Summarizing and Exploring Tabular Data in Conversational Search

Shuo Zhang*
Bloomberg
London, United Kingdom
szhang611@bloomberg.net

Zhuyun Dai*
Carnegie Mellon University
Pittsburgh, USA
zhuyund@cs.cmu.edu

Krisztian Balog
University of Stavanger
Stavanger, Norway
krisztian.balog@uis.no

Jamie Callan
Carnegie Mellon University
Pittsburgh, USA
callan@cs.cmu.edu

arXiv:2005.11490v1 [cs.IR] 23 May 2020

ABSTRACT

Tabular data provide answers to a significant portion of search queries. However, reciting an entire result table is impractical in conversational search systems. We propose to generate natural language summaries as answers to describe the complex information contained in a table. Through crowdsourcing experiments, we build a new conversation-oriented, open-domain table summarization dataset. It includes annotated table summaries, which not only answer questions but also help people explore other information in the table. We utilize this dataset to develop automatic table summarization systems as SOTA baselines. Based on the experimental results, we identify challenges and point out future research directions that this resource will support.

KEYWORDS

Table summarization; Conversational systems; Table understanding; Table navigation

ACM Reference Format:
Shuo Zhang, Zhuyun Dai*, Krisztian Balog, and Jamie Callan. 2020. Summarizing and Exploring Tabular Data in Conversational Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401205

1 INTRODUCTION

Many search queries are seeking a set of items and their attributes, e.g., "highest-grossing movies" or "best places to travel in Christmas." In a typical information retrieval system, such search results can be presented as a list or table. In browser-based search, approximately 10% of QA queries are answered by tabular data [14]. While structured data is available in large quantities, presenting search results in a tabular format is challenging in conversational search systems (e.g., Siri, Alexa, and Google Assistant). First, tables are rich in structure while low in natural language content. Reading out a table line by line will lead to a poor user experience. Second, tables can be quite large. They cannot be presented effectively in chat or voice-only user interfaces.

*The first two authors contributed equally.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '20, July 25–30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401205

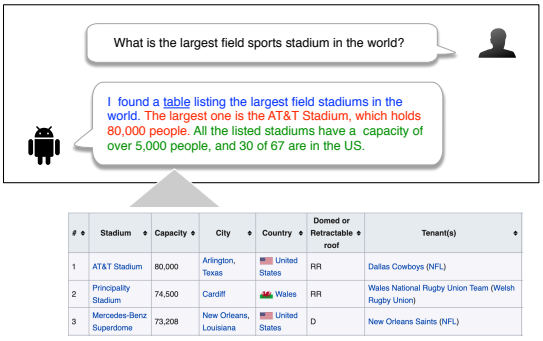

Figure 1: Table summarization in conversational search. The summary of the result table includes a leading sentence describing the table (red), answer to the question (blue), and a sentence helping to explore the table further (green).

In this work, we aim to bridge this gap by facilitating natural language summarization of tables in conversational search. Specifically, we wish to leverage the interactive nature of multi-turn conversational systems. Instead of telling users everything at once, we seek to present a concise *summary* as well as give users information that helps to further *explore* the table's content. That is, we want the system to *drive the conversation*—provide clues that aid users in what they could ask next. We illustrate the idea in Fig 1.

There exist structure-based as well as content-based methods for table summarization. The former can rely on a single predefined schema [6], automatically generated attribute value taxonomies [5], or patterns succinctly summarizing the tables [1]. The latter can utilize value lattices [11] or table rows [7] by presenting partial content. However, these methods assume a decomposable table structure and are often limited to predefined schemes, and are not conditioned on the conversation context that triggered the table.

It is an open research question what types of summaries can help people to explore the table in conversations. We propose a table summary with the following traits. First, the summary should contain the *answer to the user's question* asked in the last conversation turn. For example, "the world's largest field stadium is the AT&T stadium, and it can hold 80,000 people." Second, the summary should let the user know what is inside the table, so that they can further explore its contents. Thus, the summary may provide an *overview of the table*, e.g., "the table lists the largest field sports stadiums in the world," or *additional information* from the table, e.g, "30 of 67 are from the US." In this way, the user can keep exploring the table by asking questions like "What is the second largest one?"

