Abstract

Most Reading Comprehension methods limit themselves to queries which can be answered using a single sentence, paragraph, or document. Enabling models to combine disjoint pieces of textual evidence would extend the scope of machine comprehension methods, but currently no resources exist to train and test this capability. We propose a novel task to encourage the development of models for text understanding across multiple documents and to investigate the limits of existing methods. In our task, a model learns to seek and combine evidence – effectively performing multi-hop, alias multi-step, inference. We devise a methodology to produce datasets for this task, given a collection of query-answer pairs and thematically linked documents. Two datasets from different domains are induced, and we identify potential pitfalls and devise circumvention strategies. We evaluate two previously proposed competitive models and find that one can integrate information across documents. However, both models struggle to select relevant information; and providing documents guaranteed to be relevant greatly improves their performance. While the models outperform several strong baselines, their best accuracy reaches 54.5% on an annotated test set, compared to human performance at 85.0%, leaving ample room for improvement.

1 Introduction

Devising computer systems capable of answering questions about knowledge described using text has been a longstanding challenge in Natural Language Processing (NLP). Contemporary end-to-end Reading Comprehension (RC) methods can learn to extract the correct answer span within a given text and approach human-level performance (Kadlec et al., 2016; Seo et al., 2017a). However, for existing datasets, relevant information is often concentrated locally within a single sentence, emphasizing the role of locating, matching, and aligning information between query and support text. For example, Weissenborn et al. (2017) observed that a simple binary word-in-query indicator feature boosted the relative accuracy of a baseline model by 27.9%.

We argue that, in order to further the ability of machine comprehension methods to extract knowledge from text, we must move beyond a scenario where relevant information is coherently and explicitly stated within a single document. Methods with this capability would aid Information Extraction (IE) applications, such as discovering drug-drug interaction...
tions (Gurulingappa et al., 2012) by connecting protein interactions reported across different publications. They would also benefit search (Carpineto and Romano, 2012) and Question Answering (QA) applications (Lin and Pantel, 2001) where the required information cannot be found in a single location.

Figure 1 shows an example from WIKIPEDIA, where the goal is to identify the country property of the Hanging Gardens of Mumbai. This cannot be inferred solely from the article about them without additional background knowledge, as the answer is not stated explicitly. However, several of the linked articles mention the correct answer India (and other countries), but cover different topics (e.g. Mumbai, Arabian Sea, etc.). Finding the answer requires multi-hop reasoning: figuring out that the Hanging Gardens are located in Mumbai, and then, from a second document, that Mumbai is a city in India.

We define a novel RC task in which a model should learn to answer queries by combining evidence stated across documents. We introduce a methodology to induce datasets for this task and derive two datasets. The first, WIKIHOP, uses sets of WIKIPEDIA articles where answers to queries about specific properties of an entity cannot be located in the entity’s article. In the second dataset, MEDHOP, the goal is to establish drug-drug interactions based on scientific findings about drugs and proteins and their interactions, found across multiple MEDLINE abstracts. For both datasets we draw upon existing Knowledge Bases (KBs), WIKIDATA and DRUG BANK, as ground truth, utilizing distant supervision (Mintz et al., 2009) to induce the data – similar to Hewlett et al. (2016) and Joshi et al. (2017).

We establish that for 74.1% and 68.0% of the samples, the answer can be inferred from the given documents by a human annotator. Still, constructing multi-document datasets is challenging; we encounter and prescribe remedies for several pitfalls associated with their assembly – for example, spurious co-locations of answers and specific documents.

For both datasets we then establish several strong baselines and evaluate the performance of two previously proposed competitive RC models (Seo et al., 2017a; Weissenborn et al., 2017). We find that one can integrate information across documents, but neither excels at selecting relevant information from a larger documents set, as their accuracy increases significantly when given only documents guaranteed to be relevant. The best model reaches 54.5% on an annotated test set, compared to human performance at 85.0%, indicating ample room for improvement.

In summary, our key contributions are as follows: Firstly, proposing a cross-document multi-step RC task, as well as a general dataset induction strategy. Secondly, assembling two datasets from different domains and identifying dataset construction pitfalls and remedies. Thirdly, establishing multiple baselines, including two recently proposed RC models, as well as analysing model behaviour in detail through ablation studies.

2 Task and Dataset Construction Method

We will now formally define the multi-hop RC task, and a generic methodology to construct multi-hop RC datasets. Later, in Sections 3 and 4 we will demonstrate how this method is applied in practice by creating datasets for two different domains.

Task Formalization A model is given a query $q$, a set of supporting documents $S_q$, and a set of candidate answers $C_q$ – all of which are mentioned in $S_q$. The goal is to identify the correct answer $a^* \in C_q$ by drawing on the support documents $S_q$. Queries could potentially have several true answers when not constrained to rely on a specific set of support documents – e.g., queries about the parent of a certain individual. However, in our setup each sample has only one true answer among $C_q$ and $S_q$. Note that even though we will utilize background information during dataset assembly, such information will not be available to a model: the document set will be provided in random order and without any metadata. While certainly beneficial, this would distract from our goal of fostering end-to-end RC methods that infer facts by combining separate facts stated in text.

