A preliminary analysis of such entity-seeking questions from online forums reveals that almost all of them contain multiple sentences – they often elaborate on a user's specific situation before asking the actual question.

In order to *understand* and answer such a user question, we convert the question into a machine representation consisting of labels identifying the *informative* portions in a question. We are motivated by our work's applicability to a wide variety of domains and therefore choose not to restrict the representation to use a domain-specific vocabulary. Instead, we design an *open* semantic representation, inspired in part by Open QA (Fader, Zettlemoyer, and Etzioni 2014), in which we explicitly annotate the answer (entity) type; other answer attributes, while identified, are not further categorized. Eg. in Figure 1 'place to stay' is labeled as $entity.type$ while 'budget' is labeled as an $entity.attr$. We also allow attributes of the *user* to be represented. Domain specific annotations such as $location$ for tourism questions are permitted. Such labels are then be supplied to a downstream information retrieval (IR) or a QA component to directly present an answer entity.

We pose the task of understanding MSEQs as a semantic labeling (shallow parsing)[1] task where tokens from the question are annotated with a semantic label from our open representation. However, in contrast to related literature on semantic role labeling (Yang and Mitchell 2017), slot filling tasks (Bapna et al. 2017) and query formulation (Wang and Nyberg 2016; Vtyurina and Clarke 2016; Nogueira and Cho 2017), semantic parsing of MSEQs raise several novel challenges. MSEQs express a wide variety of intents and requirements which span across multiple sentences, requiring the model to capture within-sentence as well as inter-sentence interactions effectively. In addition, questions can be unnecessarily belabored requiring the system to reason about what is important and what is not. Lastly, we find that generating training data for parsing MSEQs is hard due to the complex nature of the task, further requiring the models to operate in low training data settings.

In order to address these challenges and label MSEQs, we use a bi-directional LSTM CRF (BiDiLSTM CRF) (Huang, Xu, and Yu 2015) and extend it in two ways: (i) We encode knowledge by incorporating hand-designed features as well as semantic constraints over the entire multi-sentence question during end-to-end training. (ii) We use partially labeled questions, that are easier to source, to improve training.

In summary, our paper makes the following contributions:

1. We present the novel task of understanding multi-sentence entity-seeking questions (MSEQs). We define *open* se-

*This work was carried out as part of PhD research at IIT Delhi. The author is also a regular employee at IBM Research AI.

†Work carried out when Barun was an undergraduate student at IIT Delhi.

[1]We use the phrases 'semantic labeling' and 'semantic parsing' interchangeably in this paper.

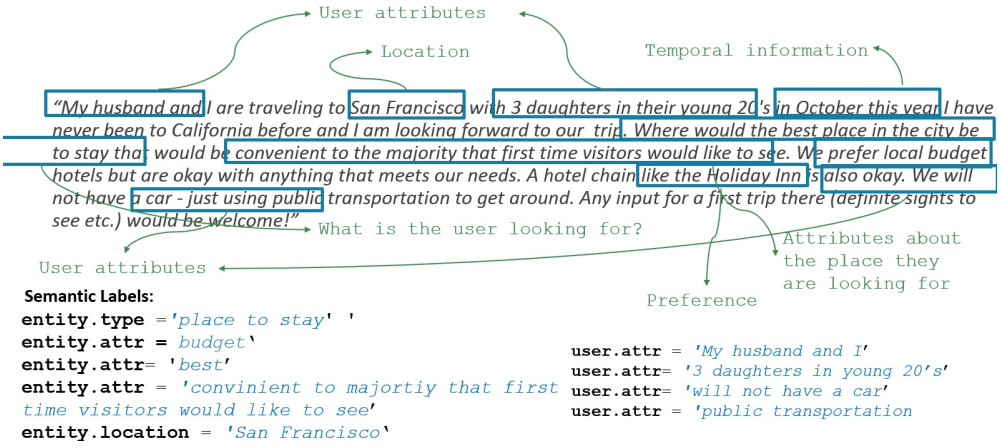

Figure 1: An entity-seeking MSEQ and annotated with our semantic labels

mantic labels, which minimize schema or ontology specific semantic vocabulary and can easily generalize across domains. These semantic labels identify *informative* portions of a question that can be used by a downstream answering component.

2. The core of our model uses a BiDiLSTM CRF model. We extend this by providing hand-designed features and using Constrained Conditional Model (CCM) (Chang, Ratinov, and Roth 2007) inference, which allows us to specify within-sentence as well as inter-sentence (hard and soft) constraints, encoding prior knowledge about the labeling task.
3. We present detailed experiments on our models using the tourism domain as an example. We also demonstrate how crowd-sourced partially labeled questions can be effectively used in our constraint based tagging framework to help improve labeling accuracy. We find that our best model achieves 7 pt improvement in F1 scores over a baseline BiDiLSTM CRF.
4. We demonstrate the applicability of our semantic labels in an end-QA task: the novel task of directly answering tourism-related MSEQs using a web based semi-structured knowledge source. Our semantic labels help formulate a more effective query to knowledge sources and our system answers 53% more questions with 42 % more accuracy as compared to baselines

Related Work

To the best of our knowledge, we are the first to explicitly address the task of *understanding* multi-sentence entity-seeking questions and demonstrate its use in an answering task.