As a first step towards the development of automatic approaches, we create a test collection for this task. Specifically, we use the above traits to define crowdsourcing tasks, and collect manually-written

table summaries along with corresponding relevance assessments. We also compares existing state-of-the-art natural language generation models on our dataset to gain further insights.

In summary, the main contributions of this work are: (1) introduction of the task of table summarization in conversational search; (2) creation of a test collection using crowdsourcing, which is made publicly available;[1] (3) investigation of how general-purpose abstractive summarization methods perform as baselines; (4) analysis of the baseline results and identification of future directions.

2 CREATING A TEST COLLECTION

Our objective is to create a test collection to study table summerization in a conversational setting. Specifically, given a user query q in a conversation and a result table T, we aim to create a summary S of this table that can help answer the query. We describe the sampling of queries and tables in Sect. 2.1 and detail the crowdsourcing experiments we carried out to collect summaries and annotations in Sects. 2.2 and 2.3, respectively.

2.1 Queries and Tables

Our test collection comprises of 200 tables, each with a corresponding question as the conversation context. That is, $\langle q, T \rangle$ pairs constitute the input to summary generation.

The tables are randomly selected from the WikiTableQuestions dataset [8], which has formerly been used for semantic parsing on tables. This dataset contains 2108 Wikipedia tables on a large variety of topics, and over 22k questions for querying these tables. While this dataset can be reused for our task, it does not provide table summaries. We require each selected table to have at least fix rows and four columns, because tables smaller than this may be displayed in their entirety, without needing summarization. We manually identify the type of each table. There are 64 Sport, 33 Place, 27 Music, 16 Film, 12 Culture, 11 Traffic, 8 Product, and 30 Other tables (including Politics, TV series, Award, and Company).

The questions are either sampled from the WikiTableQuestions dataset (45 questions), or written by two of the authors (155 questions). The former are mostly fact-checking questions, e.g., "How many points did Toronto have more than Montreal in their first game?" The latter contains both fact-checking questions and more open-ended ones, e.g., "tell me about the car models made by Ford."

2.2 Collecting Candidate Summaries

We aim to collect summaries that are suitable in a conversational setting. To achieve this goal, we designed a crowd-sourcing task that mimics a real conversation with a friend. The instructions were given such that the desired traits for table summaries are emphasized: *relevant to the question* and provides *clues for exploration.*

Specifically, the instructions given for the crowdsourcing task are as follows: "Image talking to a friend on the phone. Your friend asks you a question, and you find the following table on the Web. Remember that your friend cannot see the table. Your goal is to let your friend to capture essential information in the table related to the question. Your summary should be short but comprehensive. Try to describe several rows or columns that you find interesting."

As shown in Fig 2, we show the question, the title of the table, and its Wikipedia page link to the crowd worker. The worker needs

[1] https://github.com/iai-group/sigir2020-tablesum

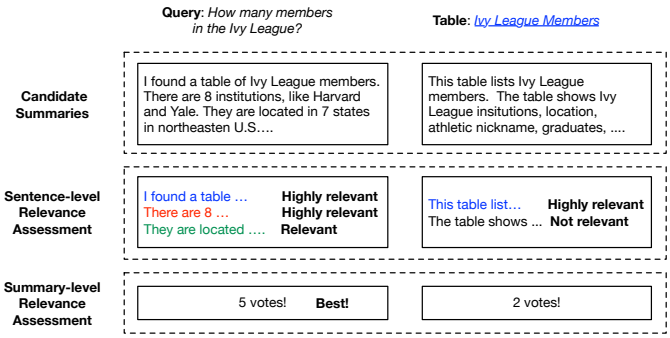

Figure 2: Illustration of our crowdsourcing workflow.

to click on the link, read the table, and write the summary in a text input box that has a character counter. We limit the summary to be around in 30-50 words (150-250 characters).[2] For any $\langle q, T \rangle$ pair, we invited five workers and suggested them spend at least 5 minutes in writing the summary. The average time per assignment reported from Amazon Mechanical Turk was 19 minutes 36 seconds. In the end, we collected $200 \times 5 = 1000$ candidate summaries.