Dataset Assembly We assume that there exists a document corpus $D$, together with a KB containing fact triples $(s, r, o)$ – with subject entity $s$, relation $r$, and object entity $o$. For example, one such fact could be (Hanging Gardens of Mumbai, country, India). We start with individual KB facts and transform them into query-answer pairs by leaving the object slot empty, i.e. $q = (s, r, ?)$ and $a^* = o$.

Next, we define a directed bipartite graph, where
vertices on one side correspond to documents in \(D\), and vertices on the other side are entities from the KB – see Figure 2 for an example. A document node \(d\) is connected to an entity \(e\) if \(e\) is mentioned in \(d\), though there may be further constraints when defining the graph connectivity. For a given \((q, a^*)\) pair, the candidates \(C_q\) and support documents \(S_q \subseteq D\) are identified by traversing the bipartite graph using breadth-first search; the documents visited will become the support documents \(S_q\).

As the traversal starting point, we use the node belonging to the subject entity \(s\) of the query \(q\). As traversal end points, we use the set of all entity nodes that are type-consistent answers to \(q\).

2 Note that whenever there is another fact \((s, r, o')\) in the KB, i.e. a fact producing the same \(q\) but with a different \(a^*\), we will not include \(o'\) into the set of end points for this sample. This ensures that precisely one of the end points corresponds to a correct answer to \(q\).

When traversing the graph starting at \(s\), several of the end points will be visited, though generally not all; those visited define the candidate set \(C_q\). If however the correct answer \(a^*\) is not among them we discard the entire \((q, a^*)\) pair. The documents visited to reach the end points will define the support document set \(S_q\). That is, \(S_q\) comprises chains of documents leading not only from the query subject to the correct answer candidate, but also to type-consistent false answer candidates.

With this methodology, relevant textual evidence for \((q, a^*)\) will be spread across documents along the chain connecting \(s\) and \(a^*\) – ensuring that multi-hop reasoning goes beyond resolving co-reference within a single document. Note that including other type-consistent candidates alongside \(a^*\) as end points in the graph traversal – and thus into the support documents – renders the task considerably more challenging (Jia and Liang, 2017). Models could otherwise identify \(a^*\) in the documents by simply relying on type-consistency heuristics. It is worth pointing out that by introducing alternative candidates we counterbalance a type-consistency bias, in contrast to Hermann et al. (2015) and Hill et al. (2016) who instead rely on entity masking.

2 To determine entities which are type-consistent for a query \(q\), we consider all entities which are observed as object in a fact with \(r\) as relation type – including the correct answer.

3 WIKIHOP

WIKIPEDIA contains an abundance of human-curated, multi-domain information and has several structured resources such as infoboxes and WIKIDATA (Vrandečić, 2012) associated with it. WIKIPEDIA has thus been used for a wealth of research to build datasets posing queries about a single sentence (Morales et al., 2016; Levy et al., 2017) or article (Yang et al., 2015; Hewlett et al., 2016; Rajpurkar et al., 2016). However, no attempt has been made to construct a cross-document multi-step RC dataset based on WIKIPEDIA.

A recently proposed RC dataset is WIKIREADING (Hewlett et al., 2016), where WIKIDATA tuples \((\text{item, property, answer})\) are aligned with the WIKIPEDIA articles regarding their item. The tuples define a slot filling task with the goal of predicting the answer, given an article and property. One problem with using WIKIREADING as an extractive RC dataset is that 54.4% of the samples do not state the answer explicitly in the given article (Hewlett et al., 2016). However, we observed that some of the articles accessible by following hyperlinks from the given article often state the answer, alongside other plausible candidates.

3.1 Assembly

We now apply the methodology from Section 2 to create a multi-hop dataset with WIKIPEDIA as the document corpus and WIKIDATA as structured knowledge triples. In this setup, \((\text{item, property, answer})\) WIKIDATA tuples correspond to \((s, r, o)\) triples, and the item and property of each sample...
together form our query $q$—e.g., *(Hanging Gardens of Mumbai, country, ?)*. Similar to Yang et al. (2015) we only use the first paragraph of an article, as relevant information is more often stated in the beginning. Starting with all samples in WikiReading, we first remove samples where the answer is stated explicitly in the Wikipedia article about the item.$^3$

The bipartite graph is structured as follows: (1) for edges from articles to entities: all articles mentioning an entity $e$ are connected to $e$; (2) for edges from entities to articles: each entity $e$ is only connected to the Wikipedia article about the entity. Traversing the graph is then equivalent to iteratively following hyperlinks to new articles about the anchor text entities.

For a given query-answer pair, the item entity is chosen as the starting point for the graph traversal. A traversal will always pass through the article about the item, since this is the only document connected from there. The end point set includes the correct answer alongside other type-consistent candidate expressions, which are determined by considering all facts belonging to WikiReading training samples, selecting those triples with the same property as in $q$ and keeping their answer expressions. As an example, for the Wikidata property country, this would be the set \{France, Russia, ...\}. We executed graph traversal up to a maximum chain length of 3 documents. To not pose unreasonable computational constraints, samples with more than 64 different support documents or 100 candidates are removed, discarding $\approx 1\%$ of the samples.

### 3.2 Mitigating Dataset Biases

Dataset creation is always fraught with the risk of inducing unintended errors and biases (Chen et al., 2016; Schwartz et al., 2017). As Hewlett et al. (2016) only carried out limited analysis of their WikiReading dataset, we present an analysis of the downstream effects we observe on WikiHop.

