Natural Language Engineering (2020), 1–23
doi:10.1017/S1351324920000017

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Constrained BERT BiLSTM CRF for understanding multi-sentence entity-seeking questions

Danish Contractor[1,2,*,†], Barun Patra[3,‡], Mausam[2], and Parag Singla[2]

[1]IBM Research AI, [2]Indian Institute of Technology Delhi, New Delhi, India and [3]Microsoft Corporation, Redmond, WA, USA
*Corresponding author. E-mail: dcontrac@in.ibm.com

(Received 27 September 2018; revised 6 January 2020; accepted 15 January 2020)

Abstract

We present the novel task of understanding multi-sentence *entity-seeking* questions (MSEQs), that is, the 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 BiLSTM (bidirectional LSTM) conditional random field (CRF), and to overcome the challenges of operating with low training data, we supplement it by using BERT embeddings, hand-designed features, as well as hard and soft constraints spanning multiple sentences. We find that this results in a 12–15 points gain over a vanilla BiLSTM 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 36% more questions with 35% more (relative) accuracy as compared to baselines. We also demonstrate how our framework can rapidly enable the parsing of MSEQs in an entirely new domain with small amounts of training data and little change in the semantic representation.

Keywords: Question answering; Question understanding; Multi-sentence question parsing; Sequence tagging with partially labeled data; BERT BiLSTM CRF

1. Introduction

We introduce the novel task of understanding multi-sentence questions. Specifically, we focus our attention on multi-sentence *entity-seeking* questions (MSEQs), that is, the 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 of MSEQ from a tourism forum[a], 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.

[†] Work carried as part of the author's PhD research at IIT Delhi.
[‡] Majority of the work carried out when the author was a student at IIT Delhi.
[a] http://tripadvisor.com

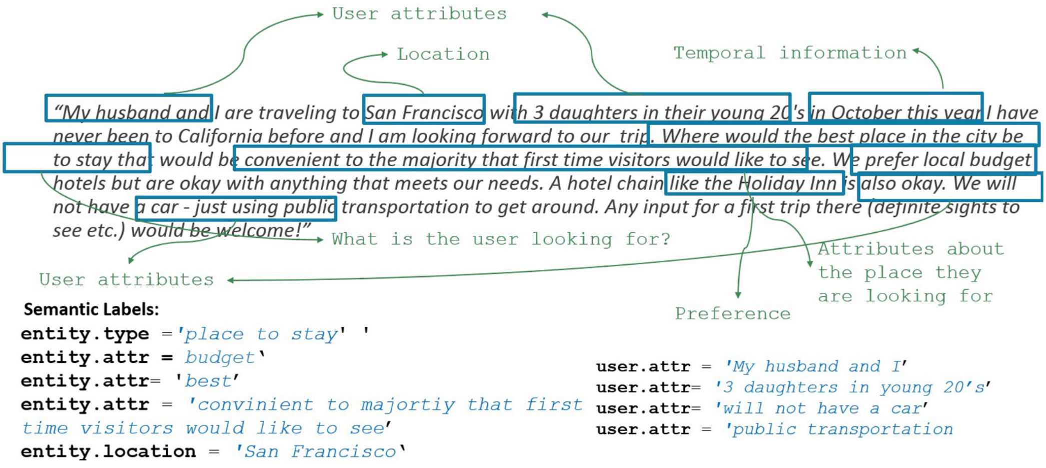

User attributes
Location
Temporal information
"My husband and I are traveling to San Francisco with 3 daughters in their young 20's in October this year I have never been to California before and I am looking forward to our trip. Where would the best place in the city be to stay that would be convenient to the majority that first time visitors would like to see. We prefer local budget hotels but are okay with anything that meets our needs. A hotel chain like the Holiday Inn is also okay. We will not have a car - just using public transportation to get around. Any input for a first trip there (definite sights to see etc.) would be welcome!"
What is the user looking for?
Attributes about the place they are looking for
User attributes
Preference
Semantic Labels:
entity.type ='place to stay' '
entity.attr = budget'
entity.attr= 'best'
entity.attr = 'convinient to majortiy that first time visitors would like to see'
entity.location = 'San Francisco'
user.attr = 'My husband and I'
user.attr= '3 daughters in young 20's'
user.attr= 'will not have a car'
user.attr = 'public transportation

Fig 1. An MSEQ annotated with our semantic labels.

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. For example, 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 can 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[b]) 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 (SRL) (Yang and Mitchell 2017), slot-filling tasks (Bapna *et al.* 2017), and query formulation (Vtyurina and Clarke 2016; Wang and Nyberg 2016; Nogueira and Cho 2017), semantic parsing of MSEQs raises 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. Thus, this requires the models to operate in low training data settings.

In order to address these challenges and label MSEQs, we use a bidirectional LSTM (conditional random field) CRF (BiLSTM CRF) (Huang, Xu and Yu 2015) as our base model and extend it in three ways. First, we improve performance by inputting contextual embeddings from BERT (Devlin *et al.* 2019) into the model. We refer to this configuration as BERT BiLSTM CRF. Second, we encode knowledge by incorporating hand-designed features as well as semantic constraints over the entire multi-sentence question during end-to-end training. This can be thought of as incorporating constrained conditional model (CCM)-style constraints and inference (Chang,

[b] We use the phrases "semantic labeling" and "semantic parsing" interchangeably in this paper.

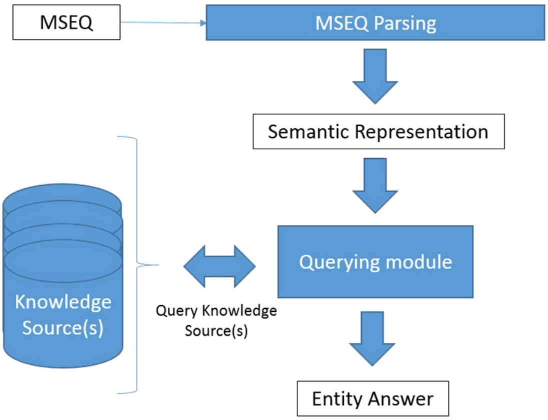


Fig. 2. Schematic representation of the QA system.

Ratinov and Roth 2007) in a neural model. Finally, we find that crowdsourcing complete annotations is hard, since the task is complex. In this work, we are able to improve training by partially labeled questions which are easier to source.

1.1 Contributions

In summary, our paper makes the following contributions:

1. We present the novel task of understanding MSEQs. We define *open* semantic 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 BERT BiLSTM CRF model. We extend this by providing hand-designed features and using CCM inference, which allows us to specify within-sentence as well as inter-sentence (hard and soft) constraints. This helps encode 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 crowdsourced 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 15 points (pt) improvement in F1 scores over a baseline BiLSTM CRF.
4. We demonstrate the applicability of our semantic labels in two different end tasks. (i) The first is a 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 36% more questions with 35% more (relative) accuracy as compared to baselines. (ii) The second task is semantic labeling of MSEQs in a new domain about book recommendations with minimal training data.

