Answering POI-Recommendation Questions using Tourism Reviews

Danish Contractor*
IBM Research AI & Indian Institute of Technology – Delhi
New Delhi, India
dcontrac@in.ibm.com

Krunal Shah†
Indian Institute of Technology – Delhi
New Delhi, India
ktgshah@gmail.com

Aditi Partap†
Indian Institute of Technology – Delhi
New Delhi, India
aaditi2@illinois.edu

Parag Singla
Indian Institute of Technology – Delhi
New Delhi, India
parags@cse.iitd.ac.in

Mausam
Indian Institute of Technology – Delhi
New Delhi, India
mausam@cse.iitd.ac.in

ABSTRACT

We introduce the novel and challenging task of answering Points-of-interest (POI) recommendation questions, using a collection of reviews that describe candidate answer entities (POIs). We harvest a QA dataset that contains 47,124 paragraph-sized user questions from travelers seeking POI recommendations for hotels, attractions and restaurants. Each question can have thousands of candidate entities to choose from and each candidate is associated with a collection of unstructured reviews. Questions can include requirements based on physical location, budget, timings as well as other subjective considerations related to ambience, quality of service etc. Our dataset requires reasoning over a large number of candidate answer entities (over 5300 per question on average) and we find that running commonly used neural architectures for QA is prohibitively expensive. Further, commonly used retriever-ranker based methods also do not work well for our task due to the nature of review-documents. Thus, as a first attempt at addressing some of the novel challenges of reasoning-at-scale posed by our task, we present a task specific baseline model that uses a three-stage cluster-select-rerank architecture. The model first clusters text for each entity to identify exemplar sentences describing an entity. It then uses a neural information retrieval (IR) module to select a set of potential entities from the large candidate set. A reranker uses a deeper attention-based architecture to pick the best answers from the selected entities. This strategy performs better than a pure retrieval or a pure attention-based reasoning approach yielding nearly 25% relative improvement in Hits@3 over both approaches. To the best of our knowledge we are the first to present an unstructured QA-style task for POI-recommendation, using real-world tourism questions and POI-reviews.

*The author is a PhD student at IIT Delhi and an employee of IBM Research.
†Work carried out when the author was a student at IIT Delhi

CIKM '21, November 1–5, 2021, Virtual Event, Australia.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482320

CCS CONCEPTS

• **Information systems → Question answering**; *Web searching and information discovery*.

ACM Reference Format:
Danish Contractor, Krunal Shah, Aditi Partap, Parag Singla, and Mausam. 2021. Answering POI-Recommendation Questions using Tourism Reviews. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482320

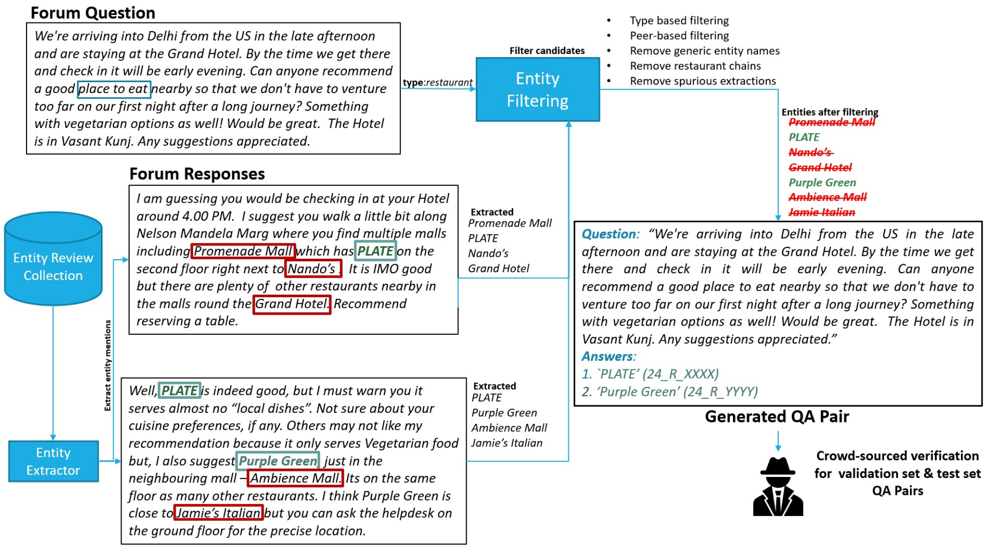
1 INTRODUCTION

According to a 2019 report[1] by Bain & Company, travellers make between $33-500$ web-searches before making bookings; some users consult in excess of 50 travel websites, spending a third of their time online conducting travel related activities. In some cases they may even post their questions on travel forums, in the hope of getting personalized travel information from other users. In 2016, TripAdvisor.com reported[2] over $900,000$ new topics being created on its travel forums annually.

Real-world questions, such as those seen on online forums are often verbose, requiring us to first determine what is crucial in the question for answering. For example, consider the forum question in Figure 1. Here the user describes *what they are looking for* (a restaurant) along with their *preferences* (vegetarian options *nearby*). They also mention where they stay and that they arrive in *"Delhi from the US in the late afternoon"*. The answer to this question would be the name of a restaurant that satisfies the requirements of the user. Answering such questions requires understanding and identifying the relevant parts of the question, reading information about each candidate answer entity in travel articles, blogs or reviews (*entity documents*), matching relevant question parts with entity documents, and ranking each candidate answer based on the degree of match.

In this paper we introduce the novel task of answering such Points-of-Interest (POI) recommendation questions using a collection of tourism reviews describing entities.[3] Our task reflects novel real-world challenges of reasoning at scale.

[1] https://www.bain.com/insights/todays-traveler-infinite-paths-to-purchase/
[2] https://www.tripadvisor.com/PressCenter-c4-Fact_Sheet.html
[3] We use the word 'entity' and 'POI' interchangeably in the rest of the paper.



We're arriving into Delhi from the US in the late afternoon and are staying at the Grand Hotel. By the time we get there and check in it will be early evening. Can anyone recommend a good place to eat nearby so that we don't have to venture too far on our first night after a long journey? Something with vegetarian options as well! Would be great. The Hotel is in Vasant Kunj. Any suggestions appreciated.

I am guessing you would be checking in at your Hotel around 4.00 PM. I suggest you walk a little bit along Nelson Mandela Marg where you find multiple malls including Promenade Mall which has PLATE on the second floor right next to Nando's . It is IMO good but there are plenty of other restaurants nearby in the malls round the Grand Hotel. Recommend reserving a table.

Well, PLATE is indeed good, but I must warn you it serves almost no "local dishes". Not sure about your cuisine preferences, if any. Others may not like my recommendation because it only serves Vegetarian food but, I also suggest Purple Green just in the neighbouring mall – Ambience Mall. Its on the same floor as many other restaurants. I think Purple Green is close to Jamie's Italian but you can ask the helpdesk on the ground floor for the precise location.