2.3 Quality Assessment

Given the set S of five summaries collected in Sect. 2.2, we aim to find the best one by labeling them with relevance through a second set of crowdsourcing experiments. Specifically, ground truth relevance labels are obtained for each sentence (Sect. 2.3.1) as well as for the entire summary (Sect. 2.3.2).

2.3.1 Sentence Level. We collect sentence-level relevance assessments by employing three judges (see the middle block in Fig. 2 for the illustration of this step). Each summary was split into sentences. In total there are 3459 sentences, i.e., 3.5 sentences per summary on average. We show each sentence to the judges, with its previous and next sentences provided as context. Sentences were judged on a three-point scale: highly relevant, relevant, and non-relevant. The annotators were situated in a scenario where they need to write a short summary. This summary should help their friend to capture the information contained in this table easily, and should provide a fact as an answer to the question. Specifically, they were given the following guidelines: (i) a sentence is **highly relevant** if it tells the table topic or provides a fact regarding the friend's question; (ii) a sentence is **relevant** if it is not directly relevant to the friend's question, but helps to figure out what the table contains for those who can not see it; and (iii) a sentence is **not relevant** if it is not about the table or is unrelated to the question.

2.3.2 Summary Level. Summary-level assessments were collected by showing the judges all candidate summaries for a table, and asking them to select the *best* one (see the bottom block in Fig. 2 for the illustration of this step). We define quality guidelines based on the following three aspects: (i) language quality: the summary should be concise and easy to read; (ii) relevance: the summary should provide a fact regarding the friend's question; and (iii) able to drive conversations: the summary should be able to attract the

[2] This is based on that (i) the short text message (SMS) limit is 160 chars, and we want the summary to fit approximately within that limit, and (ii) in spoken English, people speak 125 words per minute, and we want the summary to be 15-30 seconds long.

reader's interests, and intrigue the reader to ask follow-up questions. For example, the summary may provide an overview of the table, or show some highlights from the table that may not directly relate to the question.

For each table, the five collected summaries (cf. Sect. 2.2), together with the question, were shown to 7 judges.[3] The summaries were organized in a list; the order of summaries was randomly shuffled each time to eliminate potential rank bias. Neither the table itself nor the Wikipedia URL were given to the judge, to mimic the real use case where the user only sees/hears the summary but cannot see the table. The judges were asked to pick the highest quality summary from the five based on the above guidelines. The one with the most votes becomes the ground truth summary for the table.

Among the 200 top-voted summaries, the numbers of summaries that obtain 5, 4, 3, and 2 votes are 5, 34, 115, and 46, respectively. In other words, for 154/200 (77%) of the tables, at least 3 judges agreed on which summary was the best. By analyzing the top-voted summaries, we found that the judges tend to prefer summaries that are longer and use a diverse vocabulary (unique words). On average, the top-voted summaries have 53.1 words and 41.2 unique words; for comparison, an average summary has 46.2 words and 36.9 unique words, while the least-voted summary has 40.8 words and 33.3 unique words. Additionally, we found that the top-voted summary uses more numbers. This, to a certain extent, shows that these summaries contain more factual information. These findings are aligned with our guidelines, which require the summary to provide comprehensive information about the table to intrigue follow-up dialogue.

3 EXPERIMENTS

In this section, we compare the performance of several state-of-the-art natural language generation models [3, 10, 12] on our dataset. Our aim is to understand the extent to which table summaries can be generated automatically using current state-of-the-art approaches, and to gain insights into what the challenges of this task are.

3.1 Methods for Comparison

This section first introduces how we represent tables for the neural language generation models. Next, it introduces three stage-of-the-art models for comparison.