2. Problem statement

Given an MSEQ, our goal is to first parse and generate a semantic representation of the question using labels that identify *informative* portions of a question. The semantic representation of the question can then be used to return an entity answer for the question, using a knowledge source. Thus, our QA system consists of two modules (see Figure 2): question understanding (MSEQ parsing) and a querying module to return entity answers. The modularized two-step architecture allows us to tackle different aspects of the problem independently. The semantic representation generated by the question understanding module is generic and not tied to a specific corpora or ontology. This allows the answering module to be optimized efficiently for any knowledge source, and supports the integration of multiple data sources, each with their own schema and strengths for answering. In this paper, we experiment with the Google Places Web collection[c] as our knowledge source. It consists of semi-structured data including geographic information, entity categories, entity reviews, etc. The collection is queried using a web API that accepts an unstructured text string as query.

3. Related work

To the best of our knowledge, we are the first to explicitly address the task of *understanding* MSEQs and demonstrate its use in an answering task. There are different aspects of our work that relate to existing literature and we discuss them in this section. We begin by contrasting our work on multi-sentence question understanding and answering with recent work on question-answering (Section 3.1). We then include a review of related work on semantic representations of questions (Section 3.2) followed by a brief survey of recent literature on semantic labeling (Section 3.3). We conclude with a summary in Section 3.4.

3.1 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.* 2018). Such systems can be rapidly retrained for different domains, as they use minimal hand-constructed or domain-specific features. But, they require huge amounts of labeled QA pairs for training.

In contrast, a pipelined approach (Kwiatkowski *et al.* 2013; Berant and Liang 2014; Fader *et al.* 2014; Fader, Zettlemoyer and Etzioni 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.

We choose to summarize popular approaches in QA systems on the basis of (a) the type of questions they answer, (b) the nature of knowledge base /Corpus used for answering, and (c) the nature of answers returned by the answering system (See Table 1).

In this paper, we return entity answers to MSEQs. 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 (Berant and Liang 2014; Fader *et al.* 2014; Sun *et al.* 2015; Yin *et al.* 2015; Saha *et al.* 2016; Khot, Sabharwal and Clark 2017; Lukovnikov *et al.* 2017; Zheng *et al.* 2018; Zhao *et al.* 2019). Reading comprehension tasks (Trischler *et al.* 2016; Joshi *et al.* 2017; Trivedi *et al.* 2017; Rajpurkar *et al.* 2018; Yang *et al.* 2018; Dua *et al.* 2019) require answers to be generated from unstructured text also only return answers for relatively 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 exam questions and science questions (Seo *et al.* 2015; Clark *et al.* 2016; Guo *et al.* 2017; Khot *et al.* 2017; Palmer, Hwa and Riedel 2017; Zhang *et al.* 2018), 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 (Pithyaachariyakul and Kulkarni 2018; Qiu and Huang 2015; Shen *et al.* 2015; Tan *et al.* 2015; Bogdanova and Foster 2016) match questions with *user*-provided answers, instead of entities from background knowledge source. IR-based systems (Vtyurina and Clarke 2016; Wang and

[c] https://developers.google.com/places/web-service/intro

Table 1. Related work: QA

Question type	Knowledge type	Answer type	Related work
Single sentence	Structured (e.g., DBPedia, Freebase)	Entity	Bordes *et al.* (2014b); Bordes *et al.* (2015); Lukovnikov *et al.* (2017)
	Structured (Open IE style KBs)	Entity	Berant and Liang (2014); Fader *et al.* (2014)
	Structured + Unstructured (Open IE style KBs with supporting text passages on entities)	Entity	Das *et al.* (2017)
	Structured (databases)	Tables/table rows	Pazos R. *et al.* (2013); Saha *et al.* (2016)
	Unstructured	Text spans	Trischler *et al.* (2016); Chen *et al.* (2017); Joshi *et al.* (2017); Trivedi *et al.* (2017); Rajpurkar *et al.* (2018); Yang *et al.* (2018); Dua *et al.* (2019)
Multi-sentence	Unstructured	Text passages	Wang and Nyberg (2015); Wang and Nyberg (2016); Vtyurina and Clarke (2016)
	Multiple choice answers	Answers from specified choices	Guo *et al.* (2017); Khot *et al.* (2017); Palmer *et al.* (2017); Welbl *et al.* (2018); Zhang *et al.* (2018)
	Unstructured	Text (Answer) passages	Bogdanova and Foster (2016); Romeo *et al.* (2016); Singh and Simperl (2016); Srba and Bielikova (2016)
	Unstructured (QA pairs + Wikipedia)	Entity	Iyyer *et al.* (2014)
	Semi-structured meta-data + Unstructured (Entity Reviews)	**Entity**	**Our work**

Nyberg 2016; Pithyaachariyakul and Kulkarni 2018) query the web for open-domain questions, but return long (1000-character) passages as answers; they have not been developed for or tested on entity-seeking questions. These techniques that can handle MSEQs (Vtyurina and Clarke 2016; Wang and Nyberg 2016; Pithyaachariyakul and Kulkarni 2018) typically perform retrieval using keywords extracted from questions; these do not "understand" the questions and cannot answer many tourism questions, as our experiments show (Section 7). The more traditional solutions (e.g., semantic parsing) that parse the questions deeply can process only *single*-sentence questions (Fader *et al.* 2013, 2014; Kwiatkowski *et al.* 2013; Berant and Liang 2014; Zheng *et al.* 2018).

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 Google Places API we experiment with (as our knowledge source) has millions of entities. It is important to note that for answering an MSEQ, the answer space can include thousands of candidate entities per question, with large unstructured review documents about each entity that help determine the best answer entity. Thus, these documents are significantly longer than passages (or similar length articles) that have traditionally been used in neural QA tasks. Recently, tasks that require multi-hop reasoning have also been proposed. This involves simple QA via neural machine comprehension of longer/multi-passage documents (Trivedi *et al.* 2017; Welbl *et al.* 2018; Yang *et al.* 2018). Extending such a task for MSEQs could be an interesting extension for future work.

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

3.2 Question parsing

QA systems use a variety of different intermediate semantic representations. Most of them, including the rich body of work in NLIDB (Natural Language Interfaces for Databases) and semantic parsing, parse *single* sentence questions into a query based on the underlying ontology or database schema and are often learned directly by defining grammars, rules, and templates (Zettlemoyer 2009; Liang 2011; Berant *et al.* 2013; Kwiatkowski *et al.* 2013; Sun *et al.* 2015; Yih *et al.* 2015; Reddy *et al.* 2016; Saha *et al.* 2016; Abujabal *et al.* 2017; Cheng *et al.* 2017; Khot *et al.* 2017; Lukovnikov *et al.* 2017; Zheng *et al.* 2018). Works such as Fader *et al.* (2014) and 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 MSEQs. 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 2014a; Bordes, Weston, and Usunier 2014b; Chen *et al.* 2016; Xu *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 MSEQs. 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-relevant 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.