Question: *"We're arriving into Delhi from the US in the late afternoon and are staying at the Grand Hotel. By the time we get there and check in it will be early evening. Can anyone recommend a good place to eat nearby so that we don't have to venture too far on our first night after a long journey? Something with vegetarian options as well! Would be great. The Hotel is in Vasant Kunj. Any suggestions appreciated."*
Answers:
1. `PLATE' (24_R_XXXX)
2. 'Purple Green' (24_R_YYYY)

Figure 1: Entity Answers are extracted from forum post responses to generate QA Pairs. Entities marked in red indicate false positive extractions. Each entity in our collection has an ID of the form <city_id >_<POI type>_<number>. The dataset has three classes of POIs - restaurants (R), attractions (A) and hotels (H).

Challenges of Reasoning: Our task requires models to reason over entity reviews that could contain sarcasm, contradictory opinions etc, as well as, mentions of other entities (e.g., for comparison). Thus, the nature of reasoning for answering such questions is different from typical machine-reading comprehension [24, 37, 43], entailment-based reasoning or common-sense reasoning tasks [1, 8, 11, 21]. In addition, questions may also include requirements based on physical location, budget, timings as well as other subjective considerations about ambience, quality of service etc. Lastly, not all aspects of the question are relevant for answering which makes identifying the informational need challenging.

Challenges of Scalability: Questions have a large candidate answer space in our task since there may be thousands of POIs in a city, each represented by hundreds of reviews (e.g., New York has tens of thousands of restaurants to choose from). To address challenges of reasoning at scale in QA tasks, existing models, often employ retriever-ranker architectures which first reduce the search space by filtering documents using methods such as BM25 [39] ranking, and then apply deeper reasoning models on the reduced set. Additionally, methods may also use document structure to extract salient portions of the document or truncate documents to the first 800 – 1000 tokens [16, 25] to further improve scalability. However, as our experiments in Section 5 show, neither strategies are effective in our task.

Pruning the search-space based on TF-IDF scores does not work well because documents in our task, consist of opinions that are expressed in reviews; thus, they share a large common vocabulary, resulting in similar TF-IDF scores. In contrast, documents used in machine-reading comprehension [24, 43], entailment-based reasoning or common-sense reasoning tasks [1, 8, 21], etc have distinguishing terms (ex: due to topical entities), which help generate better TF-IDF scores. To further illustrate with an example, in our task, the average TF-IDF based inter-document cosine similarity for review documents of restaurants in New York is 0.35, while, for training data paragraphs in SQuAD [37], it is just 0.05. In addition, arbitrarily truncating review documents can cause a loss of crucial information. Thus, typical QA algorithms which apply cross-attention between question and candidate answer texts, do not scale in our task where entities may have long (bag-of-reviews) documents.

As a first attempt at addressing the novel challenges of reasoning at scale posed by our task, we present a scalable three-stage *cluster-select-rerank* model that serves as a strong baseline for this new task. It extends traditional retriever-ranker architectures by incorporating a clustering stage which first *clusters* text for each entity (independently) to identify exemplar sentences describing an entity. Then, similar to recent work on open domain QA [23], instead of employing retriever architectures based on sparse vector representations, it uses a neural information retrieval (IR) module with dense representations of questions and entities to *select* a set of potential entities from the large candidate set. Finally, it uses a *reranker* with a deeper attention-based architecture to pick the best answers from the selected entities.

1.1 Contributions

(1) We introduce a novel and challenging task of answering POI-recommendation questions using a collection of reviews describing entities. (2) We harvest a novel dataset[4] of tourism questions consisting of 47, 124 QA pairs extracted from online travel forums. Each QA pair consists of a question and an answer entity ID, which corresponds to one of the over 200, 000 entity review documents collected from the Web. (3) We present detailed experiments with

[4] Available at: https://github.com/dair-iitd/TourismQA

simple baselines such as those using BM25, POI-ratings, as well as, our task-specific model based on the cluster-select-rerank (CsrQA) architecture. We include detailed ablation studies highlighting the importance of each stage in the CsrQA pipeline. We find that the CsrQA approach serves as a strong baseline for future work on our task – it does better than a pure IR or a pure attention-based reasoning approach yielding nearly 25% relative improvement in Hits@3 over both approaches.

2 DATA COLLECTION

Most recent QA datasets have been constructed using crowdsourced workers who either create QA pairs given documents [37, 38] or identify answers for real world questions [26, 35]. Creating QA datasets manually using the crowd can be very expensive. We therefore choose to automatically harvest a dataset using tourism forums and a collection of reviews. We first crawled forum posts along with their corresponding conversation thread as well as meta-data including date and time of posting. We then also crawled reviews for restaurants and attractions for each city from a popular travel forum. Hotel reviews were scraped from a popular hotel booking website. Entity meta-data such as the address, ratings, amenities, etc was also collected where available.

Filtering Questions: We observed that apart from questions, forum users also post summaries of trips, feedback about services taken during a vacation, open-ended non entity-seeking questions such as queries about the weather and economic climate of a location, etc. We used high precision rules based on keywords and meta-data, to filter such posts (details available in released code due to lack of space). We further removed posts explicitly identified as "Trip Reports" or "Inappropriate" by the forum. Excessively long questions (≥ 1.7X more than average) were also removed because they were often other types of posts (eg: Complaints, Itineraries).

2.1 Answer Extraction

We create a list of entity names crawled for each city and use it to find entity mentions in user responses to forum posts. A high level entity class (hotel, restaurant, attraction) for each entity is also tagged based on the source of the crawl. If an entity could be associated with more than one label type, for instance a hotel that is also an attraction due to its heritage status, then the entities may independently occur in two different entity classes (as a 'hotel' and an 'attraction') along with an independent set of reviews. Each user response to a question is tagged for part-of-speech, and the nouns identified are fuzzily searched[5] in the entity list (to accommodate for typographical errors). This gives us a noisy set of "*silver*" answer entities, extracted from free-text user responses for each question. We now describe a series of steps aimed at improving the precision of extracted silver answers, resulting in our gold QA pairs (Summary in Figure 1).

2.2 Filtering of Silver Answer Entities

Type-based filtering: As a first step, we use the multi-sentence question understanding component developed by [13] to identify phrases in the question that could indicate a target entity's "*type*"and "*attribute*". For instance, in the example in Figure 1 tokens "*place to eat*" will be identified as an *entity.type* and the phrase "*has vegetarian options as well*" will be identified as *entity.attribute*.

All entities collected from the online forums come with labels (from a set of nearly 210 unique labels[6]) indicating the nature of the entity. For instance, restaurants have cuisine types mentioned, attractions are tagged as museums, parks etc. Hotels from the hotel booking website are simply identified as "hotels". We manually group the set of unique labels into 11 clusters.[7] For a given question we use the phrase tagged with the *entity.type* tag, and determine its closest matching cluster using embedding representations. Similarly, for each silver answer entity extracted we identify the most likely cluster based on frequency of cluster-hits of its list of meta-data attributes; if the two clusters do not match, we discard the QA pair. For example, in Figure 1, QA pairs that use entities *Promenade Mall* and *Ambience Mall* as answers, get discarded due to incorrect types.

