Sequential Question Answering Using Neural Coreference Resolution

Summer Wu  
Department of Computer Science  
Yale University  
summer.wu@yale.edu

Dragomir Radev  
Department of Computer Science  
Yale University  
dragomir.radev@yale.edu

Abstract

The best question answering systems currently use semantic parsers to map natural language queries to intermediate logical forms to answers. While this may work well for standalone questions, it fails to handle more complex sets of sequential questions that include references to previously asked questions. Natural language is replete with coreference; for a system to intelligently understand and respond to questions in a conversation, that system must account for the context that was previously established. Here we present a method for applying neural network-based coreference resolution techniques to the task of improving sequential question answering. This work builds off of recent research in end-to-end neural coreference resolution, semantic parsing on semi-structured tables, and the first sequential question answering (SQA) dataset. Initial experiments demonstrate accuracy levels similar to those achieved without applying coreference resolution, suggesting that either the method used for applying coreference resolution requires fine tuning, or that there may be a lack of dataset fit.

1 Introduction

In 2016, the first ever dataset of sequential questions and answers was compiled to explore a conversational question answering (QA) setting (Iyyer et al., 2016). After determining that semantic parsers are currently more effective at QA than neural parsers, Iyyer found that while the parser was able to answer the first question of the 6,066 sequences with an accuracy of 48.7%, it answered subsequent questions at significantly lower accuracy rates, bringing the overall accuracy on the dataset to a mere 32.8%. Logically, this makes sense, as the first question in each sequence was written with sufficient context to answer it, but the remaining questions require knowledge from previous questions, which existing semantic parsers do not have access to.

A practical application of an improved sequential question answering system is Professor Dragomir Radev’s Sapphire project, which seeks to develop dialogue systems for student advising. Given a question from a student regarding a class, Sapphire advises a student on a course of action. For example, a student might ask a question along the lines of, “Which math classes am I allowed to take next semester?” This is simple enough; natural language processing (NLP) research on converting Seq2SQL (Zhong et al., 2017) has discovered a method for translating natural language sentences into SQL statements. The problem becomes far more complex when the student asks a follow-up question that references elements of a previous question. Coreference is when two or more expressions in a text refer to the same person or thing. While it is easy for humans to make the connection between coreferenced elements, it is difficult for computers to backtrack and determine the subject of coreference. For instance, a follow-up question to the example of math class eligibility might be, “Of those classes, which has the highest rating?” To answer the question, a computer needs to backtrack and determine what “those classes” is referring to. Note that the coreference does not need to be a noun phrase; it could be a verb phrase, set of adjectives, or anything else referring to something that was previously mentioned. Thus, the challenge here is two-fold: first, to find the proper coreference in a set of questions and answers, and second, to use that information to improve sequential question answering.
<table>
<thead>
<tr>
<th>Name</th>
<th>Japanese</th>
<th>Distance (km)</th>
<th>Connections</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Okayama</td>
<td>Gang Shan</td>
<td>0</td>
<td>Sanyo Shinkansen, Sanyo Main Line, Ako Line, Hakubi Line, Uno Line (Seto-Ohashi Line), Tsuyama Line</td>
<td>Kita-ku, Okayama</td>
</tr>
<tr>
<td>Bizen-Mikado</td>
<td>Bei Qian</td>
<td>1.9</td>
<td>Okayama Electric Tramway: Higashiyama Line, Seikibashi Line</td>
<td>Kita-ku, Okayama</td>
</tr>
<tr>
<td>Daianji</td>
<td>Da An Si</td>
<td>3.3</td>
<td></td>
<td>Kita-ku, Okayama</td>
</tr>
<tr>
<td>Bizen-Ichinomiya</td>
<td>Bei Qian Yi</td>
<td>6.5</td>
<td></td>
<td>Kita-ku, Okayama</td>
</tr>
<tr>
<td>Kibitsu</td>
<td>Ji Bei Jin</td>
<td>8.4</td>
<td></td>
<td>Kita-ku, Okayama</td>
</tr>
<tr>
<td>Bitchu-Takamatsu</td>
<td>Bei Zhong</td>
<td>11</td>
<td></td>
<td>Kita-ku, Okayama</td>
</tr>
<tr>
<td>Ashimori</td>
<td>Zu Shou</td>
<td>13.4</td>
<td></td>
<td>Kita-ku, Okayama</td>
</tr>
<tr>
<td>Hattori</td>
<td>Fu Bu</td>
<td>16.2</td>
<td></td>
<td>Okayama</td>
</tr>
<tr>
<td>Higashi-Soja</td>
<td>Dong Zong</td>
<td>18.8</td>
<td></td>
<td>Soja</td>
</tr>
<tr>
<td>Soja</td>
<td>She</td>
<td>20.4</td>
<td>Hakubi Line Ibara Railway Ibara Line</td>
<td>Soja</td>
</tr>
</tbody>
</table>