3.3 Neural semantic parsing

There is a large body of literature dealing with semantic parsing of single sentences, especially for frames in PropBank and FrameNet (Baker *et al.* 1998; Palmer, Gildea and Kingsbury 2005). Most recently, methods that use neural architectures for SRL have been developed. For instance, work by Zhou and Xu (2015) uses a BiLSTM CRF for labeling sentences with PropBank predicate argument structures, while work by (He *et al.* 2017, 2018) relies on a BiLSTM with BIO-encoding constraints during LSTM decoding. Other recent work by Yang and Mitchell (2017) proposes a BiLSTM CRF model that is further used in a graphical model that encodes SRL structural constraints as factors. Work such as Bapna *et al.* (2017) uses a BiLSTM tagger for predicting task-oriented information slots from sentences. Our work uses similar approaches for labeling (parsing) MSEQs, but we note that such systems cannot be directly used in our task due to their model-specific optimization 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 (Section 6).

3.4 Summary

In summary, while related work shares aspects with our task there are three main distinguishing features that are not jointly addressed in existing work: (i) Question type: A major focus of existing work has been on single sentence questions, sometimes with the added complexity arising out of entity relations and co-reference. Such questions are often posed as "which/where/when/who/what" questions. However, our work uses multi-sentence questions which can additionally contain vague expression of intents as well as information that is irrelevant for the answering task. (ii) Knowledge: Most information-seeking questions either answer factoid-style questions from knowledge graphs and structured knowledge bases or answer them from paragraphs of text which contain explicit answers. In contrast, our work uses unstructured or semi-structured knowledge sources and our querying representation makes no assumptions of the underlying knowledge store. (iii) Answer-type: Existing QA systems either return answer spans (reading comprehension tasks), or documents (from the web or large text collections) to fulfill a knowledge-grounded information query that relies on explicit mention (or with some degree of semantic gap) of the answer. In contrast, our QA pipeline returns entity answers from a (black box) web API that accepts a text string as query and internally uses structured and unstructured data including entity reviews containing subjective opinions to return an answer.

In the next section, we describe our question representation (Section 4) followed by details about our labeling system (Section 5). We present experiments in Section 6 and details of our answering component in Section 7. We finally conclude the paper in Section 8 along with suggestions for future work.

4. 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[d]. 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 toward 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 (e.g., 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, for example, "will not have a car" which may become relevant for answering the question. The phrase "San Francisco" describes the location of the entity and is labeled with a domain-specific label (*entity.location*).

5. MSEQ semantic labeling

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 Section 4. We utilize a BERT BiLSTM CRF 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 CCM (Chang *et al.* 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.

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

5.1 Features

We incorporate a number of (domain-independent) features into our BERT BiLSTM CRF model where each unique feature is represented as a one-hot vector and concatenated with the BERT embedding representation of each token. In experiments with BiLSTM CRF models without BERT, we replace the BERT embeddings with pre-trained word2vec (Mikolov *et al.* 2013) embeddings that are concatenated with the one-hot feature embeddings.

Our features are described as follows: (a) Lexical features for capitalization, indicating numerals, etc., token-level features based on part-of-speech tags and named-entity recognition labels. (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—for example, for every noun and adjective, an attribute indicator feature is on if any of its ancestors is a potential *type* as indicated by the type feature; indicator features for descriptive phrases (Contractor *et al.* 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).

5.2 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 nonadjacent 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; (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 toward improving recall, the last constraint helps improve precision of *entity.type* labels.

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

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

Here, i indexes over all examples and k over all constraints. $\mathbf{x^{(i)}}$ is the ith sequence and $\mathbf{y^{(i)}}$ is its labeling. ϕ and w are feature and weight vectors, respectively. d_{C_k} and ρ_k, respectively, denote the violation score and weight associated with kth constraint. The w parameters are learned analogous to a vanilla CRF and computing ρ parameters resorts to counting. 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 we extend its use in a combined model of BERT BiLSTM CRF with CCM constraints (referred to as BERT BiLSTM CCM) that is trained end to end (Figure 3).

Specifically, let $\mathcal{Y}$ be the set of label indices[e]. Let T be the sequence length and $x_0 \cdots x_{T-1}$ be the tokens, $\phi(x^i) \in \mathbb{R}^{|\mathcal{Y}|}$ be the feature vector for the ith token (the output of the feed-forward layer in the BiLSTM-CRF), and $\phi(x^i)[j]$ denoting the feature associated with the ith token and jth

[e] We overload the notation for labels and their associated indices. So $l_t \in \mathcal{Y}$ denotes an index of a label, while *entity.type* $\in \mathcal{Y}$ denotes the index associated with entity.type.

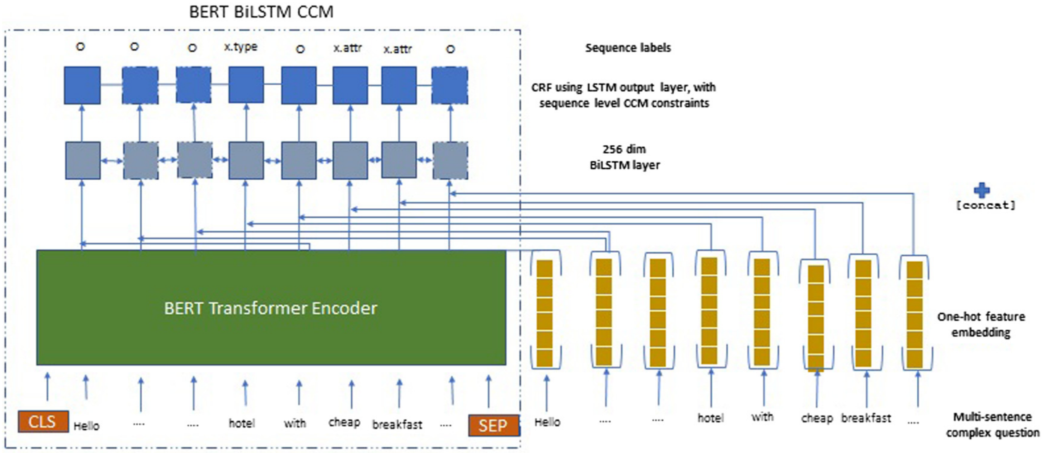


Fig. 3. BERT BiLSTM CCM with features for sequence labeling.