Peer-based filtering: As mentioned previously, all entities have their type information (hotel, attraction or restaurant) indicated as part of meta-data. Using all *silver* (entity) answers for a question, we determine the frequency counts of each type encountered and remove any silver (entity) answer that does not belong to the majority type. For example, in Figure 1, the QA pair with entity *Grand Hotel* with type "hotel" is discarded because the majority type, based on its remaining peers, is "restaurant". If there is no clear majority type, the question is discarded (i.e, all QA pairs are discarded).

Filtering entities with generic names: Some entities are often named after cities, or generic place types – for example "The Cafe" or "The Spa" which can result in spurious matches during answer extraction. We collect a list of entity types[8] from Google Places[9] and remove any answer entity whose name matches any entry in this list.

Removing entities that are chains and franchises: Answers to questions can also be names of restaurants or hotel chains. However, without adequate information to identify the actual franchisee we cannot associate them to reviews. In such cases, our answer extraction returns all entities in the city with that (same) name. We thus, discard all such QA pairs.

Removing spurious candidates: User answers in forum posts often have multiple entities mentioned not necessarily in the context of an answer but for locative references (e.g., "opposite Starbucks", or "near Wendys") or for expressing opinions on entities that are not the answer. We write simple rules[10] to remove candidates extracted in such conditions (e.g.,: if more than one entity is extracted from a sentence, we drop them all or if entity mentions are in close proximity to phrases such as "next to", "opposite" etc. they are dropped).

Additionally, we review the set of entities extracted and remove QA pairs with entity names that were common English words or phrases (eg: "August", "Upstairs", "Neighborhood" were names of restaurants that could lead to spurious matches). We removed 322 unique entity names as a result of this exercise. Note that it is the

[5] Levenstein distance<0.05

[6] List available in released code

[7] Determined empirically

[8] Examples of types include "cafe", "hospital", "bar" etc.

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

[10] available in released code

Table 1: QA Pairs in train, validation and test sets for each answer entity type

	#Ques.	QA pairs	Tokens per ques.	QA Pairs (Hotels)	QA Pairs (Restr.)	QA Pairs (Attr.)
Training	18,531	38,586	73.30	4,819	30,106	3,661
Validation	2119	4,196	70.67	585	3267	335
Test	2,173	4,342	70.97	558	3,418	366

only step that involved human annotation in the data collection pipeline thus far.

2.3 Data Collection: Error Analysis

We studied 450 QA pairs of the train-set, representing approximately 1% of the dataset, for errors in the automated data collection process. We found that our high precision filtering rules have an answer extraction accuracy of 82% on this set. The errors can be traced to one of four major causes (i) (16%) Entity name was a generic English word (e.g., "The Park") (ii) (27%) Entity matched another entity in the answer response which was not intended to be the answer entity to the original question. (e.g., Starbucks in "next to Starbucks") (iii) (31%) Entity matched another entity with a similar name but of a different target class (e.g., hotel with same name instead of restaurant). (iv) (13%) Failing to detect negations/negative sentiment (e.g., an entity mention in a post where the user says "*i wouldn't go there for the food*". (v) The remaining 13% of the errors were due to errors such as invalid questions (non-entity seeking), or incorrect answers provided by the forum users.

Due to the large candidate space it is infeasible to manually annotate each candidate with respect to a question. However, we note that the extraction accuracy is comparable to that seen in some existing datasets such as TriviaQA [22] and is useful for training.

2.4 Crowd-sourced Data Cleaning

As our error study in the previous section shows, our automated QA pair extraction methods are likely to have some degree of noise. In order to facilitate accurate bench-marking, we crowd-source and clean our validation and test sets. We use Amazon Mechanical Turk (AMT[11]) for crowd-sourcing. Workers are presented with a QA-pair, which includes the original question, an answer-entity extracted by our rules and the original forum-post response thread where the answer entity was mentioned. Workers are then asked to check if the extracted answer entity was mentioned in the forum responses as an answer to the user question. We spend $0.05 for each QA pair costing a total of $550. The crowd-sourced cleaning was of high quality; on a set of 280 expert annotated question-answer pairs, the crowd had an agreement score of 97%. As a result of the crowd-sourced cleaning, out of a total of 10, 895 QA pairs across the validation and test sets, 21.64% of the QA pairs were discarded indicating that our high precision rules for generating QA pairs have an answer extraction accuracy of 78.36%. The resulting dataset is summarized in Table 1. We note that since workers are only asked to assess the extracted answers, our QA dataset is likely to contain false negatives, i.e. candidates that may be valid answers for a question but are not extracted by our automated methods (or are not mentioned by forum users in posts) as answers. However, due to the large candidate space it is infeasible to manually annotate each candidate with respect to a question. Thus, our task also shares challenges seen in evaluating recommendation systems [42] where the relevance judgements are sparse and incomplete (unlike traditional IR tasks). We discuss the impact of partial relevance judgements in more detail in Section 5.

Table 2: Summarized statistics: Knowledge source consisting of 216, 033 entities and their reviews

Avg # Tokens	3266
Avg # Reviews	69
Avg # Tokens per Review	47
Avg # Sentences	263

Table 3: Classification of Questions - a qualitative study on 100 random samples. (%) does not sum to 100; Questions may exhibit more than one feature.

Feature	%	Examples of Phrases in Questions
Budget constraints	23	*good prices,* *money is a bit of an issue* *maximum of $250 ish in total*
Temporal elements	21	*play ends around at 22:00 (it's so late!)* *.. dinner before the show,* *theatre for a Saturday night* *open christmas eve*
Location constraint	41	*dinner near Queens Theatre,* *staying in times square;would like it close,* *options in close proximity (walking distant)* *easy to get to from the airport*
Example entities mentioned	8	*found this one - Duke of Argyll* *done the Wharf and Chinatown,* *no problem with Super 8*
Personal preferences	61	*something unique and classy,* *am not much of a shopper,* *love upscale restaurants,* *avoid the hotel restaurants,* *Not worried about eating healthy* *out with a girlfriend for a great getaway*

2.5 Data Characteristics

In our dataset, the average number of tokens in each question is approximately 73, which is comparable to the document lengths for some existing QA tasks. Additionally, our entity documents are larger than the documents used in existing QA datasets – they contain 3, 266 tokens on average. Lastly, answering any question requires studying all the possible entities in a given city – the average number of candidate answer entities per question is more than 5, 300, which further highlights the challenges of scale.