Question Answering Systems

There are two common approaches for QA systems – joint and pipelined, both with different advantages. The joint systems usually train an end-to-end neural architecture, with a softmax over candidate answers (or spans over a given passage) as the final layer (Iyyer et al. 2014; Rajpurkar et al. 2016), while a pipelined approach (Fader, Zettlemoyer, and Etzioni 2014; Berant and Liang 2014; Fader, Zettlemoyer, and Etzioni 2013; Kwiatkowski et al. 2013; Vtyurina and Clarke 2016; Wang and Nyberg 2016) divides the task into two components – question processing (understanding) and querying the knowledge source. Our work follows the second approach.

In this paper, we return entity answers to multi-sentence entity seeking questions. The problem of returning direct, (non-document/passage) answers to questions from background knowledge sources has been studied, but primarily for single sentence factoid-like questions (Fader, Zettlemoyer, and Etzioni 2014; Berant and Liang 2014; Yin et al. 2015; Sun et al. 2015; Saha et al. 2016; Khot, Sabharwal, and Clark 2017; Lukovnikov et al. 2017). Reading comprehension tasks (Rajpurkar et al. 2016; Trischler et al. 2016; Joshi et al. 2017; Trivedi et al. 2017) require answers to be generated from unstructured text also only return answers for simple single-sentence questions.

Other works have considered multi-sentence questions, but in different settings, such as the specialized setting of answering multiple-choice SAT and science questions (Seo et al. 2015; Clark et al. 2016; Khot, Sabharwal, and Clark 2017; Guo et al. 2017), mathematical word problems (Liang et al. 2016), and textbook questions (Sachan, Dubey, and Xing 2016). Such systems do not return entity answers to questions. Community QA systems (Bogdanova and Foster 2016; Shen et al. 2015; Qiu and Huang 2015; Tan, Xiang, and Zhou 2015) match questions with *user*-provided answers, instead of entities from background knowledge-source. IR-based systems (Vtyurina and Clarke 2016; Wang and Nyberg 2016) query the Web for open-domain questions, but return long (1000 character) passages as answers; they haven't been developed for, or tested on entity-seeking questions. These techniques that can handle MSEQs (Vtyurina and Clarke 2016; Wang and Nyberg 2016) typically perform retrieval using keywords extracted from questions; these do not "understand" the questions and can't answer many tourism questions, as our experiments

show. The more traditional solutions (e.g., semantic parsing) that parse the questions deeply can process only *single*-sentence questions (Fader, Zettlemoyer, and Etzioni 2014; Berant and Liang 2014; Fader, Zettlemoyer, and Etzioni 2013; Kwiatkowski et al. 2013).

Finally, systems such as QANTA (Iyyer et al. 2014) also answer complex multi-sentence questions but their methods can only select answers from a small list of entities and also require large amounts of training data with redundancy of QA pairs. In contrast, the subset of Google Places we experiment with has close to half a million entities.

We discuss literature on parsing (understanding) questions in the next section.

Question Parsing

QA systems use a variety of different intermediate semantic representations. Most of them, including the rich body of work in NLIDB and semantic parsing, parse *single* sentence questions into a query based on the underlying ontology or DB schema and are often learned directly by defining grammars, rules and templates (Zettlemoyer 2009; Liang 2011; Kwiatkowski et al. 2013; Berant et al. 2013; Yih et al. 2015; Sun et al. 2015; Saha et al. 2016; Reddy et al. 2016; Khot, Sabharwal, and Clark 2017; Cheng et al. 2017; Lukovnikov et al. 2017). Work such as (Fader, Zettlemoyer, and Etzioni 2014; Berant and Liang 2014) build *open* semantic representations for single sentence questions, that are not tied to a specific knowledge source or ontology. We follow a similar approach and develop an open semantic representation for multi-sentence entity seeking questions. Our representation uses labels that help a downstream answering component return entity answers.

Recent works build neural models that represent a question as a continuous-valued vector (Bordes, Chopra, and Weston 2014; Bordes, Weston, and Usunier 2014; Xu et al. 2016; Chen et al. 2016; Zhang et al. 2016) but such methods require significant amounts of training data. Some systems rely on IR and do not construct explicit semantic representations at all (Sun et al. 2015; Vtyurina and Clarke 2016); they rely on selecting keywords from the question for querying and as shown in our experiments do not perform well for answering multi-sentence entity-seeking questions. Work such as that by (Nogueira and Cho 2017) uses reinforcement learning to select query terms in a document retrieval task and requires a large collection of document-relevance judgments. Extending such an approach for our task could be an interesting extension for future work.

We now summarize recent methods employed to generate semantic representations of questions.

Neural Semantic Parsing

There is a large body of literature dealing with semantic parsing of single sentences, especially for frames in PropBank and Framenet (Palmer, Gildea, and Kingsbury 2005; Baker, Fillmore, and Lowe 1998). Most recently, methods that use neural architectures for SRL (Semantic Role Labeling) have been developed. For instance, work by (Zhou and Xu 2015) uses a BiDiLSTM CRF for labeling sentences with PropBank predicate argument structures, while work by (He et al. 2017; 2018) relies on a BiDiLSTM with BIO-encoding constraints during LSTM decoding. Other recent work by tomemnlp2017 proposes a BiDiLSTM CRF model that is further used in a graphical model that encodes SRL structural constraints as factors. Work such as that by (Bapna et al. 2017) uses a BiDiLSTM tagger for predicting task oriented information slots from sentences. Our work uses similar approaches for parsing MSEQs, but we note that such systems cannot be directly used in our task due to their model specific optimizations for their label space. However, we adapt the label space of the recent Deep SRL system (He et al. 2017) for our task and use its predicate tagger as a baseline for evaluation.