label. Let $w \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ be the transition matrix, with $w[i, j]$ denoting the weights associated with a transition from $i \rightarrow j$. Then

$$\begin{aligned}
\max_{\mathbb{1}} \quad & \mathcal{C}_1(\mathbb{1}) = \sum_{l\in\mathcal{Y}} \phi(x^{(i)})[l]\mathbb{1}_{0,l} + \sum_{i=1}^{T-1}\sum_{l_s\in\mathcal{Y}}\sum_{l_t\in\mathcal{Y}} (\phi(x^{(i)})[l_t] + w[l_s, l_t])\mathbb{1}_{i,l_s,l_t} \\
\text{s.t} \quad & \forall_{l\in\mathcal{Y}}\mathbb{1}_{0,l} \in \{0, 1\} \\
& \forall_{i\in 1\cdots T-1}\forall_{l_s\in\mathcal{Y}}\forall_{l_t\in\mathcal{Y}}\mathbb{1}_{i,l_s,l_t} \in \{0, 1\} \\
& \sum_{l\in\mathcal{Y}} \mathbb{1}_{0,l} = 1 \\
& \forall_{i\in 1\cdots T-2}\forall_{l\in\mathcal{Y}} \sum_{1_s\in\mathcal{Y}} \mathbb{1}_{i,l_s,l} = \sum_{1_t\in\mathcal{Y}} \mathbb{1}_{i,l,l_t}
\end{aligned} \tag{2}$$

defines the Viterbi decoding for a linear chain CRF. The variable $\mathbb{1}_{0,l} = 1$ if the first token of the sequence is tagged l in the optimal Viterbi sequence, and zero otherwise. Furthermore $\mathbb{1}_{i,l_s,l_t} = 1$ if the ith token is tagged with label l_t and the $(i-1)$th token is tagged l_s in the optimal Viterbi sequence, and is marked zero otherwise.

Type label constraints (hard): In order to model the type-based hard constraint (there has to be at least one *entity.type* label in the sequence), we add the following constraint to the optimization problem:

$$\mathbb{1}_{0,\text{entity.type}} + \sum_{i=1}^{T-1}\sum_{l_s\in\mathcal{Y}} \mathbb{1}_{i,l_s,\text{entity.type}} \geq 1 \tag{3}$$

Here, $\mathbb{1}_{0,entity.type} = 1$ if the first token is tagged as a type, while $\sum_{l_s\in\mathcal{Y}} \mathbb{1}_{i,l_s,entity.type} = 1$ if the ith token is tagged as an entity.

Attribute label constraints (soft): In order to model the attribute-based constraint (the nonexistence of an *entity.attr* label in the sequence is penalized), we introduce a dummy variable d for our ILP formulation. Then, given the constraint violation penalty η, we change the model optimization problem as

$$\begin{aligned} \max_{\mathbb{1},d} \quad & \mathcal{C}_2(\mathbb{1}, d) = \mathcal{C}_1(\mathbb{1}) - \eta \cdot d \\ \text{s.t} \quad & d \in \{0, 1\} \\ & \mathbb{1}_{0,\text{entity.attr}} + \sum_{i=1}^{T-1} \sum_{l_s \in \mathcal{Y}} \mathbb{1}_{i,l_s,\text{entity.attr}} + d \geq 1 \end{aligned} \tag{4}$$

Here, if the constraint is violated, then $d = 1$ and the objective suffers a penalty of η. Conversely, since it is a minimization over d as well, if the constraint is satisfied, then $d = 0$ and the objective is not penalized.

Inter-sentence-type constraint: We model the constraint that all *entity.type* labels should appear in a single sentence. We implement this as a soft constraint by imposing an *L1* penalty on the number of sentences containing an *entity.type* (thereby insuring that fewer sentences contain type labels). Let the number of sentences be k. Let e_i denote the index of the start of the ith sentence, such that $\{x_j, e_i \leq j < e_{i+1}\}$ are the tokens in the ith sentence (note that $e_0 = 0$). Define $z_0, ..., z_{k-1}$ to model sentence indicators, with $z_i = 1$ if the ith sentence contains a type. Let η_2 be the associated penalty. We modify the optimization problem then as follows:

$$\begin{aligned} \max_{\mathbb{1},d,\mathbb{Z}} \quad & \mathcal{C}_2(\mathbb{1}, d) - \eta_2 \cdot \left(\sum_i z_i \right) \\ \text{s.t} \quad & \forall_i z_i \in \{0, 1\} \\ & \forall_i \forall_{j, e_i \leq j < e_{i+1}} z_i - \sum_{l_s} \mathbb{1}_{j,l_s,entity.type} \geq 0 \end{aligned} \tag{5}$$

Here, the variable j indexes over the tokens for the ith sentence. $\sum_{ls} \mathbb{1}_{j,l_s,entity.type} = 1$ if the jth token is a type, and is 0 otherwise. Hence if any of the tokens in the ith sentence is labeled a type, $z_i = 1$. Note that combined with Equation (3), we also have $\sum_i z_i \geq 1$.

5.3 Partially labeled data

Data collection: In order to obtain a larger amount of labeled data for our task, we make use of crowdsourcing (AMT). Since our labeling task can be complex, we divide our crowd task into multiple steps. We first ask the 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.

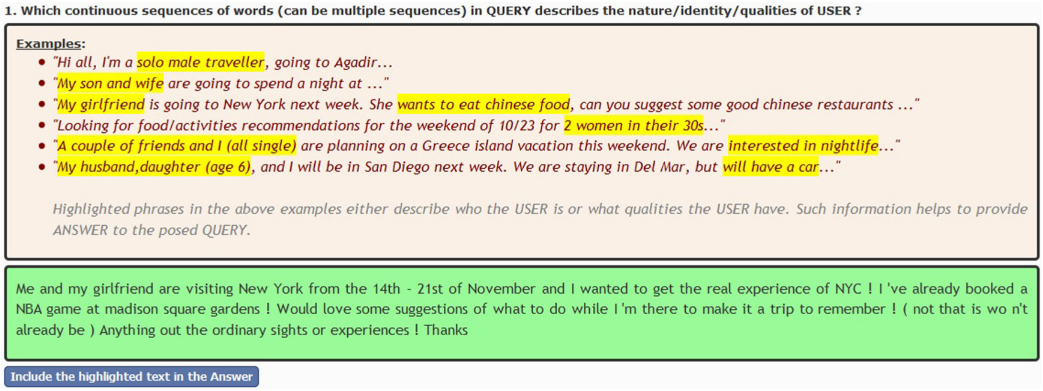
5.3.1 Crowdsourcing task

We defined three AMT tasks in the form of questionnaires:

- Questionnaire 1 : To identify posts of relevance for our task. This is to filter posts that may be unrelated to our task[f].
- Questionnaire 2 : To identify the *user* entities and its labels.
- Questionnaire 3 : To identify the answer entities and its labels.

In the first questionnaire (AMT Task 1) we ask the users to identify any non-entity-seeking questions as well the number of entity types requested in a given query. We remove any posts that

[f] Forum posts can often contain reviews, advertisements, etc., apart from types of questions that we exclude in this paper.