Table 1: One of the Wikipedia tables off of which SQA is based. This table contains information about the stations along the Kibi Line, a railway line in Japan. Our task is to develop a system that can use a semi-structured table like this one to answer questions.

Figure 1: In this first step of the end-to-end coreference model, words are passed through a bidirectional LSTM and grouped into spans. Mention scores are assigned to these spans, and entities with low scores are pruned. The goal is to keep the most salient, and thus the most likely to be coreferenced, spans.
Linguistic techniques for resolving coreference include using syntactic, semantic, discourse, or morphological information to rank possible antecedents (Deoskar, 2004; Sayed, 2003). In 2017, neural networks were used to build the first end-to-end coreference resolution model not requiring any external resources (beyond the CoNLL Shared Task dataset). The model easily outperformed existing co-reference resolution benchmarks (Lee et al., 2017).

Almost two decades have passed since research has been done in applying coreference resolution techniques to improve question answering. In 1999, Thomas Morton from the University of Pennsylvania found that by replacing nouns with their previously referenced name (as determined by a set of grammatical rules), QA accuracy improved 14% (Morton, 1999). I plan to use the same technique to make use of the coreference information—simply replacing the current element(s) with its previously determined value, as found via neural coreference resolution, then investigating whether there is a difference in SQA accuracy.

2 Materials and Methods

In this work, I combine Lee et al’s coreference model with Pasupat and Liang’s floating parser (Pasupat and Liang, 2015) and apply it to Iyyer’s Sequential Question Answering (SQA) dataset.

2.1 SQA

SQA is an expansion of WikiTableQuestions, a series of natural language question-answer pairs written about a set of semi-structured data tables from Wikipedia. SQA is the first such dataset ever compiled, and includes semi-structured tables from Wikipedia for the QA system to use to lookup the answers. The dataset contains 6,066 unique sequences of questions, and 17,553 questions total. As previously mentioned, preliminary tests on SQA did not perform very well, averaging less than 33% overall accuracy, and leave much room for improvement.

2.2 Coreference

Lee et al’s application of neural networks on the 2012 CoNLL shared task dataset to locate coreference in multilingual texts produces the highest percentage of coreference annotation accuracy to date.

Figure 2: In the second step of the end-to-end coreference neural network, the antecedent and mention scores are summed to determine the coreference score. If the coreference score between spans is sufficiently high, the spans are predicted to be in a cluster. The goal of training the network is to maximize the probability that the predicted spans are indeed in a cluster.

It does so by learning a probability distribution whose most likely configuration produces the correct clustering. The first step of the model is to compute embedding representations of spans in order to assign each span a mention score. While initially, every span up to a maximum width is considered a potential antecedent, low-scoring spans are soon pruned so that the number of spans being considered is manageable.

Finally, the coreference score (the likelihood of a span being a coreference of another) is the sum of the antecedent (relative importance to another span) and the mention (general importance in the context of the sentence) scores.