**Candidate Frequency Imbalance** A first observation is that there is a significant bias in the answer distribution of WikiReading. For example, in the majority of the samples the property country has the United States of America as the answer. A simple majority class baseline would thus prove successful, but would tell us little about multi-hop reasoning. To combat this issue, we subsampled the dataset to ensure that samples of any one particular answer candidate make up no more than 0.1% of the dataset, and omitted articles about the United States.

**Document-Answer Correlations** A problem unique to our multi-document setting is the possibility of spurious correlations between candidates and documents induced by the graph traversal method. In fact, if we were *not* to address this issue, a model designed to exploit these regularities could achieve 74.6% accuracy (detailed in Section 6).

Concretely, we observed that certain documents frequently co-occur with the correct answer, independently of the query. For example, if the article about London is present in $S_q$, the answer is likely to be the United Kingdom, independent of the query type or entity in question.

We designed a statistic to measure this effect and then used it to sub-sample the dataset. The statistic counts how often a candidate $c$ is observed as the correct answer when a certain document is present in $S_q$ across training set samples. More formally, for a given document $d$ and answer candidate $c$, let $\text{cooccurrence}(d, c)$ denote the total count of how often $d$ co-occurs with $c$ in a sample where $c$ is also the correct answer. We use this statistic to filter the dataset, by discarding samples with at least one document-candidate pair $(d, c)$ for which $\text{cooccurrence}(d, c) > 20$.

### 4 MedHop

Following the same general methodology, we next construct a second dataset for the domain of molecular biology—a field that has been undergoing exponential growth in the number of publications (Cohen and Hunter, 2004). The promise of applying NLP methods to cope with this increase has led to research efforts in IE (Hirschman et al., 2005; Kim et al., 2011) and QA for biomedical text (Hersh et al., 2007; Nentidis et al., 2017). There are a plethora of manually curated structured resources (Ashburner et al., 2000; The UniProt Consortium, 2017) which can either serve as ground truth or to induce training data using distant supervision (Craven and Kumlien, 1999; Bobic et al., 2012). Existing RC datasets are
either severely limited in size (Hersh et al., 2007) or cover a very diverse set of query types (Nentidis et al., 2017), complicating the application of neural models that have seen successes for other domains (Wiese et al., 2017).

A task that has received significant attention is detecting Drug-Drug Interactions (DDIs). Existing DDI efforts have focused on explicit mentions of interactions in single sentences (Gurulingappa et al., 2012; Percha et al., 2012; Segura-Bedmar et al., 2013). However, as shown by Peng et al. (2017), cross-sentence relation extraction increases the number of available relations. It is thus likely that cross-document interactions would further improve recall, which is of particular importance considering interactions that are never stated explicitly – but rather need to be inferred from separate pieces of evidence. The promise of multi-hop methods is finding and combining individual observations that can suggest previously unobserved DDIs, aiding the process of making scientific discoveries, yet not directly from experiments, but by inferring them from established public knowledge (Swanson, 1986).

DDIs are caused by Protein-Protein Interaction (PPI) chains, forming biomedical pathways. If we consider PPI chains across documents, we find examples like in Figure 3. Here the first document states that the drug Leuprolide causes GnRH receptor–induced synaptic potentiations, which can be blocked by the protein Progonadoliberin-1. The last document states that another drug, Triptorelin, is a superagonist of the same protein. It is therefore likely to affect the potency of Leuprolide, describing a way in which the two drugs interact. Besides the true interaction there is also a false candidate Urofollitropin for which, although mentioned together with GnRH receptor within one document, there is no textual evidence indicating interactions with Leuprolide.

4.1 Assembly

We construct MedHop using DrugBank (Law et al., 2014) as structured knowledge resource and research paper abstracts from MEDLINE as documents. There is only one relation type for DrugBank facts, interacts_with, that connects pairs of drugs – an example of a MedHop query would thus be (Leuprolide, interacts_with, ?). We start by processing the 2016 MEDLINE release using the preprocessing pipeline employed for the BioNLP 2011 Shared Task (Stenetorp et al., 2011). We restrict the set of entities in the bipartite graph to drugs in DRUGBANK and human proteins in SWISS-PROT (Bairoch et al., 2004). That is, the graph has drugs and proteins on one side, and MEDLINE abstracts on the other.

The edge structure is as follows: (1) There is an edge from a document to all proteins mentioned in it. (2) There is an edge between a document and a drug, if this document also mentions a protein known to be a target for the drug according to DRUGBANK. This edge is bidirectional, i.e. it can be traversed both ways, since there is no canonical document describing each drug – thus one can “hop” to any document mentioning the drug and its target. (3) There is an edge from a protein p to a document mentioning p, but only if the document also mentions another protein p’ which is known to interact with p according to REACTOME (Fabregat et al., 2016). Given our distant supervision assumption, these additionally constraining requirements err on the side of precision.

As a mention, similar to Percha et al. (2012), we consider any exact match of a name variant of a drug or human protein in DRUGBANK or SWISS-PROT. For a given DDI (drug1, interacts_with, drug2), we then select drug1 as the starting point for the graph traversal. As possible end points, we consider any other drug, apart from drug1 and those interacting with drug1 other than drug2. Similar to WIKIHOP, we exclude samples with more than 64 support documents and impose a maximum document length of 300 tokens plus title.