1. Which continuous sequences of words (can be multiple sequences) in QUERY describes the nature/identity/qualities of USER ?

Examples:

- "Hi all, I'm a solo male traveller, going to Agadir...
- "My son and wife are going to spend a night at ..."
- "My girlfriend is going to New York next week. She wants to eat chinese food, can you suggest some good chinese restaurants ..."
- "Looking for food/activities recommendations for the weekend of 10/23 for 2 women in their 30s..."
- "A couple of friends and I (all single) are planning on a Greece island vacation this weekend. We are interested in nightlife..."
- "My husband,daughter (age 6), and I will be in San Diego next week. We are staying in Del Mar, but will have a car..."

Highlighted phrases in the above examples either describe who the USER is or what qualities the USER have. Such information helps to provide ANSWER to the posed QUERY.

Me and my girlfriend are visiting New York from the 14th - 21st of November and I wanted to get the real experience of NYC ! I 've already booked a NBA game at madison square gardens ! Would love some suggestions of what to do while I 'm there to make it a trip to remember ! (not that is wo n't already be) Anything out the ordinary sights or experiences ! Thanks

Fig. 4. Snippet of the second questionnaire given to AMT workers.

ask for multiple entity types[g]. The second questionnaire (AMT Task 2) asks the following question to the AMT workers. We paid $0.20 to each worker for this task.

- *"Which continuous sequences of words (can be multiple sequences) in the QUESTION describes the nature/identity/qualities of USER?"*

The QUESTION refers to the actual question posed by a user on a forum page and the answer to these questions gives us the *user.attr* labels. Figure 4 shows a sample snippet of the questionnaire.

The last questionnaire asks the following questions to the AMT workers.

- *"Given that the USER is asking only a single type of recommendation/suggestion, which sequence of words (only one sequence from a single sentence, prefer a continuous sequence) in QUESTION tells you what the USER is asking for?"*
- *"What is the shortest sequence of words in 'A1 (Answer to Question 1)' describes a category? For example, place to stay, restaurant, show, place to eat, place to have dinner, spot, hotel, etc."*
- *"What words/phrases (need not be continuous, can be multiple) in the QUESTION give a sense of location about the ANSWER or 'A2' (Answer to Question 2)?"*
- *"What words/phrases (need not be continuous, can be multiple) in the QUESTION give more description about the ANSWER or the 'A2' (Answer to Question 2)?"*

These questions give us the *entity.type*, *entity.location*, and *entity.attribute* labels. We paid $0.30 to each worker for this task.

We obtain two sets of labels (different workers) on each question. However, due to the complex nature of the task we find that workers are not complete in their labeling and we therefore only use token labels where both set of workers agreed on labels. Thus, we are able to source annotations with high precision, while recall can be low. Table 2 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. For example, 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*.

[g] This is only so that additional work on resolving attributes and entities is not required. Resolving entities and their corresponding attributes is a useful direction for future work.

Table 2. Agreement for *entity* labels on AMT

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

5.3.2 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 *et al.* 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 *et al.* 2007).

Given a data set 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 (Section 5.2). 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

$$(w^{(t+1)}, \rho^{(t+1)}) = \gamma(w^{(0)}, \rho^{(0)}) + (1-\gamma)\text{Learn}(U^{(t)}) \tag{6}$$

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 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, with an added constraint that the predicted labels for the partially annotated sequences have to be consistent with the human labels. γ controls the relative importance of the labeled and partial examples. To the best of our knowledge, we are the first to exploit partial supervision from a crowdsourcing platform in this manner.

6. 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 data set, evaluation methodology, and results in detail.

6.1 Data set

For our current evaluation, we used the following three semantic labels: *entity.type*, *entity.attr*, and *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 data set 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 crowdsourced workers to our training set. As described in Section 5.3.1, 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).

6.2 Methodology

Sequence-tagged tokens identify *phrases* for each semantic label; therefore, 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* (with gold *entity.attr* being "budget", "best," and "convenient to the majority that first time visitors would like to see"), then our matching-metric will compute precision as 0.75 (1.0 for "convenient to the majority"(covered completely by "convenient to the majority that first time visitors would like to see") and 0.5 for "local budget"(partially covered by "budget")) and recall as 0.45 (1.0 for "budget" (completely covered by predicted entity "local budget"), 0.0 for "best" (not covered by any predicted entities), and 0.333 for "convenient to the majority . . . like to see"(covered by predicted "convenient to the majority")).

We use the Mallet toolkit[h] for our baseline CRF implementation and the GLPK ILP-based solver[i] for CCM inference. In the case of BiLSTM-based CRF, we use the implementation provided by Gardner *et al.* (2017). The BiLSTM 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. In case of BERT BiLSTM CRF, we use the contextualized BERT embeddings from the BERT-small pretrained model as an input to the LSTM layer and BERT implementation from HuggingFace Transformers (Wolf *et al.* 2019). For CoDL learning, we set γ to 0.9 as per original authors' recommendations.

6.3 Results

Table 3 reports the performance of our semantic labeler under different incremental configurations. We find that the models based on BiLSTM CRF and the BERT BiLSTM CRF (middle and lower halves of the table) outperform 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 BiLSTM CRF and BERT BiLSTM CRF using features are 56.2 and 64.4, respectively. As a baseline, we use the neural 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. The use of hand-designed features, CCM constraints in the BERT BiLSTM CRF (referred to as BERT BiLSTM CCM), along with learning from partially annotated crowd data has over a 15 pt gain over the baseline BiLSTM CRF model. Further, we note that the usage of hand-curated features, within-sentence and cross-sentence constraints as well as partial supervision, each help successively improve the results in all configurations. Next, we study the effect of each of these enhancements in detail.

6.3.1 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 pt. 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 is an indicator of the word's relative importance, and the feature also helps improves overall performance.

6.3.2 Effect of constraints

A closer inspection of Table 3 reveals that the vanilla CRF configuration sees more benefit in using our CCM constraints as compared to the BiLSTM CRF-based model (4 vs. 1 pt). To understand why, we study the detailed precision-recall characteristics of individual labels; the results for

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

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

Table 3. Sequence tagger $F1$ scores using CRF with all features (feat), CCM with all features and constraints, and partially supervised CCM over partially labeled crowd data. The second set of results mirrors these settings using a bidirectional LSTM CRF. Results are statistically significant (paired t test, p value < 0.02 for aggregate $F1$ for each CRF and corresponding CCM model pair). Models with "PS" as a prefix use partial supervision

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
BiLSTM CRF	53.3	47.6	52.1	51.0
BiLSTM CRF + Feat	58.4	48.1	62.0	56.2
BiLSTM CCM + Feat	59.4	49.8	62.3	57.2
PS BiLSTM CCM + Feat	62.9	50.4	61.5	58.3
BERT Labeling	59.6	50.6	59.5	56.6
BiLSTM BERT CRF	63.4	56.5	**73.4**	64.4
BiLSTM BERT CRF + Feat	63.9	**57.9**	69.2	63.7
BiLSTM BERT CCM + Feat	66.5	56.7	72.9	65.3
PS BERT BiLSTM CCM + Feat	**70.8**	56.0	72.4	**66.4**