Our dataset contains QA pairs for 50 cities. The total number of entities in our dataset is 216, 033. Details about the knowledge source are summarized in Table 2. In almost every city, the most common entity class is restaurants. On average, each question has 2 gold answers extracted. Questions can include requirements based on physical location, budget, timings as well as other *subjective* considerations related to ambience, quality of service etc. A qualitative study of 100 random questions suggests that 61% of the questions contain personal preferences of users, 23% of the questions contain budgetary constraints, while 41% contain locative constraints (Table 3).

[11] http://requester.mturk.com

3 RELATED WORK

Given a POI-recommendation question (as in Figure 1) its target class (restaurant), the city (Delhi), a candidate space of target-class entities (POIs) for each corresponding city, and a collection of reviews describing the entities, the goal of our task is to score each candidate with respect to a question, for relevance.

POI-Recommendation Tasks: Existing work on POI recommendation relies on the use of structured [5, 27] or semi-structured data [3, 17] to offer personalized recommendations to users. For instance, work such as [17] extracts aspect-polarity sentiments from reviews and models users using their past POI-visits to make recommendations. Other approaches include those using user data and social-influence graphs [27], spatial-features [44, 46], temporal features [20, 44, 46] and opinions [3, 17] to make personalized recommendations to users. Queries are often structured or consist of simple keywords or phrases[6, 17, 45]. In contrast, we pose the problem as a Question-Answering task where a user provides a detailed question describing their preferences and constraints; the user provides no additional background or historical record. The system needs to return answers (POIs) by analyzing the question, as well as, a collection of unstructured review documents (associated with each POI). To the best of our knowledge we are the first to formulate and present an unstructured QA-style task for POI-recommendation, using real-world tourism questions and POI-reviews.

QA and IR Tasks: Question answering tasks such as those based on reading comprehension require answers to be generated either based on a single passage, or after reasoning over multiple passages (or small-sized documents) (e.g. *SQuAD* [37], *HotpotQA* [43], *NewsQA* [41]). Answers to questions are assumed to be stated explicitly in the documents [37] and can be derived with single or multi-hop reasoning over sentences mentioning facts [43]. In contrast, in our task, answers are entities represented by documents (review documents). Other variants of existing QA tasks add an additional layer of complexity where the document containing the answer may not be known and needs to be retrieved from a large corpus before answers can be extracted/generated (e.g. *SearchQA* [15], *MS MARCO* [35], *TriviaQA* [22]). Models for these open-QA tasks typically use retriever-ranker architectures based on sparse vector representations like TF-IDF and BM25 ranking [39] to retrieve and sub-select candidate documents [7]; deeper reasoning is then performed over this reduced space to return answers for scalability. However, we find that in our task, retrieval strategies such as BM25 perform poorly[12] and are thus not effective in reducing the candidate space (see Section 5). As a result, our task requires processing 500 times more documents per questions and also requires reasoning over large entity review-documents that consist of noisy, subjective opinions. Further, traditional QA models such as BiDAF [40] or those based on BERT [14] are infeasible[13] to train for our task. Thus, while existing tasks and datasets have been useful in furthering research in comprehension, inference and reasoning, we find that they do not always reflect the complexities of real-world question answering motivated in our task. We note that our work is also related to QA tasks defined for "Community Question-Answering (CQA)"[19]. However, in contrast to CQA tasks aimed at fetching existing answers from forum threads or finding similar questions on a forum[18, 19], in our task, tourism POI-recommendation questions are answered by returning entity answers using a collection of reviews describing entities.

Our QA task is one that also shares characteristics of information retrieval (IR), because, similar to adhoc document retrieval, answers in our task are associated with long entity documents, though they are without any additional structure. The goal of IR, specifically document retrieval tasks, is to retrieve documents for a given query. Neural models for IR focus on identifying dense representations for queries and documents to maximize mutual relevance in latent space [23, 32, 33]. To improve dealing with rare words, neural models also incorporate lexical matching along with semantic matching [34]. However, unlike typical retrieval tasks, the challenge for answering in our task is not merely that of semantic gap – subjective opinions need to be *reasoned* over and aggregated in order to assess relevance of the entity document. This is similar to other reading comprehension QA tasks that require deeper reasoning over text, but in our task, such deeper reasoning is in a retrieval task.

In this paper we use a coarse-to-fine architecture that sub-selects documents using dense representations from neural IR and then uses a deep reasoner over the selected subset (Section 4) to re-rank the entities (represented by documents).

4 THE CLUSTER-SELECT-RERANK MODEL

A model built for our task needs to address its novel challenges of reasoning at scale. As mentioned previously, each entity in our task is represented by a long, bag-of-reviews document; existing approaches, such as arbitrarily truncating documents to reduce length [22], are not appropriate due to the lack of structure. Further, there are thousands of candidate entities for each question and reducing the candidate search space using TF-IDF style methods [7] do not work well due to the nature of review documents (reviews express opinions about similar aspects/topics as opposed to the distinct, informative topics seen in typical QA/IR tasks). Finally, answering a question requires deep reasoning over the document for each candidate and the challenges of scale make the application of existing models intractable.

Our proposed baseline for this task consists of three major components designed to address these challenges: (1) a **clustering** module to generate representative entity documents that are smaller in size (in terms of document length), (2) a fast scalable neural retrieval model that uses dense representations of questions and entities to **select** candidate entities to reduce the search space, and (3) a QA-style **re-ranker** that reasons over the selected answers and scores them to return the final top-ranked answers. We refer to it as CsrQA and now describe each component in detail.

4.1 *Cluster:* Representative Entity Document Creation

As stated previously, entity documents in our dataset are much larger than documents used by previous QA tasks. In order to make training a sufficiently expressive neural model tractable, CsrQA

[12] Hits@3 of 7%

[13] BiDAF requires 43 hours for 1 epoch (4 K-80 GPUs)