Document Sub-sampling The bipartite graph for MedHop is orders of magnitude more densely connected than for WIKIHOP. This can lead to poten-
tially large support document sets $S_q$, to a degree where it becomes computationally infeasible for a majority of existing RC models. After the traversal has finished, we subsample documents by first adding a set of documents that connects the drug in the query with its answer. We then iteratively add documents to connect alternative candidates until we reach the limit of 64 documents – while ensuring that all candidates have the same number of paths through the bipartite graph.

**Mitigating Candidate Frequency Imbalance**

Some drugs interact with more drugs than others – *Aspirin* for example interacts with 743 other drugs, but *Isotretinoin* with only 34. This leads to similar candidate frequency imbalance issues as with WIKIHOP – but due to its smaller size MEDHOP is difficult to sub-sample. Nevertheless we can successfully combat this issue by masking entity names, detailed in Section 6.2.

## 5 Dataset Analysis

Table 1 shows the dataset sizes. Note that WIKIHOP inherits the train, development, and test set splits from WIKIREADING – i.e., the full dataset creation, filtering, and sub-sampling pipeline is executed on each set individually. Also note that subsampling according to document-answer correlation significantly reduces the size of WIKIHOP from $\approx 528$K training samples to $\approx 44$K. While in terms of samples, both WIKIHOP and MEDHOP are smaller than other large-scale RC datasets, such as *SQuAD* and WIKIREADING, the supervised learning signal available per sample is arguably greater. One could, for example, re-frame the task as binary path classification: given two entities and a document path connecting them, determine whether a given relation holds. For such a case, WIKIHOP and MEDHOP would have more than 1M and 150K paths to be classified, respectively. Instead, in our formulation, this corresponds to each single sample containing the supervised learning signal from an average of 19.5 and 59.8 unique document paths.

Table 2 shows statistics on the number of candidates and documents per sample on the respective training sets. For MEDHOP, the majority of samples have 9 candidates, due to the way documents are selected up until a maximum of 64 documents is reached. Few samples have less than 9 candidates, and samples would have far more false candidates if more than 64 support documents were included. The number of query types in WIKIHOP is 277, whereas in MEDHOP there is only one: `interacts_with`.

### 5.1 Qualitative Analysis

To establish the quality of the data and analyze potential distant supervision errors, we sampled and annotated 100 samples from each development set.

**WIKIHOP** Table 3 lists characteristics along with the proportion of samples that exhibit them. For 45%, the true answer either uniquely follows from multiple texts directly or is suggested as likely. For 26%, more than one candidate is plausibly supported by the documents, including the correct answer. This is often due to hypernymy, where the appropriate level of granularity for the answer is difficult to predict – e.g. *(west suffolk, administrative_entity, ?)* with candidates *suffolk and england*. This is a direct consequence of including type-consistent false answer candidates from WIKIDATA, which can lead to questions with several true answers. For 9% of the cases a single document suffices; these samples contain a document that states enough information about *item and answer* together. For example, the query *(Louis Auguste, father, ?)* has the correct answer *Louis XIV of France*, and *French king Louis XIV is mentioned within the same doc-

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIKIHOP</td>
<td>43,738</td>
<td>5,129</td>
<td>2,451</td>
<td>51,318</td>
</tr>
<tr>
<td>MEDHOP</td>
<td>1,620</td>
<td>342</td>
<td>546</td>
<td>2,508</td>
</tr>
</tbody>
</table>

Table 1: Dataset sizes for our respective datasets.

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>max</th>
<th>avg</th>
<th>median</th>
</tr>
</thead>
<tbody>
<tr>
<td># cand. – WH</td>
<td>2</td>
<td>79</td>
<td>19.8</td>
<td>14</td>
</tr>
<tr>
<td># docs. – WH</td>
<td>3</td>
<td>63</td>
<td>13.7</td>
<td>11</td>
</tr>
<tr>
<td># tok/doc – WH</td>
<td>4</td>
<td>2,046</td>
<td>100.4</td>
<td>91</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>max</th>
<th>avg</th>
<th>median</th>
</tr>
</thead>
<tbody>
<tr>
<td># cand. – MH</td>
<td>2</td>
<td>9</td>
<td>8.9</td>
<td>9</td>
</tr>
<tr>
<td># docs. – MH</td>
<td>5</td>
<td>64</td>
<td>36.4</td>
<td>29</td>
</tr>
<tr>
<td># tok/doc – MH</td>
<td>5</td>
<td>458</td>
<td>253.9</td>
<td>264</td>
</tr>
</tbody>
</table>

Table 2: Candidates and documents per sample and document length statistics. WH: WIKIHOP; MH: MEDHOP.
Unique multi-step answer. 36%
Likely multi-step unique answer. 9%
Multiple plausible answers. 15%
Ambiguity due to hyponymy. 11%
Only single document required. 9%

Answer does not follow. 12%
WIKIDATA/WIKIPEDIA discrepancy. 8%

Table 3: Qualitative analysis of WIKIHOP samples.