Problem Statement

Given a multi-sentence entity seeking question, our goal is to first parse and generate a semantic representation of the question. These semantic labels identify *informative* portions of a question that can be used by a downstream answering component. The semantic representation of the question is then used to return an entity answer for the question using a knowledge source.

Semantic Labels for MSEQs

As mentioned earlier, our question understanding component parses an MSEQ into an *open* semantic representation. Our choice of representation is motivated by two goals. First, we wish to make minimal assumptions about the domain of the QA task and therefore, minimize domain-specific semantic vocabulary[2]. Second, we wish to identify only the *informative* elements of a question, so that a robust downstream QA or IR system can meaningfully answer it. As a first step towards a generic representation for an MSEQ, we make the assumptions that a multi-sentence question is asking only one final question, and that the expected answer is one or more entities. This precludes Boolean, comparison, 'why'/'how', and multiple part questions

We have two labels associated with the entity being sought: *entity.type* and *entity.attr*, to capture the type and the attributes of the entity, respectively. We also include a label *user.attr* to capture the properties of the user asking the question. The semantic labels of $entity.type$ and $entity.attr$ are generic and will be applicable to any domain. Other generic labels to identify related entities (eg: in questions where users ask for entities similar to a list of entities) could also be defined. We also allow the possibility of incorporating additional labels which are domain specific. For instance, for the tourism domain, location could be important, so we can include an additional label $entity.location$ describing the location of the answer entity.

Figure 1 illustrates the choice of our labels with an example from the tourism domain. Here, the user is interested in finding a 'place to stay' ($entity.type$) that satisfies some properties such as 'budget' ($entity.attr$). The question includes some information about the user herself e.g., 'will not have a car' which may become relevant for answering

[2]Our representation can easily be generalized to include domain-specific semantic labels, if required.

the question. The phrase 'San Francisco' describes the location of the entity and is labeled with a domain specific label ($entity.location$).

MSEQ Semantic Parsing

We formulate the task of outputting the semantic representation for a user question as a sequence labeling problem. There is a one to one correspondence between our token-level label set and the semantic labels described in earlier. We utilize a BiDiLSTM CRF (Huang, Xu, and Yu 2015) for sequence labeling and as described previously, we extend the model in order to address the challenges posed by MSEQs: (a) First, we incorporate hand-engineered features especially designed for our labeling task (b) Second, we make use of a Constrained Conditional Model (CCM) (Chang, Ratinov, and Roth 2007) to incorporate within-sentence as well as inter-sentence constraints. These constraints act as a prior and help ameliorate the problems posed by our low-data setting. (c) Third, we use Amazon Mechanical Turk (AMT) to obtain additional partially labeled data which we use in our constraint driven framework.

Features

We incorporate a number of (domain-independent) features into our BiDiLSTM CRF model where each unique feature is represented as a one-hot vector and concatenated with the word-vector representation of each token.

Our features are described as follows: (a) Lexical features for capitalization, indicating numerals etc., token-level features based on POS and NER (b) hand-designed $entity.type$ and $entity.attr$ specific features. These include indicators for guessing potential types, based on targets of WH (*what*, *where*, *which*) words and certain verb classes; multi-sentence features that are based on dependency parses of individual sentences that aid in attribute detection, e.g., for every noun and adjective, an attribute indicator feature is on if any of its ancestors is a potential $type$ as indicated by type feature; indicator features for descriptive phrases (Contractor, Mausam, and Singla 2016), such as adjective-noun pairs. (c) For each token, we include cluster ids generated from a clustering of word2vec vectors (Mikolov et al. 2013) run over a large tourism corpus. (d) We also use the counts of a token in the entire post, as a feature for that token (Vtyurina and Clarke 2016).

Constraints

Since we label multiple-sentence questions, we need to capture patterns spanning across sentences. One alternative would be to model these patterns as features defined over non-adjacent tokens (labels). But this can make the modeling quite complex. Instead, we model them as global constraints over the set of possible labels.

We design the following constraints: (i) type constraint (hard): every question must have at least one $entity.type$ token, and (ii) attribute constraint (soft), which penalizes absence of an $entity.attr$ label in the sequence. (iii) a soft constraint that prefers all $entity.type$ tokens occur in the same sentence. The last constraint helps reduce erroneous $entity.type$ labels but allows the labeler to choose $entity.type$-labeled tokens from multiple sentences only if it is very confident. Thus, while the first two constraints are directed towards improving recall, the last constraint helps improve precision of $entity.type$ labels

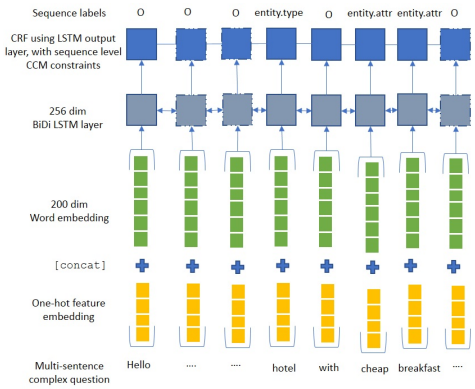

Figure 2: BiDi LSTM CCM for sequence labeling.