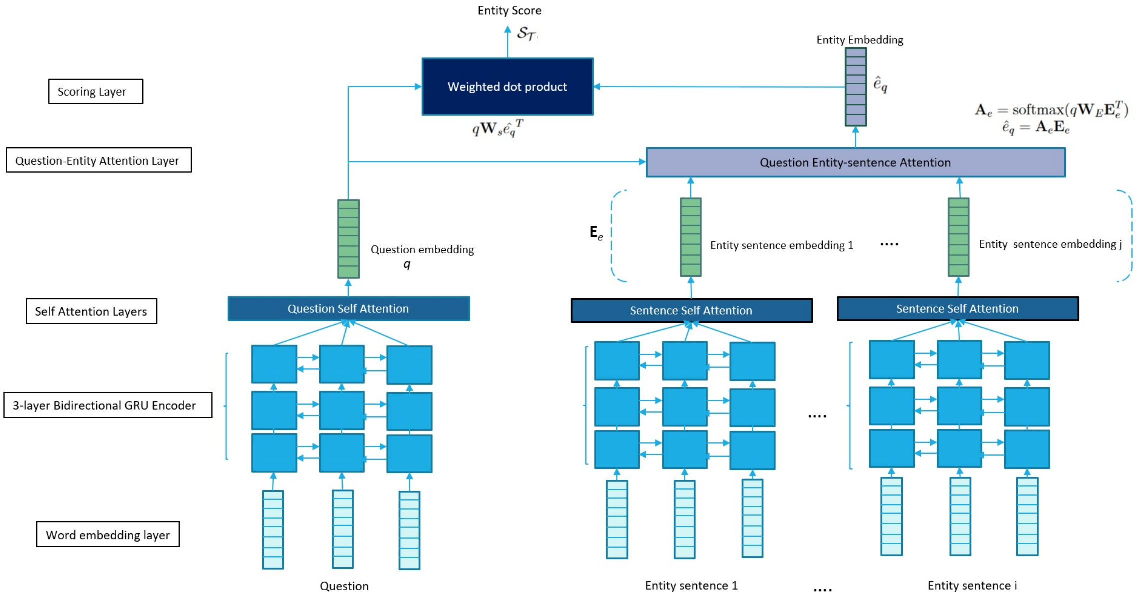


Figure 2: Reasoning network used to re-rank candidates shortlisted by the Duet model.

first constructs smaller representative documents[14] for each entity using the full entity documents (containing all reviews for an entity). It encodes each review sentence using the pre-trained universal sentence encoder (USE) [4] to generate sentence embeddings. It then cluster sentences within each document using k-means clustering and uses the top-k (nearest to the cluster centroid) sentences from each cluster to represent the entity. In our experiments we use k = 10 and generate 10 clusters per entity, thus reducing our document size to 100 sentences each. This constitutes an approximately 70% reduction in document size but we note that our documents are still larger than those used in most QA tasks.

4.2 *Select:* Shortlisting Candidate Answers

In this step, CsrQA trains a neural retrieval model with the question as the query and representative entity documents as the text corpus. As its retrieval model, it uses the recently improved Duet network [33]. Duet is an interaction-based neural network that compares elements of the question with different parts of a document and then aggregates evidence for relevance. It uses both local as well as distributed representations to capture lexical and semantic features. It is quite scalable for our task, since its neural design is primarily based on CNNs.

The local distributed representations are created using a term-document matrix (interaction features), which contains inverse-document-frequency scores of words, for each term-position in the document. This matrix is passed through a convolution network and its output is fed through a series of fully connected linear layers to return a score. We retain the default hyper-parameters as specified in [33].

The distributed model uses the Glove vector embeddings [36] to create vector representations, for words questions and entity documents. These are then independently passed through convolution layers with window-based max-pooling. The question representations are fed to a fully-connected linear layer while the entity document representations are further encoded using another convolution layer. The representations from the question and the entity document are combined together and then jointly processed using a series of fully connected linear layers to return a score. The score from the local and distributed model are added to return the final score. Please see Mitra et al.'s paper for more details.

Duet is trained over the QA-pair training dataset and 10 randomly sampled negative examples and uses cross-entropy loss. Duet can be seen as ranking the full candidate answer space for a given question, since it scores each representative entity document. CsrQA selects the top-30 candidate entities from this ranked list for a deeper reading and reasoning, as described in the next section.

4.3 *Rerank:* Answering over Selected Candidates

In this step, our goal is to perform deeper reading and reasoning over the shortlisted candidate answers to build the best QA system. The CsrQA implements a model for re-ranking using recurrent encoding and attention-based matching.

Input Layer: It uses 128-dimensional word2vec embeddings [31] to encode each word of a question and a representative entity document. It uses a three layer bi-directional GRU [10], which is shared between the question and the review sentence encoder.

Self Attention Layer: It learns shared self-attention (intra-attention) weights [9] for questions and representative entity documents and generates attended embedding representations for both. Let the hidden state of the sequence (question or entity sentence) be given

[14] a representative document for an entity is a set of sentences selected from the full set of reviews for an entity

by matrix $\mathbf{H}$ where the ith hidden state is represented by h_i. Then the attended representation (s) of the sequence is given by

$$\mathbf{A} = \text{softmax}(v_a \tanh(\mathbf{W}_a \mathbf{H}^T)) \quad \text{and} \quad s = \mathbf{AH} \tag{1}$$

where $\mathbf{A}$ is the attention matrix, $\mathbf{W}_a$ and v_a are attention parameters. We generate attended representations for both, the question as well as, for each sentence from the representative entity document. Let q be the attended representation of the question and let $\mathbf{E}_e$ be attended representation of the entity sentences as a matrix .

Question-Entity Attention (QEA) Layer: In order to generate an entity embedding, it attends over the entity sentence embeddings in matrix $\mathbf{E_e}$ with respect to the question [29]. This helps identify "important" sentences and the sentence embeddings are then combined based on their attention weights to create the entity embedding (which are thus, question-dependent). The question attended entity representation $\hat{e}_q$ is thus given by:

$$\mathbf{A}_e = \text{softmax}(q\mathbf{W}_E\mathbf{E}_e^T) \quad \text{and} \quad \hat{e}_q = \mathbf{A}_e\mathbf{E}_e \tag{2}$$

where $\mathbf{A}_e$ is the entity-sentence attention matrix and $\mathbf{W}_E$ is a parameter matrix.

Scoring Layer: Finally, given a question representation q and the entity embedding ($\hat{e}_q$), the model uses a weighted dot product between the two vectors q, $\hat{e}_q$ to generate the final score $\mathcal{S}_{\mathcal{T}}$, and is given by $q\mathbf{W}_s\hat{e_q}^T$ ($\mathbf{W}_s$ is a parameter). The model is summarized in Figure 2. The network is trained using hinge loss by sampling 10 negative (incorrect answer) entities for each question-answer pair.

5 EXPERIMENTS

We ask the following questions in our experiments: (1) What is the performance of the CsrQA model compared to other simpler baselines for this task? (2) How does the CsrQA baseline compare with neural IR and neural QA models? (3) What is the impact of false negatives? (4) What is the effect of the size of candidate space? (5) What is the performance of the system across different answer entity types (classes)?

5.1 Models for comparison

We began by trying to adapt traditional reading comprehension QA models such as BiDAF [40] for our task, but we found they were infeasible to run – just 1 epoch of training using 10 negative samples per QA pair, and our representative entity documents,[15] took BiDAF over 43 hours to execute on 4 K-80 GPUs. Running a trained BiDAF model (forward-pass) on our test data using the same system configuration would take even longer and was projected to require over 9 days. Similarly, we also tried using models based on BERT [14] on our representative entity documents, but again, it did not scale for our task. In the absence of obvious scalable QA baselines, we compare the performance of CsrQA with other task-specific baselines.

Random Entity Baseline: Returns a random ranking of the candidate answer space.

Ratings Baseline: Returns a global (question-independent) ranking of candidate entities based on user review ratings of entities.