While training models on distantly supervised data is useful, one should ideally evaluate methods on a manually validated test set. We thus identified subsets of the respective test sets for which the correct answer set. Similar to our qualitative analysis of MEDHOP, annotators were shown the query-answer pair as a fact and the chain of relevant documents leading to the answer. They were then instructed to answer (1) whether they knew the fact before; (2) whether the fact follows from the texts (with options “fact follows”, “fact is likely”, and “fact does not follow”); and (3) whether a single or several of the documents are required. Each sample was shown to three annotators and a majority vote was used to aggregate the annotations. Annotators were familiar with the fact 4.6% of the time; prior knowledge of the fact is thus not likely to be a confounding effect on the other judgments. Inter-annotator agreement as measured by Fleiss’ kappa is 0.253 in (2), and 0.281 in (3) – indicating a fair overall agreement, according to Landis and Koch (1977). Overall, 9.5% of samples have no clear majority in (2).

Among samples with a majority judgment, 59.8% are cases where the fact “follows”, for 14.2% the fact is judged as “likely”, and as “not follow” for 25.9%. This again provides good justification for the distant supervision strategy.

Among the samples with a majority vote for (2) of either “follows” or “likely”, 55.9% were marked with a majority vote as requiring multiple documents to infer the fact, and 44.1% as requiring only a single document. The latter number is larger than initially expected, given the construction of samples through graph traversal. However, when inspecting cases judged as “single” more closely, we observed that many indeed provide a clear hint about the correct answer within one document, but without stating it explicitly. For example, for the fact (Witold Cichy, country_of_citizenship, Poland) with documents $d_1$: Witold Cichy (born March 15, 1986 in Wodzislaw Laski) is a Polish footballer[...] and $d_2$: Wodzislaw Laski[...] is a town in Silesian Voivodeship, southern Poland[...], the information provided in $d_1$ suffices for a human given the background knowledge that Polish is an attribute related to Poland, removing the need for $d_2$ to infer the answer.

5.3 Validated Test Sets

While training models on distantly supervised data is useful, one should ideally evaluate methods on a manually validated test set. We thus identified subsets of the respective test sets for which the correct
answer can be inferred from the text. This is in contrast to prior work such as Hermann et al. (2015), Hill et al. (2016), and Hewlett et al. (2016), who evaluate only on distantly supervised samples. For WIKIHop, we applied the same annotation strategy as described in Section 5.2. The validated test set consists of those samples labeled by a majority of annotators (at least 2 of 3) as “follows”, and requiring “multiple” documents. While desirable, crowdsourcing is not feasible for MedHop since it requires specialist knowledge. In addition, the number of document paths is ≈3x larger, which along with the complexity of the documents greatly increases the annotation time. We thus manually annotated 20% of the MedHop test set and identified the samples for which the text implies the correct answer and where multiple documents are required.

6 Experiments

This section describes experiments on WIKIHop and MedHop with the goal of establishing the performance of several baseline models, including recent neural RC models. We empirically demonstrate the importance of mitigating dataset biases, probe whether multi-step behavior is beneficial for solving the task, and investigate if RC models can learn to perform lexical abstraction. Training will be conducted on the respective training sets, and evaluation on both the full test set and validated portion (Section 5.3) allowing for a comparison between the two.

6.1 Models

Random Selects a random candidate; note that the number of candidates differs between samples.

Max-mention Predicts the most frequently mentioned candidate in the support documents $S_q$ of a sample – randomly breaking ties.

Majority-candidate-per-query-type Predicts the candidate $c \in C_q$ that was most frequently observed as the true answer in the training set, given the query type of $q$. For WIKIHop, the query type is the property $p$ of the query; for MedHop there is only the single query type – interacts with.

TF-IDF Retrieval-based models are known to be strong QA baselines if candidate answers are provided (Clark et al., 2016; Welbl et al., 2017). They search for individual documents based on keywords in the question, but typically do not combine information across documents. The purpose of this baseline is to see if it is possible to identify the correct answer from a single document alone through lexical correlations. The model forms its prediction as follows: For each candidate $c$, the concatenation of the query $q$ with $c$ is fed as an OR query into the whoosh text retrieval engine. It then predicts the candidate with the highest TF-IDF similarity score:

$$\arg\max_{c \in C_q} \max_{s \in S_q} (TF-IDF(q + c, s))$$

(1)

Document-cue During dataset construction we observed that certain document-answer pairs appear more frequently than others, to the effect that the correct candidate is often indicated solely by the presence of certain documents in $S_q$. This baseline captures how easy it is for a model to exploit these informative document-answer co-occurrences. It predicts the candidate with the highest score across $C_q$:

$$\arg\max_{c \in C_q} \max_{d \in S_q} (cooccurrence(d, c))$$

(2)

Extractive RC models: FastQA and BiDAF In our experiments we evaluate two recently proposed LSTM-based extractive QA models: the Bidirectional Attention Flow model (BiDAF, Seo et al. (2017a)), and FastQA (Weissenborn et al., 2017), which have shown a robust performance across several datasets. These models predict an answer span within a single document. We adapt them to a multi-document setting by sequentially concatenating all $d \in S_q$ in random order into a superdocument, adding document separator tokens. During training, the first answer mention in the concatenated document serves as the gold span. At test time, we measured accuracy based on the exact match between the prediction and answer, both lowercased, after removing articles, trailing white spaces and punctuation, in the same way as Rajpurkar et al. (2016). To rule out any signal stemming from the order of documents in the superdocument, this order is randomized both at training and test time. In a preliminary experiment we also trained models using different random document order permutations, but found that performance did not change significantly.