In order to use our constraints, we employ Constrained Conditional Models (CCMs) for our task (Chang, Ratinov, and Roth 2007) which use an alternate learning objective expressed as the difference between the original log-likelihood and a constraint violation penalty:

$$\sum_i w^T \phi(\mathbf{x^{(i)}}, \mathbf{y^{(i)}}) - \sum_i \sum_k \rho_k d_{C_k}(\mathbf{x^{(i)}}, \mathbf{y^{(i)}}) \quad (1)$$

Here, i indexes over all examples and k over all constraints. $\mathbf{x^{(i)}}$ is the i^{th} sequence and $\mathbf{y^{(i)}}$ its labeling. ϕ and w are feature and weight vectors respectively. d_{C_k} and ρ_k denote the violation score and weight associated with k^{th} constraint. The w parameters are learned analogous to a vanilla CRF and computing ρ parameters resorts to counting. Hard constraints have an infinite weight. Inference in CCMs is formulated as an Integer Linear Program (ILP); see Chang et al.(2007) for details. The original CCM formulation was in the context of regular CRFs (Lafferty, McCallum, and Pereira 2001) and and we extend its use in a combined model of BiDiLSTM CRF with CCM constraints that is trained end-to-end (Figure 2).

Partially Labeled Data

Data Collection: In order to obtain a larger amount of labeled data for our task, we make use of crowd-sourcing (Amazon Mechanical Turk). Since our labeling task can be fairly complex, we divide our crowd task into multiple steps. We first ask crowd to (i) filter out forum questions that are not entity-seeking questions. For the questions that remain, the crowd provides (ii) $user.*$ labels, and (iii) $entity.*$ labels. Taking inspiration from (He, Lewis, and Zettlemoyer 2015), for each step, instead of directly asking for token labels, we ask a series of indirect questions as described in the next section that can help source high precision annotations.

We obtain two sets of labels (different workers) on each question. However, due to the complex nature of the task we

	type	*attr*	*loc*
Avg. token level agreement	47.98	37.78	68.56

Table 1: Agreement for $entity$ labels on AMT

find that workers are not complete in their labeling and we therefore only use token labels where both the set of workers agreed on labels. Thus we are able to source annotations with high precision, while recall can be low. Table 1 shows token-level agreement statistics for labels collected over a set of 400 MSEQs from the tourism domain. Some of the disagreement arises from labeling errors due to complex nature of the task. In other cases, the disagreement results from their choosing one of the several possible correct answers. E.g., in the phrase "*good restaurant for dinner*" one worker labels $entity.type$ ='restaurant', $entity.attr$ ='good' and $entity.attr$ ='dinner', while another worker simply chooses the entire phrase as $entity.type$.

Training with partially labeled posts We devise a novel method to use this partially labeled data, along with our small training set of expert labeled data, to learn the parameters of our CCM model. We utilize a modified version of Constraints driven learning (CODL) (Chang, Ratinov, and Roth 2007) which uses a semi-supervised iterative weight update algorithm, where the weights at each step are computed using a combination of the models learned on the labeled and the unlabeled set (Chang, Ratinov, and Roth 2007).

Given a dataset consisting of a few fully labeled as well as unlabeled examples, the CoDL learning algorithm first learns a model using only the labeled subset. This model is then used to find labels (in a hard manner) for the unlabeled examples while taking care of constraints. A new model is then learned on this newly annotated set and is combined with the model learned on the labeled set in a linear manner. The parameter update can be described as:

$$(w^{(t+1)}, \rho^{(t+1)}) = \gamma(w^{(0)}, \rho^{(0)}) + (1-\gamma)\mathrm{Learn}(\mathrm{U}^{(t)}) \quad (2)$$

Here, t denotes the iteration number, $U^{(t)}$ denotes the unlabeled examples and Learn is a function that learns the parameters of the model. In our setting, Learn trains the neural network via back-propagation. Instead of using unlabeled examples in $U^{(t)}$ we utilize the partially labeled set whose values have been filled in using parameters at iteration t and, inference over the set involves predicting only the missing labels. This is done using the ILP based formulation described previously and γ controls the relative importance of the labeled and partial examples. To the best our knowledge, we are the first to exploit partial supervision from a crowd-sourcing platform in this manner.

Experimental Evaluation

The goal of our experimental evaluation was to analyze the effectiveness of our proposed model for the task of understanding MSEQs. We next describe our dataset, evaluation methodology and our results in detail.

Dataset

For our current evaluation, we used the following three semantic labels: $entity.type$, $entity.attr$, $entity.location$. We also used a default label $other$ to mark any tokens not matching any of the semantic labels.