BM25 Retrieval: We index each entity along with its reviews into Lucene.[16] Each question is transformed into a query using the default query parser that removes stop words and creates a disjunctive term query. Entities are scored and ranked using BM25 ranking [39]. Note that this baseline is considered a strong baseline for information retrieval (IR) and is, in general, considered better or at par with many neural IR models for typical IR tasks [30].

Review-AVG Model: It uses averaged vector embeddings of the review sentences to represent each document - we use universal sentence embeddings (USE) [4] to pre-compute vector representations for each sentence and average them to create a document representation. Questions are encoded using a self-attended bi-directional GRU [9] to generate a question representation. An entity is scored via a weighted dot product between question and document embeddings.

5.1.1 *Ablation Models.*

RsrQA: This model highlights the value of the clustering step and the creation of representative entity documents. We replace the clustering phase of our CsrQA model and use 100 randomly-selected review-sentences to represent entities. Thus, this model is effectively CsrQA, but without clustering.

We also tried to create a model that creates document representations by selecting 100 sentences from an entity document by indexing them in Lucene and then using the question as a query. However, this method, understandably, returned very few sentences – the questions (query) are longer than a sentence on average and the lexical gap is too big to overcome with simple expansion techniques. Lastly, if we give the full entity document instead of a representative one, the neural select-rerank model cannot be trained due to GPU memory limitations.

CsQA : This model returns answers by running the neural information retrieval model, Duet, on the clustered representative documents. This model is effectively CsrQA, but without re-ranking.

CrQA : This model returns answers by running the reasoner directly on the clustered representative documents. Thus, this model does not use neural IR to select and reduce the candidate search space. This model is effectively CsrQA, but without selection.

CrQA without QEA : Instead of generating entity embeddings by question attention over entity-sentences (as in Equation 2), we could also generate question-independent, self-attended, entity embeddings (e) as given by:

$$\mathbf{A'}_e = \text{softmax}(v_E \tanh(\mathbf{W'}_E\mathbf{E}_e^T)) \quad \text{and} \quad e = \mathbf{A'}_e\mathbf{E}_e \tag{3}$$

where $\mathbf{A'}_e$ is the entity-sentence attention matrix and $\mathbf{W'}_E$, v_E are parameters.

5.1.2 *Hyper-parameter Settings.*

For all experiments we set $\delta = 1$ in our max-margin criterion. We used Adam Optimizer [28] with a learning rate of 0.001 for training. The convolution layers in the Duet model (retriever) used kernel sizes of 1 and 3 for local and distributed interactions respectively. Hidden nodes were initialized with size of input word embeddings, 128 dimensions. The reasoning network (re-ranker) was trained for 5 days on 6 K80 GPUs (approx. 14 epochs) using 10 negative samples for each QA pair. We used 3-layer 128-dimensional bidirectional GRUs to encode questions

[15] the smaller-size documents created after clustering sentences

[16] *http://lucene.apache.org/*

Table 4: Performance of different systems including the CsrQA model on our task. Hits@N reported in % (t-test p-value<0.0005).

Method	Hits@3	Hits@5	Hits@30	MRR
Random	0.32	0.58	3.78	0.007
Ratings	0.37	0.92	3.33	0.007
BM25	6.72	9.98	30.60	0.071
Review-AVG	7.87	11.83	30.65	0.084
RsrQA	10.22	14.63	36.99	0.104
CrQA	16.89	23.75	52.51	0.159
CrQA without QEA	14.91	19.97	47.58	0.141
CsQA	17.25	23.01	52.65	0.161
CsrQA	**21.44**	**28.20**	**52.65**	**0.186**

and review sentences. Input word embeddings were updated during training and USE embeddings returned 512 dimension embeddings. While training the reasoning network (re-ranker) takes 11.5 hours per epoch on 4 K-80 GPUs, executing only the forward pass on our development and test sets that use the full-answer space takes 3.5 days. Note that since CsrQA is a pipelined model, its components for selection and re-ranking are trained independently, and both components use the representative entity documents during training.

5.2 Metrics for Model evaluation

The goal of our task is to return an entity (represented by a document) as the answer to a user question. Since our relevance judgements are incomplete and un-ranked, we only assess the relevance of a candidate answer, regardless of whether or not, there could be multiple better ranked answers.

Hits@N : For a question q_i, let the set of top ranked N entities returned by the system be E^i_N, and let the set of correct (gold) answer entities for the question be set G^i; the aggregated *Hits@N* is then given by, $\frac{\sum_{i=1}^{M} \mathbb{1}((E^i_N \cap G^i) \neq \phi)}{M}$, where M is total number of QA-pairs evaluated. Intuitively, this metric assigns a score if *any* of the top-N answer entities returned for a question are correct.

Mean Reciprocal Rank (MRR): In addition, we report MRR where we consider only the highest ranked gold answer (if multiple gold answers exist for a question).

5.3 Results

Table 4 compares CsrQA against other models. We find that all non-neural baselines perform poorly on the task. Even the strong baseline of BM25 retrieval, which is commonly used in retrieval tasks, is not as effective for this dataset. Methods such as BM25 are primarily aimed at addressing challenges of semantic gap while in our task, answers require *reasoning* over subjective opinions in entity documents. We also observe that the performance of the neural model, Review-AVG, is comparable to that of BM25.

The RsrQA model that uses randomly sampled review-sentences to represent entity-documents, has a low Hits@3 of 10.22%. In contrast, both the CsQA and CrQA models, that use the clustered representative entity-documents have higher scores than RsrQA. This highlights the value of creating smaller-sized representative documents using clustering. Our final proposed baseline for this task, CsrQA, has an Hits@3 of approximately 21.44% (last row of Table 4).

Table 5: Performance of different systems including the CsrQA model on our task as measured using human judgements (Human Scores) and gold-reference data (Machine Scores) on 100 questions from the validation data. Scores reported in %.

	Human Scores	Machine Scores
Method	Hits@3	Hits@3
CrQA	50.0	19.79
CsQA	63.51	22.92
CsrQA	**65.63**	**33.33**

We also find that CsrQA does better than CrQA. We hypothesize that since training the reasoner is compute-intensive, it is unable to see many hard negative samples for a question even after a long time of training. As a result, it optimizes its loss on the negatives seen during training, but may not perform well when the full candidate set is provided at test-time. However, when the reasoner is used for re-ranking in CsrQA (at test time), the *select* module first shortlists good candidates and the reasoner's job is then just limited to finding the best ones from the small set of relatively good candidates.

We also note that the CrQA model without the QEA layer suffers a significant deterioration in performance as building question-specific entity embeddings probably helps the model focus on the salient information necessary to answer a question.

Comparing CsrQA & CsQA suggests that, while the scalable matching of CsQA is useful enough for filtering candidates, it is not good enough to return the best answer.