4 We also tested assigning the gold span randomly to any one of the mention of the answer, with insignificant changes.
For BiDAF, the default hyperparameters from the implementation of Seo et al. (2017a) are used, with pretrained GloVe (Pennington et al., 2014) embeddings. However, we restrict the maximum document length to 8,192 tokens and hidden size to 20, and train for 5,000 iterations with batchsize 16 in order to fit the model into memory. For FastQA we use the implementation provided by the authors, also with pre-trained GloVe embeddings, no character-embeddings, no maximum support length, hidden size 50, and batch size 64 for 50 epochs.

While BiDAF and FastQA were initially developed and tested on single-hop RC datasets, their usage of bidirectional LSTMs and attention over the full sequence theoretically gives them the capacity to integrate information from different locations in the (super-)document. In addition, BiDAF employs iterative conditioning across multiple layers, potentially making it even better suited to integrate information found across the sequence.

### 6.2 Lexical Abstraction: Candidate Masking

The presence of lexical regularities among answers is a problem in RC dataset assembly — a phenomenon already observed by Hermann et al. (2015). When comprehending a text, the correct answer should become clear from its context — rather than from an intrinsic property of the answer expression. To evaluate the ability of models to rely on context alone, we created masked versions of the datasets: we replace any candidate expression randomly using 100 unique placeholder tokens, e.g. “Mumbai is the most populous city in MASK7.”

Masking is consistent within one sample, but generally different for the same expression across samples. This not only removes answer frequency cues, it also removes statistical correlations between frequent answer strings and support documents. Models consequently cannot base their prediction on intrinsic properties of the answer expression, but have to rely on the context surrounding the mentions.

### 6.3 Results and Discussion

Table 5 shows the experimental outcomes for WIKIHop and MEDHop, together with results for the masked setting: we will first discuss the former. A first observation is that candidate mention frequency does not produce better predictions than a random guess. Predicting the answer most frequently observed at training time achieves strong results: as much as 38.8% / 44.2% and 58.4% / 67.3% on the two datasets, for the full and validated test sets respectively. That is, a simple frequency statistic together with answer type constraints alone is a relatively strong predictor, and the strongest overall for the “unmasked” version of MEDHop.

The TF-IDF retrieval baseline clearly performs better than random for WIKIHop, but is not very strong overall. That is, the question tokens are helpful to detect relevant documents, but exploiting only this information compares poorly to the other baselines. On the other hand, as no co-mention of an interacting drug pair occurs within any single document in MEDHop, the TF-IDF baseline performs worse than random. We conclude that lexical matching with a single support document is not enough to build a strong predictive model for both datasets.

The Document-cue baseline can predict more than a third of the samples correctly, for both datasets, even after sub-sampling frequent document-answer pairs for WIKIHop. The relative strength of this and other baselines proves to be an important issue when designing multi-hop datasets, which we addressed through the measures described in Section 3.2. In Table 4 we compare the two relevant baselines on WIKIHop before and after filtering measures. The absolute strength of these baselines before filtering shows how vital addressing this issue is: 74.6% accuracy could be reached through exploiting the cooccurrence(d,c) statistic alone. This underlines the paramount importance of investigating and addressing dataset biases that otherwise would confound seemingly strong RC model performance. The relative drop demonstrates that

<table>
<thead>
<tr>
<th>Model</th>
<th>Unfiltered</th>
<th>Filtered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document-cue</td>
<td>74.6</td>
<td>36.7</td>
</tr>
<tr>
<td>Maj. candidate</td>
<td>41.2</td>
<td>38.8</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>43.8</td>
<td>25.6</td>
</tr>
</tbody>
</table>

Table 4: Accuracy comparison for simple baseline models on WIKIHop before and after filtering.
the measures undertaken successfully mitigate the issue. A downside to aggressive filtering is a significantly reduced dataset size, rendering it infeasible for smaller datasets like MEDHOP.

Among the two neural models, BiDAF is overall strongest across both datasets – this is in contrast to the reported results for SQuAD where their performance is nearly indistinguishable. This is possibly due to the iterative latent interactions in the BiDAF architecture: we hypothesize that these are of increased importance for our task, where information is distributed across documents. It is worth emphasizing that unlike the other baselines, both FastQA and BiDAF predict the answer by extracting a span from the support documents without relying on the candidate expressions $C_q$.

In the masked setup all baseline models reliant on lexical cues fail in the face of the randomized answer expressions, since the same answer option has different placeholders in different samples. Especially on MEDHOP, where dataset sub-sampling is not a viable option, masking proves to be a valuable alternative, effectively circumventing spurious statistical correlations that RC models can learn to exploit.

Both neural RC models are able to largely retain or even improve their strong performance when answers are masked: they are able to leverage the textual context of the candidate expressions. To understand differences in model behavior between WIKIHOP and MEDHOP, it is worth noting that drug mentions in MEDHOP are normalized to a unique single-word identifier, and performance drops under masking. In contrast, for the open-domain setting of WIKIHOP, a reduction of the answer vocabulary to 100 random single-token mask expressions clearly helps the model in selecting a candidate span, compared to the multi-token candidate expressions in the unmasked setting. Overall, although both neural RC models clearly outperform the other baselines, they still have large room for improvement compared to human performance at 74% / 85% for WIKIHOP.