We use 150 expert-annotated tourism forum questions (9200 annotated tokens) as our labeled dataset and perform leave-one out cross-validation. This set was labeled by two experts, including one of the authors, with high agreement. For experiments with partially labeled learning, we add 400 partially-annotated questions from crowd-sourced workers to our training set. As described previously, each question is annotated by two workers and we retain token labels marked the same by two workers, while treating the other labels as unknown. We still compute a leave one out cross-validation on our original 150 expert-annotated questions (complete crowd data is included in each training fold).

Methodology

Sequence-tagged tokens identify *phrases* for each semantic label – so, instead of reporting metrics at the token level, we compute a more meaningful joint metric over tagged phrases. We define a matching-based metric that first matches each extracted segment with the closest one in the gold set, and then computes segment level precision using constituent tokens. Analogously, recall is computed by matching each segment in gold set with the best one in extracted set. As an example, for Figure 1, if the system extracts "convenient to the majority" and "local budget" for $entity.attr$ then our matching-metric will compute precision as 0.75 (1.0 for "convenient to the majority" and 0.5 for "local budget)" and recall as 0.45 (1.0 for "budget", 0.0 for "best" and 0.364 for "convenient to the majority ... like to see").

We use the Mallet toolkit[3] for CRF implementation and the GLPK ILP-based solver[4] for CCM inference. In the case of BiDiLSTM based CRF, the BiDiLSTM network at each time step feeds into a linear chain CRF layer. The input states in the LSTM are modeled using a 200 dimension word vector representation of the token. These word vector representations were with pre-trained using the word2vec model(Mikolov et al. 2013) on a large collection of $80,000$ tourism questions. For CoDL learning we set γ to 0.9 as per original authors' recommendations.

Results

Table 2 reports the performance of our semantic labeler under different configurations. We find that a BiDiLSTM CRF (lower half of the table) approach outperforms a CRF system (upper half of the table) in each comparable setting - for instance, using a baseline vanilla CRF based system using all features gives us an aggregate F1 of 50.8 while the the performance of a BiDi LSTM CRF approach using features is 56.2. As a baseline we use the predicate tagger from the Deep SRL system (He et al. 2017) to utilize our label space and we find that it performs similar to our CRF setup. Our

[3] http://mallet.cs.umass.edu/

[4] https://www.gnu.org/software/glpk/

Model	F1 (entity.type)	F1 (entity.attr)	F1 (entity.loc)	F1 (aggr)
Deep SRL (He et al. 2017)	48.4	47.8	53.2	49.8
CRF (all features)	51.4	45.3	55.7	50.8
CCM	59.6	50.0	56.1	55.2
CCM (with all crowd data)	55.1	42.2	46.7	48.0
PS CCM	58.5	**50.6**	60.3	56.5
BiDi LSTM CRF	53.3	47.6	52.1	51.0
BiDi LSTM CRF+Feat	58.4	48.1	62.0	56.2
BiDi LSTM CCM+Feat	59.4	49.8	**62.3**	57.2
PS BiDi LSTM CCM+Feat	**62.9**	50.4	61.5	**58.3**

Table 2: Sequence tagger $F1$ scores using CRF with all features (feat), CCM with all features & constraints, and partially-supervised CCM over partially labeled crowd data. The second set of results mirror these settings using a bi-directional LSTM CRF. Results are statistically significant (paired t-test,p value$<$0.000124 for aggregate $F1$). Models with "PS" as a prefix use partial supervision.

Algorithm	Prec	Recall	F1
CRF (all features)	66.9	41.7	51.4
CCM (all features)	62.1	57.2	59.6
BiDI LSTM CRF with Features	54.1	63.6	58.4
BiDi LSTM CCM with Features	55.1	64.5	59.4

Table 3: (i) Precision and Recall of *entity.type* with and without CCM inference.

best model combines a BiDiLSTM CRF with hand-designed features, CCM constraints along with learning from partially annotated crowd data. This model has nearly a 7 pt gain over the baseline BiDiLSTM CRF model. Further, usage of hand-curated features, within-sentence and cross-sentence constraints as well as partial supervision, each help successively improve the results. Next, we study the effect of each of these enhancements in detail.

Effect of features In an ablation study performed to learn the incremental importance of each feature, we find that descriptive phrases, and our hand-constructed multi-sentence type and attribute indicators improve the performance of each label by 2-3 points. Word2vec features help type detection because $entity.type$ labels often occur in similar contexts, leading to informative vectors for typical type words. Frequency of non stopword words in the multi-sentence post are an indicator of the word's relative importance, and the feature also helps improves overall performance.

Effect of constraints A closer inspection of Table 2 reveals that the vanilla CRF configuration sees more benefit in using our CCM constraints as compared to the BiDiLSTM CRF based setup (4pt vs 1pt). To understand why, we study the detailed precision-recall characteristics of individual labels; the results for $entity.type$ are reported in Table 3. We find that the BiDiLSTM CRF based setup has significantly higher recall than its equivalent vanilla CRF counterpart while the opposite trend is observed for precision. As a result, since two of the three constraints employed by us in CCM are oriented towards improving recall[5], we find that they improve overall F1 more by finding tags that were otherwise of lower probability (i.e. improving recall).