2.3 Floating Parser

As Iyyer found that the floating parser (FP) had a significantly higher accuracy rate (32.8%) on SQA than an end-to-end neural network (17.4%), I decided to use FP as the base for my project’s SQA system. FP is trained from question-answer pairs as follows: first, the semi-structured HTML table in which the answer is located is encoded in a knowledge graph. Then, the system parses the question into candidate logical forms, ranks the candidates with a log-linear model, and finally executes the highest-ranked logical form to get the answer. Beam search and pruning strategies are used to limit the number of candidates being considered at any point.
Figure 3: Visualization of the predicted coreference clusters output by the neural network, with some of their corresponding locations bolded in the document. The clusters in this example are partially correct- "these stations" in the second sentence is indeed clustered with "all the stations on the kibi line." However, the following "these"’s refer to distances, not stations, which our model was unable to detect.

Figure 4: How does floating parser work? First, the dataset is converted into a knowledge graph, \( w \). \( x \) is placed into the knowledge base and parsed into a set of logical forms \( Z_x \). The highest ranked form \( z \) is executed and returns the answer \( y \).

2.4 A Combined Model

Word embeddings were trained using an end-to-end neural network on the OntoNotes CoNLL 2012 Shared Task dataset, which consists of coreference data from genres including newswire, broadcast news, broadcast conversation, magazines, telephone conversations, web data, and pivot text. The embeddings were then used to configure a coreference resolution model that predicts clusters of mentions.

Sequential questions from the SQA dataset were combined into a single block of text before being input into the coreference model, returning a set of predicted clusters. The first (with the required caveat of being the most specific) reference was then used to replace all the later references.

The modified SQA dataset was converted to LispTree in preparation for applying FP. Using FP, experiments were conducted on various splits of the data using different parameters such as choice of dataset and beam size. Due to time constraints, the number of experiments I was able to run was limited, and given more time, I would absolutely run more.

3 Results

I ran experiments across three different combinations of splits and attributes, the results of which are summarized in Table 2.

The “Dev Set” or development set contains 12,276 questions in its training split and 2,265 questions in its test split. Each run took approximately four hours.

The “Test Set” (or the full data set) contains 14,541 questions in its training split and 3,012 questions in its test split. Each run took approximately five hours.

In each case, the model accuracy on the modified training set was slightly higher than the model accuracy on the original training set, yet the model accuracy on the modified test set was slightly lower than that on the original. As a representative example, while the Test Set’s accuracy on the training set improved slightly from 59.8% without coreference resolution to 60.1% with coreference resolution, its accuracy on questions not in the training set decreased slightly from 32.8% to 32.7%. As improving the accuracy on the test set is the primary goal, this could be a sign of overfitting to the training set. Then again, since the dif-
what are all the stations on the kibi line?

what are the distances of these stations from the start of the line?

do of these, which is larger than 1 km?

do of these, which is smaller than 2 km?

which station is this distance from the start of the line?

what are all the stations on the kibi line?

what are the distances of all the stations on the kibi line from the start of the kibi line?

all the stations on the kibi line, which is larger than 1 km?

all the stations on the kibi line, which is smaller than 2 km?

which station is this distance from the start of the kibi line?

Table 2: Questions from the SQA dataset before and after using the predicted clusters to preprocess input data and replace coreferences. As mentioned in the caption under Figure 3, inaccuracies in clustering result in less accurate questions.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev Set (n = 100)</td>
<td>0.659</td>
<td>0.165</td>
</tr>
<tr>
<td>Dev Set M (n = 100)</td>
<td>0.668</td>
<td>0.146</td>
</tr>
<tr>
<td>Dev Set (beam size = 50)</td>
<td>0.610</td>
<td>0.325</td>
</tr>
<tr>
<td>Dev Set M (beam size = 50)</td>
<td>0.614</td>
<td>0.320</td>
</tr>
<tr>
<td>Dev Set</td>
<td>0.614</td>
<td>0.320</td>
</tr>
<tr>
<td>Dev Set M</td>
<td>0.622</td>
<td>0.316</td>
</tr>
<tr>
<td>Test Set</td>
<td>0.598</td>
<td>0.328</td>
</tr>
<tr>
<td>Test Set M</td>
<td>0.601</td>
<td>0.327</td>
</tr>
</tbody>
</table>

Table 3: A selection of experiments, M means run on a dataset that was “modified” with coreference preprocessing. “Dev Set” is the development set and the first two experiments are run on just 100 examples. The next two are run on all the examples in the development set, but using a reduced beam size of 50. The penultimate pair runs on the development set with a default beam size of 200, and the final pair runs on the test set. Both of these experiments took hours to run.