Table 4. (i) Precision and recall of *entity.type* with and without CCM inference

Algorithm	Prec	Recall	F1
CRF (all features)	66.9	41.7	51.4
CCM (all features)	62.1	57.2	59.6
BiLSTM CRF with features	54.1	63.6	58.4
BiLSTM CCM with features	55.1	64.5	59.4
BiLSTM BERT CRF with features	66.4	61.5	63.9
BiLSTM BERT CCM with features	65.0	68.0	66.5

entity.type are reported in Table 4. We find that the BiLSTM CRF-based model has significantly higher recall than their equivalent vanilla CRF counterpart while the opposite trend is observed for precision. As a result, since two of the three constraints we used in CCM are oriented toward improving recall[j], we find that they improve overall F1 more by finding tags that were otherwise of lower probability (i.e., improving recall). Interestingly, in case of the BERT BiLSTM CRF-based model, we find that precision-recall characteristics are similar (higher precision than recall) to those seen in the vanilla CRF-based setup, and thus again, the benefit of using constraints is larger.

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

6.3.3 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 crowdsourced 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 3) hurts the performance significantly resulting in an aggregate $F1$ of just 48.0 while using partially labeled data with CCM results in an F1 of 56.5. The corresponding $F1$ scores of partially supervised BiLSTM CCM and BERT BiLSTM CCM systems (trained using partially labeled data) are 58.3 and 66.4, respectively.

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.

7. MSEQ semantic labels: Application

We now demonstrate the usefulness of our MSEQ semantic labels and tagging framework (i) 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—and (ii) by demonstrating the creation of an MSEQ labeler for a different domain (books recommendation).

7.1 MSEQ Labeler based QA system

Our novel QA task evaluation attempts to return entity answers for multi-sentence tourism forum questions. 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[k]. 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 flags 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 occurs within a specified window of the trigger word. Table 5 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.

7.1.1 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 shortlists 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 three to four words, to form the final query. In our setting, we lack the data to train a supervised method for selecting these words from the tf-idf-ranked list. Therefore, for best performance, instead of using supervised learning for further shortlisting

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

Table 5. Performance of negation detection using gold sequence labels and system generated labels

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

Table 6. QA task results using the Google places web API as knowledge source

System	Acc@3 (%)	MRR	Recall (%)
WebQA	31.6	0.28	19.5
WebQA (manual)	41.8	0.35	39.4
MSEQ-QA	**56.7**	**0.46**	**53.6**

keywords (as in the original paper), in our implementation an expert chooses 3–4 best words manually from the top 10 words. This query executed against the Google Places collection API 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 110 of those that were not entity-seeking. The remaining 190 questions form 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.

7.1.2 MSEQ-QA: Results

Results: Table 6 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[l], 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[m] (MSEQ-QA) results in nearly 15 point higher accuracy with a 14 point higher recall compared to WebQA (manual), because of a more directed and effective query to Google Places collection.

Overall, our semantic labels-based QA system (MSEQ-QA) answers approximately 54% of the questions with an accuracy of 57% for this challenging task of answering MSEQs.

7.1.3 MSEQ-QA: Qualitative study and error analysis

Table 7 presents some examples of questions[n] answered by the MSEQ Labeler-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

[l] A question can ask for multiple things, for example, "museums" as well suggestions for "hotels".

[m] We use the best performing parsing system PS BERT BiLSTM CCM with features.

[n] Actual questions posted on forums at TripAdvisor.com

Table 7. 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

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 (70s) 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	What can you do in Helsinki on a Sunday morning? What would you recommend a tourist to do or see on a Sunday morning? I'll be arriving at 7 in the morning, and it seems like everything closed on a Sunday morning—either it's not open on Sundays or else it'll open but later on in the day.	Things to do/see	Senate Square, 00170 Helsinki, Finland Ateneum, Kaivokatu 2, 00100 Helsinki, Finland
4	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 won't be spending much time in the hotel) at walking distance from such street food local shopping market. Then on the second 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
5	Hi there. I am going to Tallinn in a month from just one night on a Saturday. I am 28 and am going with five of my friends. Where should we stay so we are near the best clubs in the city? Any recommendations are appreciated!!! Thanks.	Place to stay close to clubs	Club Prive, Tallinn, Estonia
6	A few friends and I are coming up to Newport for a couple of nights and are looking for restaurant suggestions. We are thinking something casual for the first night. Is Flo's any good? And then something nicer on Saturday night. . ..preferably a restaurant with good seafood. Also, any suggestions for good breakfast?	Restaurant based on cuisine	The Red Parrot Restaurant, 48 Thames St, Newport, RI 02840, United States
7	Dear All forum members, I am Yash Khatri from Delhi. I am traveling to Srinagar from July 13th, 2016, to July 17th, 2016.I am going there for a show, and I'll be free on July 15th and 16th, 2016. I was thinking to hire a bike at Srinagar and travel to Gulmarg or Pahalgam.Queries: (1) Where can I rent a bike at Srinagar and how much will it cost me? (2) What is better for a quick visit; Gulmarg or Pahalgam? Please help! Thanks	Motorcycle rental	Kashmir Bikers – Bike Rentals, Sheikh complex, Shiraz chowk, Khanyar, Near J&K Bank, Khanyar, Srinagar, Jammu and Kashmir, 190003
8	In a couple of weeks, we will have an almost 2-hour layover in Zagreb before flying on to Dubrovnik. Any recommendations for lunch ?	A location for lunch that can be visited in a 2 hour layover	Hotel Dubrovnik, Gajeva ul. 1, 10000, Zagreb, Croatia
9	Hi, I am looking for a good hotel in Shillong (preferably near Police bazar) which would offer free Wi-Fi and 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 8. Classification of errors made by our MSEQ-labels-based answering system (using Google Places web API as knowledge source)

Error type	Error (%)	Examples
Incorrect answer returned due to incorrect *entity.type*	23	Bad *entity.type* extractions results in incorrect answers
Incorrect answer returned by knowledge source	23	*entity.attribute* criteria was not fulfilled—for example, shop allows renting bicycles but not for tours
Incorrect answer returned due to incomplete labeling	17	*entity.attribute* not getting extracted
Incorrect answer/answer not returned due to knowledge source limitations	37	Query requesting places "around" a city, or between two cities, *entity.type* extracted as "day trips", "cruises", etc. Requests for *entity.type* where queries were about bus services, activities, and train stations

instances, attributes, or locations. For example, in *Q*1 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 *Q*2 the user is looking for recommendations for "SOM tours" (Sound of Music Tours). A quick internet search shows that our system's answer, "Bob's Special Tours," is famous for their SOM tours in that area. This question also requests for restaurant suggestions in the old town, but since we focus on returning answers for just one *entity.type*, this part of the question is not attempted by our system. Questions with more than one *entity.type* requests are fairly common and this sometimes results in confusion for our system especially if *entity.attribute* tags relate to different *entity.type* values. Since we do not attempt to disambiguate or link different *entity.attribute* tags to their corresponding *entity.type* values, this is often a source of error. Our constraint that forces all *entity.type* labels to come from one sentences mitigates this to some extent, but this is can still be a source of errors. *Q*4 is incorrect because the entity returned does not fulfill the location constraints of being close to the "bazar" while *Q*5 returns an incorrect entity type.