Effect of partial-supervision In order to further understand the effect of partial-supervision, we trained a CCM based model that makes use of *all* the crowd-sourced labels for training, by adding conflicting labels for a question as two independent training data points. As can be seen, using the entire noisy crowd-labeled sequences (row labeled "CCM (with all crowd data)" in upper half of Table 2) hurts the performance significantly resulting in an aggregate $F1$ of just 48.0 while the corresponding partially-supervised CCM system trained using partially labeled data has an $F1$ of 56.5.

Overall: Our results demonstrate that the use of each of hand-engineering features, within-sentence and inter-sentence constraints and use of partially labeled data help improve the accuracy of labeling MSEQs.

MSEQ-QA

We now demonstrate the usefulness of our MSEQ semantic labels and tagging framework by enabling a QA end task which returns entity answers for multi-sentence MSEQs. To the best of our knowledge we are the first to attempt such a QA task.

We use our sequence tagger described previously to generate the semantic labels of the questions. These semantic labels and their targets are used to formulate a query to the Google Places collection, which serves as our knowledge source.[6]. The Google places collection contains details about eateries, attractions, hotels and other points of interests from all over the world, along with reviews and ratings from users. It exposes an end point that can be used to execute free text queries and it returns entities as results.

We convert the semantic-labels tagged phrases into a Google Places query via the transformation: "concat($entity.attr$) $entity.type$ in $entity.location$". Here, concat lists all attributes in a space-separated fashion. Since some of the attributes may be negated in the original question, we filter out these attributes and do not include it as part of the query for Google Places.

Detection of Negations: We use a list of *triggers* that indicate negation. We start with a manually curated set of seed words, and expand it using synonym and antonym counter fitted word vectors (Mrksic et al. 2016). The resulting set of *trigger* words flag the presence of a negation in a sentence. We also define the scope of a negation trigger as a token (or a set of continuous tokens with the same label) labeled by our sequence tagger that occur within a specified window of the trigger word. Table reports the accuracy of our negation rules as evaluated by an author. The 'Gold' columns denote the performance when using gold semantic label mentions. The 'System' columns are the performance when using labels generated by our sequence tagger.

[5]Recall that we require at least one *entity.type* (hard constraint) and prefer at least one *entity.attr* (soft constraint)

[6]*https://developers.google.com/places/web-service/*

	Gold			System		
	P	R	F1	P	R	F1
Negations	86	66	74.6	85	62	71.7

Table 4: Performance of negation detection using gold sequence labels, and system generated labels

System	Acc@3 (%)	MRR	Recall (%)
WebQA	18.8	0.16	40
WebQA (manual)	40.2	0.37	31.2
MSEQ-QA	**56.9**	**0.47**	**47.8**

Table 5: QA task results using the Google Places web API as knowledge source.

Baseline Since there are no baselines for this task, we adapt and re-implement a recent complex QA system (called WebQA) originally meant for finding appropriate Google results (documents) to questions posed in user forums (Vtyurina and Clarke 2016). WebQA first short-lists a set of top 10 words in the question using a tf-idf based scheme computed over the set of all questions. A supervised method is then used to further, shortlist 3-4 words from that form the query. For best performance, instead of using supervised learning for further shortlisting keywords (as in the original paper), in our implementation an expert chooses 3-4 best words manually. This query on executed against the Google places collection returns answer entities instead of documents.

We randomly select 300 new unseen questions (different from the questions used in the previous section), from a tourism forum website and manually remove 105 of those that were not entity-seeking. The remaining 195 questions forms our test set. Our annotators manually check each entity-answer returned by the systems for correctness. Inter-annotator agreement for relevance of answers measured on 1300+ entities from 100 questions was 0.79. Evaluating whether an entity answer returned is correct is subjective and time consuming. For each entity answer returned, annotators need to manually query a web-search engine to evaluate whether an entity returned by the system adequately matches the requirements of the user posting the question. Given the subjective and time consuming nature of this task, we believe 0.79 is an adequate level of agreement on entity answers.

MSEQ-QA: Results Results: Table 5 reports Accuracy@3, which gives credit if any one of the top three answers is a correct answer. We also report Mean Reciprocal Rank (MRR). Both of these measures are computed only on the subset of attempted questions (any answer returned). Recall is computed as the percentage of questions answered correctly within the top three answers over all questions. In case the user question requires more than one entity type[7], we mark an answer correct as long as one of them is attempted and answered correctly. Note that these answers are ranked by Google Places based on relevance to the query. As can be seen, the use of our semantic labels (MSEQ-QA) results in nearly 17 point higher accuracy with a 16 point higher recall compared to WebQA (manual), because of a more directed & effective query to Google Places collection.

Overall, our semantic labels based QA system (MSEQ-QA) answers approximately 48% of the questions with an accuracy of 57% for this challenging task of answering MSEQs.

MSEQ-QA : Qualitative Study and Error Analysis Table 6 presents some examples of questions answered by the MSEQL based QA system. As can be seen our system supports a variety of question intents/entities and due to our choice of an open semantic representation, we are not limited to specific entity types, entity instances, attributes or locations. For example, in $Q1$ the user is looking for "local dinner suggestions" on Christmas eve, and the answer entity returned by our system is to dine at the "St. Peter Stiftskulinarium" in Salzburg, while in $Q2$ the user is looking for recommendations for "SOM tours" (Sound of Music Tours). $Q3$ is incorrect because the entity returned does not fulfill the location constraints of being close to the "bazar" while $Q5$ returns an incorrect entity type. $Q4$ is a complicated question with strict location, budget & attribute constraints.
Error Analysis: We conducted a detailed error study on 105 of the test set questions and we find that approximately 58% of questions were not answered by our system due to limitations of the knowledge source while approximately 42% of the recall loss in the system can be traced to errors in the semantic labels.