Comparing results on the full and validated test

### Table 5

<table>
<thead>
<tr>
<th>Model</th>
<th>WIKIHOP standard</th>
<th>WIKIHOP masked</th>
<th>MEDHOP standard</th>
<th>MEDHOP masked</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test</td>
<td>test*</td>
<td>test</td>
<td>test*</td>
</tr>
<tr>
<td>Random</td>
<td>11.5</td>
<td>12.2</td>
<td>12.2</td>
<td>13.0</td>
</tr>
<tr>
<td>Max-mention</td>
<td>10.6</td>
<td>15.9</td>
<td>13.9</td>
<td>20.1</td>
</tr>
<tr>
<td>Majority-candidate-per-query-type</td>
<td>38.8</td>
<td>44.2</td>
<td>12.0</td>
<td>13.7</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>25.6</td>
<td>36.7</td>
<td>14.4</td>
<td>24.2</td>
</tr>
<tr>
<td>Document-cue</td>
<td>36.7</td>
<td>41.7</td>
<td>7.4</td>
<td>20.3</td>
</tr>
<tr>
<td>FastQA</td>
<td>25.7</td>
<td>27.2</td>
<td>35.8</td>
<td>38.0</td>
</tr>
<tr>
<td>BiDAF</td>
<td>42.9</td>
<td>49.7</td>
<td>54.5</td>
<td>59.8</td>
</tr>
</tbody>
</table>

### Table 6

<table>
<thead>
<tr>
<th>Model</th>
<th>WIKIHOP</th>
<th>MEDHOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>standard</td>
<td>gold chain</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>test*</td>
</tr>
<tr>
<td>BiDAF</td>
<td>42.9</td>
<td>49.7</td>
</tr>
<tr>
<td>BiDAF mask</td>
<td>54.5</td>
<td>59.8</td>
</tr>
<tr>
<td>FastQA</td>
<td>25.7</td>
<td>27.2</td>
</tr>
<tr>
<td>FastQA mask</td>
<td>35.8</td>
<td>38.0</td>
</tr>
</tbody>
</table>

Comparing results on the full and validated test...
Table 7: Test accuracy (masked) when only documents containing answer candidates are given (rem).

<table>
<thead>
<tr>
<th></th>
<th>WikiHop</th>
<th>MedHop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test</td>
<td>test*</td>
</tr>
<tr>
<td>BiDAF</td>
<td>54.5</td>
<td>59.8</td>
</tr>
<tr>
<td>BiDAF rem</td>
<td>44.6</td>
<td>57.7</td>
</tr>
<tr>
<td>FastQA</td>
<td>35.8</td>
<td>38.0</td>
</tr>
<tr>
<td>FastQA rem</td>
<td>38.0</td>
<td>41.2</td>
</tr>
</tbody>
</table>

sets, we observe that the results consistently improve on the validated sets. This suggests that the training set contains the signal necessary to make inference on valid samples at test time, and that noisy samples are harder to predict.

6.4 Using only relevant documents

We conducted further experiments to examine the RC models when presented with only the relevant documents in $S_q$, i.e., the chain of documents leading to the correct answer. This allows us to investigate the hypothetical performance of the models if they were able to select and read only relevant documents: Table 6 summarizes these results. Models improve greatly in this gold chain setup, with up to 81.2% / 85.7% on WikiHop in the masked setting for BiDAF. This demonstrates that RC models are capable of identifying the answer when few or no plausible false candidates are mentioned, which is particularly evident for MedHop, where documents tend to discuss only single drug candidates. In the masked gold chain setup, models can then pick up on what the masking template looks like and achieve almost perfect scores. Conversely, these results also show that the models’ answer selection process is not robust to the introduction of unrelated documents with type-consistent candidates. This indicates that learning to intelligently select relevant documents before RC may be among the most promising directions for future model development.

6.5 Removing relevant documents

To investigate if the neural RC models can draw upon information requiring multi-step inference we designed an experiment where we discard all documents that do not contain candidate mentions, including the first documents traversed. Table 7 shows the results: we can observe that performance drops across the board for BiDAF. There is a significant drop of 3.3%/6.2% on MedHop, and 10.0%/2.1% on WikiHop, demonstrating that BiDAF, is able to leverage cross-document information. FastQA shows a slight increase of 2.2%/3.2% for WikiHop and a decrease of 2.7%/4.1% on MedHop. While inconclusive, it is clear that FastQA with fewer latent interactions than BiDAF has problems integrating cross-document information.