*Q*9 is a complicated question with strict location, budget, and attribute constraints, and the top-ranked returned entity "Hotel Pegasus Crown" fulfills the most requirements of the user[o].

Error analysis: We conducted a detailed error study on 105 of the test set questions and we find that approximately 60% of questions were not answered by our QA system pipeline due to limitations of the knowledge source while approximately, 40% of the recall loss in the system can be traced to errors in the semantic labels. See Table 8 for a detailed error analysis.

7.2 Understanding MSEQs in another domain

In contrast to methods that require tens of thousands of training data points, our question understanding framework works with a few hundred questions. We demonstrate the general applicability of our features and constraints by employing them on the task of understanding multi-sentence questions seeking *book* recommendations.

Using questions collected from an online book reading forum,[p] we annotated[q] 95 questions with their semantic labels. We retrained both CRF- and CCM-based supervised systems as before

[o] The hotel does not offer a spa and even with manual search we could not find a better answer.

[p] https://forums.onlinebookclub.org/

[q] Inter-annotator agreement measured on 30% of the data was 0.75.

Table 9. Labeling performance for book recommendation questions (paired *t* test, *p* value < 0.01 for aggregate *F1* in vanilla CRF and CCM model pairs & BiLSTM CRF and CCM model pairs)

Algorithm	F1 (type)	F1 (attr)	F1 (aggr)
CRF	41.5	42.1	41.8
CCM	52.1	43.8	47.9
BiLSTM CRF	52.6	39.9	46.3
BiLSTM CRF + Feat	54.6	45.1	49.9
BiLSTM CCM + Feat	55.9	44.6	50.3
BiLSTM BERT CRF	68.44	53.7	61.1
BiLSTM BERT CRF + Feat	**70.8**	52.0	61.4
BiLSTM BERT CCM + Feat	69.4	**55.8**	**62.6**

on this data set. Because location is not relevant for books, we use the two general labels: *entity.type* and *entity.attr*.

We train the labeler with no feature adaptation or changes from the one developed for tourism, retaining the same constraints as before. We tune the hyper-parameters with a grid search. Table 9 shows the performance of our sequence labeler over leave-one-out cross-validation. We find that that our generic features for *type* and *attr* defined earlier work acceptably well for this domain as well and we obtain $F1$ scores comparable to those seen for tourism. These experiments demonstrate that simple semantic labels can indeed be useful to represent multi-sentence questions and that such a representation is easily applicable to different domains.

8. Conclusion and future work

We have presented the novel task of understanding MSEQs. 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 BiLSTM 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 7 pt gain over baseline BiLSTM CRFs. The use of contextualized pretrained embeddings such as BERT result in an additional 6–8 pt improvement. We further demonstrate the strength of our work by applying the semantic labels toward a novel end-QA task that returns entity answers for MSEQs from a web API-based unstructured knowledge source that outperforms baselines. Further, we demonstrate how our approach allows rapid bootstrapping of MSEQ semantic parsers for new domains.

We see our paper as the first attempt toward end-to-end QA in the challenging setting of multi-sentence questions answered directly on the basis of information in unstructured and semi-structured knowledge sources. Our best model answers 54% of the questions with an Accuracy@3 of 57%. Our work opens up several future research directions, which can be broadly divided in two categories. First, we would like to improve on the existing system in the pipelined setting. 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. Currently, about 60% of recall loss is due to limitations in the knowledge source and query formulation, while a sizeable 40% may be addressed by improvements to question understanding.

As a second direction, we would like to train an end-to-end neural system to solve our QA task. This would require generating a large data set of labeled QA pairs which could perhaps be sourced semiautomatically using data available in tourism QA forums. However, answer posts in forums can often refer to multiple entities and automatically inferring the exact answer entity for the question can be challenging. Further, we would have to devise efficient techniques to deal with hundreds of thousands of potential class labels (entities). Comparing the performance of the pipelined model and the neural model and examining if one works better than the other in specific settings would also be interesting to look at.

We will make our training data and other resources available for further research.

Acknowledgments. We would like to thank Poojan Mehta, who designed and set up the annotation tasks on AMT, and Krunal Shah, who assisted with the evaluation of the QA system. We would also like to acknowledge the IBM Research India PhD program that enables the first author to pursue the PhD at IIT Delhi. We thank the reviewers for their feedback and Dinesh Raghu, Sachindra Joshi, Dan Weld, Oren Etzioni, Peter Clark, Gaurav Pandey, Dinesh Khandelwal for their helpful suggestions on early versions of this paper. We thank all AMT workers who participated in our tasks.

Financial Section. Mausam is supported by Google language understanding and knowledge discovery focused research grants, a Bloomberg award, grants by 1MG and a Microsoft Azure sponsorship. Parag Singla is supported by the DARPA Explainable Artificial Intelligence (XAI) Program with number N66001-17-2-4032. Mausam and Parag Singla are both supported by the IBM AI-Horizons Network Grant, IBM SUR awards as well as Visvesvaraya faculty awards by Govt. of India.

References

Abujabal A., **Yahya M.**, **Riedewald M. and Weikum G.** (2017). Automated template generation for question answering over knowledge graphs. *Pages 1191–1200 of: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3–7, 2017*.

Baker C.F., **Fillmore C.J. and Lowe**, **J.B.** (1998). The Berkeley FrameNet Project. *Pages 86–90 of: Proceedings of the 17th International Conference on Computational Linguistics—Volume 1, COLING '98*. 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. *Pages 1533–1544 of: 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.*

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. *Pages 1290–1295 of: 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*.

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

Bordes A., **Chopra S. and Weston J.** (2014b). Question answering with subgraph embeddings. *Pages 615–620 of: 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.

Bordes A., **Usunier N.**, **Chopra**, **S. and Weston**, **J.** (2015). Large-scale simple question answering with memory networks. *arxiv preprint arxiv:1506.02075.*

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

Chen D., **Fisch A.**, **Weston J. and Bordes A.** (2017). Reading wikipedia to answer open-domain questions. *Pages 1870–1879 of:* **Barzilay R. and Kan M.-Y.** (eds), *ACL (1)*. Association for Computational Linguistics.