Conclusion and Future Work

We have presented the novel task of understanding entity-seeking multi-sentence questions. MSEQs are an important class of questions, as they appear frequently on online forums. They expose novel challenges for semantic parsing as they contain multiple sentences requiring cross-sentence interactions and also need to be built in low data settings due to challenges associated with sourcing training data. We define a set of open semantic labels that we use to formulate a multi-sentence question parsing task.

Our solution consists of sequence labeling based on a BiDiLSTM CRF model. We use hand-engineered features, inter-sentence CCM constraints, and partially-supervised training, enabling the use of crowdsourced incomplete annotation. We find these methods results in a 7pt gain over baseline BiDiLSTM CRFs. We further demonstrate the strength of our work by applying the semantic labels towards a novel end-QA task that returns entity answers for MSEQs from a web API based unstructured knowledge source that outperforms baselines.

An error analysis on our test set suggests the need for a deeper IR system that parses constructs from our semantic representation to execute multiple sub-queries. As another direction of work, we would like to train an end to end neural system to solve our QA task. This would require generating a large dataset of labeled QA pairs which could perhaps be sourced semi-automatically using data available in tourism QA forums.

[7] A question can ask for multiple things, eg., 'museums' as well suggestions for "hotels".

No.	Question	Entity Type	System Answer
1	My family and my brother's family will be in Salzburg over Christmas 2015. We have arranged to do the Sleigh Ride on Christmas day but are keen to do a local style Christmas Day dinner somewhere. Any suggestions?	Special Dinner place	St. Peter Stiftskulinarium, Sankt-Peter-Bezirk 14, 5020 Salzburg
2	Heading to Salzburg by car on Friday September 18th with my wife and her parents (70's) and trying to make the most of the one day. Thinking about a SOM tour, but not sure what the best tour is, not a big fan of huge groups or buses, but would sacrifice for my Mother in Law (LOL). Also thinking about Old Town or the Salzburg Fortress. Any suggestions? Where to park to have easy access as well as a great place for dinner.Thanks so much!	Tour	Bob's Special Tours, Rudolfskai 38, 5020 Salzburg, Austria
3	I am planning to visit Agra for 2 days in mid Dec with my friends.My plan is to try some street food and do some local shopping on day 1 and thus wish to stay in a good budget 3 star hotel (as I wont be spending much time in the hotel) at walking distance from such street foodlocal shopping market.Then on the 2nd day, I want to just relax and enjoy the hotel.(I have booked a premium category hotel, Radisson Blu for this day hoping for a relaxed stay)Please suggest some good hotel or market around which I should book an hotel for my first day.	Hotel with location constraints	Hotel Taj Plaza, Agra, Taj Mahal East Gate, Near Hotel Oberoi Amar Vilas, VIP Road, Shilpgram, Agra, Uttar Pradesh 282001, India
4.	Hi,I am looking for a good hotel in Shillong (preferably near Police bazar) which would offer free wifi, spa and are okay with unmarried couples. My budget is 3k maximum. please suggest the best place to stay.	Hotel with location and budget constraints	Hotel Pegasus Crown, Ward's Lake Road, Police Bazar, Shillong, Meghalaya 793001, India ;

Table 6: Some sample questions from our test set and the answers returned by our system. Answers in green are identified as correct while those in red are incorrect.

References

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley framenet project. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, COLING '98, 86–90. Stroudsburg, PA, USA: Association for Computational Linguistics.

Bapna, A.; Tur, G.; Hakkani-Tur, D.; and Heck, L. 2017. Towards zero shot frame semantic parsing for domain scaling. In *Interspeech 2017*.

Berant, J., and Liang, P. 2014. Semantic parsing via paraphrasing. In *Association for Computational Linguistics (ACL)*.

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1533–1544.

Bogdanova, D., and Foster, J. 2016. This is how we do it: Answer reranking for open-domain how questions with paragraph vectors and minimal feature engineering. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 1290–1295.

Bordes, A.; Chopra, S.; and Weston, J. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 615–620.

Bordes, A.; Weston, J.; and Usunier, N. 2014. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I*, 165–180.

Chang, M.; Ratinov, L.; and Roth, D. 2007. Guiding semi-supervision with constraint-driven learning. In *In Proc. of the Annual Meeting of the ACL*.

Chen, L.; Jose, J. M.; Yu, H.; Yuan, F.; and Zhang, D. 2016. A semantic graph based topic model for question retrieval in community question answering. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, 287–296. New York, NY, USA: ACM.

Cheng, J.; Reddy, S.; Saraswat, V.; and Lapata, M. 2017. Learning structured natural language representations for semantic parsing. *arXiv preprint arXiv:1704.08387*.

Clark, P.; Etzioni, O.; Khot, T.; Sabharwal, A.; Tafjord, O.; Turney, P.; and Khashabi, D. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 2580–2586. AAAI Press.