Overall, we find that each component of the CsrQA baseline is critical in its contribution towards its performance on the task. Moreover, strong IR only (CsQA) and QA only baselines (CrQA) are not as effective as their combination in CsrQA.

Effect of False Negatives: Since the dataset contains incomplete relevance judgements (false negatives), our metrics may under-report system performance. Thus, we assess the actual system performance using a blind human-study, and also assess, whether metrics computed using the gold-entity answers as reference answers (machine scores), correlate with human relevance judgements (human scores), on the top-3 answers returned by a system. We conduct a blind human-study using the CsQA, CrQA and CsrQA models on a subset of 100 randomly selected questions (300 output pairs) from the validation-set. Two human evaluators (κ=0.79) were presented the top-3 answers from each model in random order and were asked to mark each answer for relevance – we ask the evaluators to manually query a web-search engine and assess if each question-recommendation pair (returned by a model) adequately matches the requirements of the user posting that question. As can be seen from Table 5, the absolute performance of the systems as measured by the human annotators is higher indicating the presence of false negatives in the dataset. Thus, the machine scores under-report actual performance. To assess whether performance improvements measured using our gold-data (machine scores) correlate with human judgements, we compute pair-wise correlation coefficients between CsrQA, CsQA and CrQA. We find that there is moderately positive

Table 6: Performance (Hits@3 in %) on test-set questions with different candidate answer space sizes.

Candidate Space Size	No. of Questions	CsQA	CrQA	CsrQA
<=1000	631	28.69	30.27	**32.49**
>1000	1542	12.58	11.41	**16.93**

correlation [2] with high confidence between the human judgements and gold-data based measurements for Hits@3 ($\bar{\rho}$ = 0.39, p-value<0.0009) as well as on Hits@5 ($\bar{\rho}$ = 0.32, p-value<0.04). The machine-scores are thus, useful to benchmark models despite the presence of false negatives in the dataset.

Error Analysis: We conducted an error analysis of the CsrQA model using 100 questions used during the human evaluation. We found that nearly 35% of the errors made were on questions involving location constraints while, 9% of the errors were due to either budgetary or temporal constraints not being satisfied. This suggests that models which additionally incorporate reasoning over geo-spatial information or incorporate numeric reasoning (eg: for budgets) may be helpful for this task. The remaining 65% of the errors collectively constitute not fulfilling user preferences of cuisine, age appropriate and/or celebration activities, hotel preferences, etc, suggesting a large scope for improvement in reasoning models. We now present a detailed study of the answering characteristics of the system.

5.4 QA System Answering Characteristics

Effect of candidate space size: Table 6 breaks down the performance of systems based on size of the candidate space encountered while answering. In questions where the candidate space is relatively smaller (<1000), we find CrQA model has slightly better performance than the CsQA model. However, in large candidate spaces we find the CsQA model is more effective in pruning the candidate search space and performs better than the CrQA model. The CsrQA model outperforms both systems regardless of candidate space size, highlighting the benefit of our method.

Performance across different entity-classes: Questions on restaurants dominate the dataset (Table 1) and they also have a larger candidate space, with 1, 501 questions in the test set having a search space greater than 1, 000 candidates. In this sub-class of questions, the CsQA model, which does not do deep reasoning, answers more questions correctly in the top-3 ranks, as compared to the CrQA model (Hits@3 12.65% and 11.1% respectively). On the other hand, we find that in hotels and attractions the search space in most questions isn't as large, and both the CsQA and CrQA models have comparable performance. However, CsrQA outperforms both systems regardless of entity class (relative gain of 5–32% in Hits@3).

Effect of the size of space for re-ranking: The performance improvement of the CsrQA model over the CrQA model suggests that the re-ranker gets confused as the set of candidate entities increases. We studied the performance of the CsrQA model by varying the number of candidates it had to re-rank (Table 7). As expected, as we increase the number of candidates available for re-ranking, the Hits@3 begins to drop finally settling at approximately 15% (on validation data) when the full candidate space is available. However, we find that the variation in Hits@3 is small, indicating that there are only a few candidates (approx. 30-40) per question that the model is most confused about. Thus, since max-margin ranking models can be sensitive to the quality of negative samples, identifying harder candidates and making them available at training, could help models better distinguish between candidates.

Table 7: Performance of CsrQA on the validation data reduces, as the size of candidate space (selected by CsQA) to be re-ranked increases.

top-k	Hits@3	Hits@5	Hits@30	MRR
10	19.39	25.86	33.88	0.160
20	**19.53**	**26.85**	47.33	0.171
30	19.01	26.66	54.32	0.171
40	18.59	26.76	57.24	**0.172**
50	18.68	26.85	57.95	0.171
60	18.64	25.77	58.66	0.169
80	18.26	25.34	**58.94**	0.169
100	18.26	25.02	58.75	0.167
Full	14.67	21.43	53.56	0.147

6 CONCLUSION

In the spirit of defining a question answering challenge that is closer to a real-world QA setting, we introduce the novel task of returning a POI recommendation for given user question based on a collection of unstructured reviews describing entities. We harvest a dataset of over 47,000 QA pairs, which enables end to end training of models. Due to the nature of questions and the review documents, one of the biggest challenges in this dataset is that of scalability. Our task requires processing 500 times more documents per question than most existing QA tasks, and individual documents are also much larger in size. We thus present a cluster-select-rerank architecture based method that brings together neural IR and QA models, and serves as a strong baseline for this task. We believe that further research on this task will significantly improve the state-of-the-art in question answering. For instance, neuro-symbolic methods that additionally reason on locative and budgetary constraints could be an interesting direction of future work as they occur in nearly 64% of the questions in our dataset. A recent paper on spatial constraints in QA [12] is based on this work and dataset. While our final system registers a 25% relative improvement over other simpler baseline models, a correct answer is in top-3 for only 21% of the questions, which points to the difficulty of the task. An interesting extension to our baseline CsrQA model could be to make the clustering step question-dependent – this could help representative entity documents better reflect important information for each question.

7 ACKNOWLEDGEMENTS

We would like to thank Yatin Nandwani, Sumit Bhatia, Dhiraj Madan, Dinesh Raghu, Sachin Joshi, Gaurav Pandey, Dinesh Khandelwal for their helpful suggestions during the course of this work. We would also like to thank Shashank Goel for re-implementing the data collection scripts. We would also like to acknowledge the IBM Research India PhD program that enables the first author to pursue the PhD at IIT Delhi. This work was supported by an IBM AI Horizons Network (AIHN) grant, IBM SUR awards, Visvesvaraya faculty awards by Govt. of India to both Mausam & Parag, the Jai Gupta Chair fellowship and grants by Google, Bloomberg & 1MG to Mausam. We thank the IITD HPC facility for computing resources.

REFERENCES

[1] Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2019. ComQA: A Community-sourced Dataset for Complex Factoid Question Answering with Paraphrase Clusters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 307–317. https://doi.org/10.18653/v1/N19-1027

[2] Haldun Akoglu. 2018. User's guide to correlation coefficients. *Turkish Journal of Emergency Medicine* 18 (2018), 91 – 93.