Table 3: A selection of experiments, M means run on a dataset that was “modified” with coreference preprocessing. “Dev Set” is the development set and the first two experiments are run on just 100 examples. The next two are run on all the examples in the development set, but using a reduced beam size of 50. The penultimate pair runs on the development set with a default beam size of 200, and the final pair runs on the test set. Both of these experiments took hours to run.

The results do not look very promising. However, that does not mean coreference is ineffectual on answering sequential questions, especially since coreference resolution aligns with our semantic constructions of questions and FP. Logically, a system must be able to resolve coreference in order to answer a series of questions. Thus, the results suggest that further experimentation is required, and perhaps even that this dataset is not the best fit for the problem.

Moreover, given the tabular content of this dataset, replacing phrases with their coreferences may not actually be that helpful. For instance, in the excerpted questions found in Table 2, coreference resolution replaces ambiguous references to the Kibi line and the stations along it with explicit ones. In a general context, it is necessary for a system to have this information to answer the question, however in the context of the SQA dataset where the specific table containing the answer is given (in this case stations along the Kibi line; see Table 1), the additional information does not help narrow down the choices for the parser.

4 Discussion

The results do not look very promising. However, that does not mean coreference is ineffectual on answering sequential questions, especially since coreference resolution aligns with our semantic constructions of questions and FP. Logically, a system must be able to resolve coreference in order to answer a series of questions. Thus, the results suggest that further experimentation is required, and perhaps even that this dataset is not the best fit for the problem.

Since each experiment took hours to run, the number of experiments I was able to run between when the combined model was complete and the project’s due date was limited. Given more time, the next area I would explore would be normalizing how the system decides whether an answer is right or wrong. Currently, an answer is considered correct only when the answer matches exactly; it seems to me that if there are two lists with 90% overlap, partial correctness should be awarded.

Moreover, given the tabular content of this dataset, replacing phrases with their coreferences may not actually be that helpful. For instance, in the excerpted questions found in Table 2, coreference resolution replaces ambiguous references to the Kibi line and the stations along it with explicit ones. In a general context, it is necessary for a system to have this information to answer the question, however in the context of the SQA dataset where the specific table containing the answer is given (in this case stations along the Kibi line; see Table 1), the additional information does not help narrow down the choices for the parser.

5 Conclusion

A system was created to apply neural coreference resolution to the task of improving sequential question answering. While coreference resolution on SQA caused the accuracy on the training set to
improve slightly, it actually resulted in a slight decrease in accuracy on the test set. I believe this result is more indicative of a lack of dataset fit than it is of the potential for coreference resolution to improve question answering. Used in the proper context, this method described in this paper paves the way for future work in conversational QA systems.

Personally, I found this experience to be a fantastic learning opportunity, as well as a wonderful introduction to computer science, and specifically NLP, research. Digging through cutting edge research papers, reproducing some of the top computer scientists’ work, and eventually adding my own touch, was challenging. It took me months of rereading, searching online, and even emailing one of the authors to reproduce all of their experiments and confirm their results myself. After I understood their code, I had to be creative to figure out how to combine three separate models into a single system. Thus, seeing each element of the final system plug into the next was incredibly rewarding. This problem of improving SQA fascinates me, and I hope to continue working on it at least a little longer.

Acknowledgments

Thank you to Professor Dragomir Radev for advising me, Rui Zhang for coreference related pointers, and Dr. Mohit Iyyer for sending me scripts that helped me replicate his experiments. Additionally, thank you to Kenton Lee and the researchers he worked with, as well as Panupong Pasupat and the researchers he worked with, for making their respective experiments extensible and open source.

References


Tom Morton. 1999. Using coreference to improve passage retrieval for question answering. AAAI.