Contractor, D.; Mausam; and Singla, P. 2016. Entity-balanced gaussian plsa for automated comparison. In *Proceedings of NAACL-HLT*, 69–79.

Fader, A.; Zettlemoyer, L. S.; and Etzioni, O. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, 1608–1618.

Fader, A.; Zettlemoyer, L.; and Etzioni, O. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 1156–1165. New York, NY, USA: ACM.

Guo, S.; Liu, K.; He, S.; Liu, C.; Zhao, J.; and Wei, Z. 2017. Ijcnlp-2017 task 5: Multi-choice question answering in examinations. In *IJCNLP*, 34–40.

He, L.; Lee, K.; Lewis, M.; and Zettlemoyer, L. 2017. Deep

semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 473–483.

He, L.; Lee, K.; Levy, O.; and Zettlemoyer, L. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, 364–369.

He, L.; Lewis, M.; and Zettlemoyer, L. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 643–653.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991.

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

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *CoRR* abs/1705.03551.

Khot, T.; Sabharwal, A.; and Clark, P. 2017. Answering complex questions using open information extraction. *CoRR* abs/1704.05572.

Kwiatkowski, T.; Choi, E.; Artzi, Y.; and Zettlemoyer, L. S. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1545–1556.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, 282–289. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Liang, C.; Hsu, K.; Huang, C.; Li, C.; Miao, S.; and Su, K. 2016. A tag-based statistical english math word problem solver with understanding, reasoning and explanation. In *IJCAI*, 4254–4255. IJCAI/AAAI Press.

Liang, P. S. 2011. *Learning Dependency-Based Compositional Semantics*. Ph.D. Dissertation, University of California, Berkeley.

Lukovnikov, D.; Fischer, A.; Lehmann, J.; and Auer, S. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, 1211–1220. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Mrksic, N.; Séaghdha, D. Ó.; Thomson, B.; Gasic, M.; Rojas-Barahona, L. M.; Su, P.; Vandyke, D.; Wen, T.; and Young, S. J. 2016. Counter-fitting word vectors to linguistic constraints. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 142–148.

Nogueira, R., and Cho, K. 2017. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 574–583.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.* 31(1):71–106.

Qiu, X., and Huang, X. 2015. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, 1305–1311.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Reddy, S.; Täckström, O.; Collins, M.; Kwiatkowski, T.; Das, D.; Steedman, M.; and Lapata, M. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics* 4:127–140.

Sachan, M.; Dubey, K.; and Xing, E. P. 2016. Science question answering using instructional materials. In *ACL (2)*. The Association for Computer Linguistics.

Saha, D.; Floratou, A.; Sankaranarayanan, K.; Minhas, U. F.; Mittal, A. R.; and Özcan, F. 2016. Athena: an ontology-driven system for natural language querying over relational data stores. *Proceedings of the VLDB Endowment* 9(12):1209–1220.

Seo, M. J.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; and Malcolm, C. 2015. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*, 1466–1476. The Association for Computational Linguistics.

Shen, Y.; Rong, W.; Jiang, N.; Peng, B.; Tang, J.; and Xiong, Z. 2015. Word embedding based correlation model for question/answer matching. *arXiv preprint arXiv:1511.04646*.

Sun, H.; Ma, H.; Yih, W.-t.; Tsai, C.-T.; Liu, J.; and Chang, M.-W. 2015. Open domain question answering via semantic enrichment. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, 1045–1055. New York, NY, USA: ACM.

Tan, M.; Xiang, B.; and Zhou, B. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR* abs/1511.04108.

Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; and Suleman, K. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Trivedi, P.; Maheshwari, G.; Dubey, M.; and Lehmann, J. 2017. A corpus for complex question answering over knowledge graphs. In *Proceedings of 16th International Semantic Web Conference - Resources Track (ISWC'2017)*.

Vtyurina, A., and Clarke, C. L. 2016. Complex questions:let me google it for you. In *Proceedings of the second Web QA Workshop WEBQA 2016*.

Wang, D., and Nyberg, E. 2016. Mu oaqa at trec 2016 liveqa: An attentional neural encoder-decoder approach for answer rankin. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016*.

Xu, K.; Reddy, S.; Feng, Y.; Huang, S.; and Zhao, D. 2016. Question Answering on Freebase via Relation Extraction and Textual Evidence. In *Proceedings of the Association for Computational Linguistics (ACL 2016)*. Berlin, Germany: Association for Computational Linguistics.

Yang, B., and Mitchell, T. M. 2017. A joint sequential and relational model for frame-semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 1247–1256.

Yih, W.; Chang, M.; He, X.; and Gao, J. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *ACL (1)*, 1321–1331. The Association for Computer Linguistics.

Yin, P.; Duan, N.; Kao, B.; Bao, J.; and Zhou, M. 2015. Answering questions with complex semantic constraints on open knowledge bases. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, 1301–1310. New York, NY, USA: ACM.

Zettlemoyer, L. S. 2009. *Learning to map sentences to logical form*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Zhang, K.; Wu, W.; Wang, F.; Zhou, M.; and Li, Z. 2016. Learning distributed representations of data in community question answering for question retrieval. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, 533–542. New York, NY, USA: ACM.

Zhou, J., and Xu, W. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, 1127–1137.