7 Related Work

Related Datasets End-to-end text-based QA has witnessed a surge in interest with the advent of large-scale datasets, which have been assembled based on Freebase (Berant et al., 2013; Bordes et al., 2015), Wikipedia (Yang et al., 2015; Rajpurkar et al., 2016; Hewlett et al., 2016), web search queries (Nguyen et al., 2016), news articles (Hermann et al., 2015; Onishi et al., 2016), books (Hill et al., 2016; Paperno et al., 2016), science exams (Welbl et al., 2017), and trivia (Boyd-Graber et al., 2012; Dunn et al., 2017). Besides TriviaQA (Joshi et al., 2017), all these datasets are confined to single documents, and RC typically does not require a combination of multiple independent facts. In contrast, WikiHop and MedHop are specifically designed for cross-document RC and multi-step inference. There exist other multi-hop RC resources, but they are either very limited in size, such as the FraCaS test suite, or based on synthetic language (Weston et al., 2016). TriviaQA partly involves multi-step reasoning, but the complexity largely stems from parsing compositional questions. Our datasets center around compositional inference from comparatively simple queries and the cross-document setup ensures that multi-step inference goes beyond resolving co-reference.

Compositional Knowledge Base Inference

Combining multiple facts is common for structured knowledge resources which formulate facts using first-order logic. KB inference methods include Inductive Logic Programming (Quinlan, 1990; Pazzani et al., 1991; Richards and Mooney, 1991) and probabilistic relaxations to logic like Markov Logic (Richardson and Domingos, 2006; Schoenmackers et al., 2008). These approaches suffer from
limited coverage and inefficient inference, though efforts to circumvent sparsity have been undertaken (Schoenmackers et al., 2008; Schoenmackers et al., 2010). A more scalable approach to composite rule learning is the Path Ranking Algorithm (Lao and Cohen, 2010; Lao et al., 2011), which performs random walks to identify salient paths between entities. Gardner et al. (2013) circumvent these sparsity problems by introducing synthetic links via dense latent embeddings. Several other methods have been proposed, using composition functions such as vector addition (Bordes et al., 2014), RNNs (Neelakantan et al., 2015; Das et al., 2017), and memory networks (Jain, 2016).

All of these previous approaches center around learning how to combine facts from a KB, i.e., in a structured form with pre-defined schema. That is, they work as part of a pipeline, and either rely on the output of a previous IE step (Banko et al., 2007), or on direct human annotation (Bollacker et al., 2008) which tends to be costly and biased in coverage. However, recent neural RC methods (Seo et al., 2017a; Shen et al., 2017) have demonstrated that end-to-end language understanding approaches can infer answers directly from text – sidestepping intermediate query parsing and IE steps. Our work aims to evaluate whether end-to-end multi-step RC models can indeed operate on raw text documents only – while performing the kind of inference most commonly associated with logical inference methods operating on structured knowledge.

**Text-Based Multi-Step Reading Comprehension**

Fried et al. (2015) have demonstrated that exploiting information from other related documents based on lexical semantic similarity is beneficial for re-ranking answers in open-domain non-factoid QA. Jansen et al. (2017) chain textual background resources for science exam QA and provide multi-sentence answer explanations. Beyond, a rich collection of neural models tailored towards multi-step RC has been developed. Memory networks (Weston et al., 2015; Sukhbaatar et al., 2015; Kumar et al., 2016) define a model class that iteratively attends over textual memory items, and they show promising performance on synthetic tasks requiring multi-step reasoning (Weston et al., 2016). One common characteristic of neural multi-hop models is their rich structure that enables matching and interaction between question, context, answer candidates and combinations thereof (Peng et al., 2015; Weissenborn, 2016; Xiong et al., 2017; Liu and Perez, 2017), which is often iterated over several times (Sordoni et al., 2016; Neumann et al., 2016; Seo et al., 2017b; Hu et al., 2017) and may contain trainable stopping mechanisms (Graves, 2016; Shen et al., 2017). All these methods show promise for single-document RC, and by design should be capable of integrating multiple facts across documents. However, thus far they have not been evaluated for a cross-document multi-step RC task – as in this work.

**Learning Search Expansion** Other research addresses expanding the document set available to a QA system, either in the form of web navigation (Nogueira and Cho, 2016), or via query reformulation techniques, which often use neural reinforcement learning (Narasimhan et al., 2016; Nogueira and Cho, 2017; Buck et al., 2018). While related, this work ultimately aims at reformulating queries to better acquire evidence documents, and not at answering queries through combining facts.

**8 Conclusions and Future Work**

We have introduced a new cross-document multi-hop RC task, devised a generic dataset derivation strategy and applied it to two separate domains. The resulting datasets test RC methods in their ability to perform composite reasoning – something thus far limited to models operating on structured knowledge resources. In our experiments we found that contemporary RC models can leverage cross-document information, but a sizeable gap to human performance remains. Finally, we identified the selection of relevant document sets as the most promising direction for future research.

Thus far, our datasets center around factoid questions about entities, and as extractive RC datasets, it is assumed that the answer is mentioned verbatim. While this limits the types of questions one can ask, these assumptions can facilitate both training and evaluation, and future work – once free-form abstractive answer composition has advanced – should move beyond. We hope that our work will foster research on cross-document information integration, working towards these long term goals.
Acknowledgments

We would like to thank the reviewers and the action editor for their thoughtful and constructive suggestions, as well as Matko Bošnjak, Tim Dettmers, Pasquale Minervini, Jeff Mitchell, and Sebastian Ruder for several helpful comments and feedback on drafts of this paper. This work was supported by an Allen Distinguished Investigator Award, a Marie Curie Career Integration Award, the EU H2020 SUMMA project (grant agreement number 688139), and an Engineering and Physical Sciences Research Council scholarship.

References