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. *Pages 287–296 of: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM '16*. 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. *Pages 2580–2586 of: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16*. AAAI Press.

Contractor D., Mausam and Singla P. (2016). Entity-balanced gaussian PLSA for automated comparison. *Pages 69–79 of: Proceedings of NAACL-HLT*.

Das R., Zaheer M., Reddy S. and McCallum A. (2017). Question answering on knowledge bases and text using universal schema and memory networks. *Pages 358–365 of: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, Vancouver, Canada, July 30–August 4, Volume 2: Short papers.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. *Pages 4171–4186 of: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2019*, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (long and short papers).

Dua D., Wang Y., Dasigi P., Stanovsky G., Singh S and Gardner M. (2019). DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In: Proceedings of NAACL.

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

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

Gardner M., Grus J., Neumann M., Tafjord O., Dasigi P., Liu N.F., Peters M., Schmitz M. and Zettlemoyer, L.S. (2017). Allennlp: A deep semantic natural language processing platform.

Guo S., Liu K., He S., Liu C., Zhao J. and Wei Z. (2017). IJCNLP-2017 task 5: Multi-choice question answering in examinations. Pages 34–40 of: IJCNLP.

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

He L., Lee K., Lewis M. and Zettlemoyer L. (2017). Deep semantic role labeling: What works and what's next. *Pages 473–483 of: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, Vancouver, Canada, July 30–August 4, Volume 1: Long papers.

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

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 Daume 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. *Pages 1545–1556 of: 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.

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

Liang C.-C., Hsu K.-Y., Huang C.-T., Li C.-M., Miao S.-Y. and Su K.-Y. (2016). A tag-based statistical english math word problem solver with understanding, reasoning and explanation. *Pages 4254–4255 of: IJCAI*. IJCAI/AAAI Press.

Liang P.S. (2011). *Learning Dependency-Based Compositional Semantics*. PhD Thesis, 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. *Pages 1211–1220 of: Proceedings of the 26th International Conference on World Wide Web, WWW '17*. 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., Seaghdha D.Ó., Thomson B., Gasic M., Rojas-Barahona, L.M., Su P.H., Vandyke D., Wen T.-H. and Young S.J. (2016). Counter-fitting word vectors to linguistic constraints. *Pages 142–148 of: 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.

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

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

Palmer M., Hwa R. and Riedel S. (eds). (2017). *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, Copenhagen, Denmark, September 9–11, 2017. Association for Computational Linguistics.

Pazos R., Rodolfo A., Gonzalez B., Juan J., Aguirre L., Marco A., Martιnez F., Jose A., Fraire H. and Hector J. (2013). *Natural Language Interfaces to Databases: An Analysis of the State of the Art*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 463–480.

Pithyaachariyakul C. and Kulkarni A. (2018). Automated question answering system for community-based questions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Qiu X. and Huang X. (2015). Convolutional neural tensor network architecture for community-based question answering. *Pages 1305–1311 of: IJCAI*.

Rajpurkar P., Jia R. and Liang P. (2018). Know what you don't know: Unanswerable questions for squad. *Pages 784–789 of: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, Melbourne, Australia, July 15–20, 2018, Volume 2: Short papers.

Reddy S., Tackstrom 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.

Romeo S., Da San Martino G., Barron-Cedeno A., Moschitti A., Belinkov Y., Hsu W.-N., Zhang Y., Mohtarami M. and Glass J. (2016). Neural attention for learning to rank questions in community question answering. In *Proceedings of the 26th International Conference on Computational Linguistics*, Osaka, Japan.

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. *Pages 1466–1476 of: EMNLP*. 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*.

Singh, P. and Simperl E. (2016). Using semantics to search answers for unanswered questions in q&a forums. *Pages 699–706 of: Proceedings of the 25th International Conference Companion on world Wide Web*. International World Wide Web Conferences Steering Committee.

Srba I. and Bielikova M. (2016). A comprehensive survey and classification of approaches for community question answering. *ACM Transactions on the Web* **10**(3), 18:1–18:63.

Sun H., Ma H., Yih W.-T., Tsai C.-T., Liu J. and Chang M.-W. (2015). Open domain question answering via semantic enrichment. *Pages 1045–1055 of: Proceedings of the 24th International Conference on World Wide Web, WWW '15*. 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.A. (2016). Complex questions: Let me Google it for you. In *Proceedings of the Second Web QA Workshop WEBQA 2016*.

Wang D. and Nyberg E. (2015). CMU OAQA at TREC 2015 liveQA: Discovering the right answer with clues. *In: Proceedings of the Twenty-Fourth Text Retrieval Conference, TREC 2015*, Gaithersburg, Maryland, USA, November 17–20, 2015.

Wang D. and Nyberg E. (2016). CMU OAQA at TREC 2016 liveQA: An attentional neural encoder-decoder approach for answer ranking. In *Proceedings of the Twenty-Fifth Text Retrieval Conference, TREC 2016*.

Welbl J., Stenetorp P. and Riedel S. (2018). Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics* **6**, 287–302.

Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M. and Brew J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arxiv*, abs/1910.03771.

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. *Pages 1247–1256 of: Proceedings of the 2017 Conference on empirical methods in natural language processing, EMNLP 2017*, Copenhagen, Denmark, September 9–11, 2017.

Yang Z., **Qi P.**, **Zhang S.**, **Bengio Y.**, **Cohen W.W.**, **Salakhutdinov R. and Manning C.D.** (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yih W.-T., **Chang M.-W.**, **He X. and Gao J.** (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base. *Pages 1321–1331 of: ACL (1)*. 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. *Pages 1301–1310 of: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15*. New York, NY, USA: ACM.

Zettlemoyer, **L.S.** (2009). *Learning to Map Sentences to Logical Form*. PhD Thesis, 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. *Pages 533–542 of: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM '16*. New York, NY, USA: ACM.

Zhang X., **Wu J.**, **He Z.**, **Liu X. and Su Y.** (2018). Medical exam question answering with large-scale reading comprehension. *Pages 5706–5713 of: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI 18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, New Orleans, Louisiana, USA, February 2–7, 2018.

Zhao, **W.**, **Chung**, **T.**, **Goyal**, **A.K. and Metallinou**, **A.** (2019). Simple question answering with subgraph ranking and joint-scoring. *Pages 324–334 of: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (long and short papers).

Zheng W., **Yu J.X.**, **Zou L. and Cheng H.** (2018). Question answering over knowledge graphs: Question understanding via template decomposition. *Proceedings of the VLDB Endowment* **11**(11), 1373–1386.

Zhou J. and Xu W. (2015). End-to-end learning of semantic role labeling using recurrent neural networks. *Pages 1127–1137 of: 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**.

Cite this article: Contractor D, Patra B, Mausam and Singla P. Constrained BERT BiLSTM CRF for understanding multi-sentence entity-seeking questions. *Natural Language Engineering* https://doi.org/10.1017/S1351324920000017