[3] Ramesh Baral, XiaoLong Zhu, SS Iyengar, and Tao Li. 2018. Reel: Review aware explanation of location recommendation. In *Proceedings of the 26th conference on user modeling, adaptation and personalization*. 23–32.

[4] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *CoRR* abs/1803.11175 (2018). arXiv:1803.11175 http://arxiv.org/abs/1803.11175

[5] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. 2020. Relevance Models for Multi-Contextual Appropriateness in Point-of-Interest Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).

[6] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. 2020. Relevance Models for Multi-Contextual Appropriateness in Point-of-Interest Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1981–1984. https://doi.org/10.1145/3397271.3401197

[7] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Association for Computational Linguistics (ACL)*.

[8] Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An Adversarially-Authored Question Answering Dataset for Common Sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*. 63–69.

[9] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long Short-Term Memory-Networks for Machine Reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 551–561. https://doi.org/10.18653/v1/D16-1053

[10] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[11] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *NAACL*.

[12] Danish Contractor, Shashank Goel, Mausam, and Parag Singla. 2021. Joint Spatio-Textual Reasoning for Answering Tourism Questions. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 1978–1989. https://doi.org/10.1145/3442381.3449857

[13] Danish Contractor, Barun Patra, Mausam, and Parag Singla. 2021. Constrained BERT BiLSTM CRF for understanding multi-sentence entity-seeking questions. *Nat. Lang. Eng.* 27, 1 (2021), 65–87. https://doi.org/10.1017/S1351324920000017

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *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)*. 4171–4186. https://aclweb.org/anthology/papers/N/N19/N19-1423/

[15] Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *CoRR* abs/1704.05179 (2017). arXiv:1704.05179 http://arxiv.org/abs/1704.05179

[16] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Association for Computational Linguistics, Hong Kong, China, 1–13. https://doi.org/10.18653/v1/D19-5801

[17] Qing Guo, Zhu Sun, Jie Zhang, Qi Chen, and Yin-Leng Theng. 2017. Aspect-Aware Point-of-Interest Recommendation with Geo-Social Influence. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization* (Bratislava, Slovakia) *(UMAP '17)*. Association for Computing Machinery, New York, NY, USA, 17–22. https://doi.org/10.1145/3099023.3099066

[18] Shahar Harel, Sefi Albo, Eugene Agichtein, and Kira Radinsky. 2019. Learning Novelty-Aware Ranking of Answers to Complex Questions. In *The World Wide Web Conference*. 2799–2805.

[19] Doris Hoogeveen, Li Wang, Timothy Baldwin, and Karin M Verspoor. 2018. Web forum retrieval and text analytics: A survey. *Foundations and Trends in Information Retrieval* 12, 1 (2018), 1–163.

[20] Saeid Hosseini, Hongzhi Yin, X. Zhou, and S. Sadiq. 2018. Leveraging multi-aspect time-related influence in location recommendation. *World Wide Web* 22 (2018), 1001–1028.

[21] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2391–2401. https://doi.org/10.18653/v1/D19-1243

[22] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. 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*. 1601–1611. https://doi.org/10.18653/v1/P17-1147

[23] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. arXiv:2004.04906 [cs.CL]

[24] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking Beyond the Surface:A Challenge Set for Reading Comprehension over Multiple Sentences. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*.

[25] Souvik Kundu and Hwee Tou Ng. 2018. A Question-Focused Multi-Factor Attention Network for Question Answering. In *AAAI*. AAAI Press, 5828–5835.

[26] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics* (2019). https://tomkwiat.users.x20web.corp.google.com/papers/natural-questions/main-1455-kwiatkowski.pdf

[27] Huayu Li, Yong Ge, Richang Hong, and Hengshu Zhu. 2016. Point-of-Interest Recommendations: Learning Potential Check-Ins from Friends. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 975–984. https://doi.org/10.1145/2939672.2939767

[28] Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning Sparse Neural Networks through L_0 Regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. https://openreview.net/forum?id=H1Y8hhg0b

[29] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 1412–1421. http://aclweb.org/anthology/D/D15/D15-1166.pdf

[30] Ryan McDonald, George Brokos, and Ion Androutsopoulos. 2018. Deep Relevance Ranking using Enhanced Document-Query Interactions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 1849–1860. https://aclanthology.info/papers/D18-1211/d18-1211

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., USA, 3111–3119. http://dl.acm.org/citation.cfm?id=2999792.2999959

[32] Bhaskar Mitra and Nick Craswell. 2018. An Introduction to Neural Information Retrieval. *Foundations and Trends® in Information Retrieval* 13, 1 (December 2018), 1–126. https://www.microsoft.com/en-us/research/publication/introduction-neural-information-retrieval/

[33] Bhaskar Mitra and Nick Craswell. 2019. An Updated Duet Model for Passage Re-ranking. *arXiv preprint arXiv:1903.07666* (2019).

[34] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1291–1299.

[35] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *CoCo@NIPS (CEUR Workshop Proceedings, Vol. 1773)*. CEUR-WS.org.

[36] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

[37] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. 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. 784–789. https://aclanthology.info/papers/P18-2124/p18-2124

[38] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. CoQA: A Conversational Question Answering Challenge. *CoRR* abs/1808.07042 (2018). arXiv:1808.07042 http://arxiv.org/abs/1808.07042

[39] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. https://doi.org/10.1561/1500000019

[40] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional Attention Flow for Machine Comprehension. *CoRR* abs/1611.01603 (2016). arXiv:1611.01603 http://arxiv.org/abs/1611.01603

[41] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. NewsQA: A Machine Comprehension Dataset. *CoRR* abs/1611.09830 (2016). arXiv:1611.09830 http://arxiv.org/abs/1611.09830

[42] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2020. Assessing ranking metrics in top-N recommendation. *Inf. Retr. J.* 23, 4 (2020), 411–448. https://doi.org/10.1007/s10791-020-09377-x

[43] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[44] Quan Yuan, Gao Cong, and Aixin Sun. 2014. Graph-Based Point-of-Interest Recommendation with Geographical and Temporal Influences. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (Shanghai, China) *(CIKM '14)*. Association for Computing Machinery, New York, NY, USA, 659–668. https://doi.org/10.1145/2661829.2661983

[45] Yifei Yuan, Jingbo Zhou, and Wai Lam. 2020. Point-of-Interest Oriented Question Answering with Joint Inference of Semantic Matching and Distance Correlation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Suzhou, China, 542–550. https://aclanthology.org/2020.aacl-main.54

[46] Zixuan Yuan, Hao Liu, Yanchi Liu, Denghui Zhang, Fei Yi, N. Zhu, and Hui Xiong. 2020. Spatio-Temporal Dual Graph Attention Network for Query-POI Matching. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020).