Representating Tables as Text Sequence. Most of the current state-of-the-art natural language generation models expect the input to be a 1-dimensional text sequence. Inspired by Hancock et al. [4], we flatten tables into a sequence of words, and use a "key:value" format to preserve the structure. Specifically, we take one row at a time, and pair the cells with the corresponding column header. For example, the cell value "164,119" in the column "Sales" generates "Sales:164,119." Next, we concatenate the header-cell pairs in the same row using commas, and concatenate the rows to represent the entire table. Row indexes are listed at the beginning of the corresponding rows. Finally, we add the page title and table caption to provide context. We also add the total number of rows to the flattened table, because many manual summaries start with "there are in total X rows." The page title, table caption, and the total number of rows are also represented in "key: value" format, where the key is "PageTitle," "Caption," or "TotalRows," and the value is the corresponding text. During training, the models are expected to understand the meanings of these keys and learn to use their values to generate summaries.

We consider three neural natural language generation models to generate summaries based on the flattened tables. We take these methods as state-of-the-arts baselines that others can reproduce and use to test their own solutions.

CopyNet [3] is an LSTM-based encoder-decoder model that incorporates the copying mechanism. During training, the model encodes the input table using a layer of bidirectional LSTM and tries to decode it into the human-written summary. The copying mechanism can choose sub-sequences in the input and put them at proper places in the output. During testing, the model only sees the input table and automatically generates a text summary.

GPT-2 [9] is a large language generation model pre-trained over 40GB of text data crawled from the Web. Unlike CopyNet, which must be learned from scratch, GPT-2 already learns general language patterns and needs less task-specific training data. It is a good fit for our task as we only collected summaries for 200 tables. GPT-2's model consists of the decoder part of the Transformer [12], and was trained with a causal language modeling objective. Since there is no encoder in GPT-2, the model does not have a clear separation between the input table and the target summary. Therefore, we concatenate the table and the summary, separated by a special "#summary#" token. We train GPT-2 on this text input to predict every token in the sequence conditioned on its previous tokens. For testing, we feed GPT-2 with the original table + "#summary" and let GPT-2 predict the following tokens.

Text-to-Text Transfer Transformer (T5) [10] is another large, pre-trained language generation model. T5 employs both the encoder and the decoder part of the Transformer, so that it can be trained and tested with standard (table, summarization) pairs as used for CopyNet. Compared to GPT-2, T5 is pre-trained on an even larger corpus of 645GB text data using a cleaned subset of Common Crawl. T5 achieved state-of-the-art results on several summarization benchmarks [10].

Implementation Details. We used the AllenNLP [2] implementation of CopyNet, with 300-dimensional GloVe embeddings, one-layer bidirectional LSTM for encoding, and one-layer LSTM for decoding. GPT-2 used the Huggingface [13] implementation. The small version of GPT-2 was used due to GPU memory limitation.[4] Training follows the hyper-parameters recommended by Wolf et al. [13]. T5 used the implementations provided by the authors.[5] It is trained on Google TPU using the default hyperparameters. For all models, we only used the first 256 tokens from the table.

The training process was through 5-fold cross-validation. Each fold used 160 tables for training and 40 tables for testing. To test the model, we compare the automatically generated summary with the best summary that received the most votes from the judges. We use ROUGE and BLEU as our evaluation metrics. During training, we used any summary that had non-zero votes. For each table, 3.6 out of the 5 candidate summaries get at least one vote. This brings more training data than just using the top-voted summary, and was shown to lead to slightly better experiential performance.

[3] The number of judges were based on our budget.

[4] GPU type is Nvidia GeForce RTX 2080 Ti with 11GB memory

[5] https://github.com/google-research/text-to-text-transfer-transformer

Table 1: Evaluation results of three automatic summarization methods. We compare GPT-2 against CopyNet and T5 against GPT-2 for statistical significance testing. ‡ denotes significance at the 0.005 level.

Method	ROUGR-L	ROUGE-1	ROUGE-2	BLEU
CopyNet	0.030	0.041	0.012	0.80
GPT-2	0.200‡	0.272‡	0.073‡	5.35‡
T5	**0.276**‡	**0.362**‡	**0.143**‡	**10.43**‡

3.2 Experimental Results and Discussion

The evaluation results of three models are listed in Table 1. Among the three models, CopyNet achieves the weakest performance on all metrics. CopyNet is not pre-trained and needs much more training examples than our 160 tables. On the other hand, the two pre-trained models, GPT-2 and T5, can generate reasonable summaries when trained on the limited amount of examples. Both of them obtain substantial and statistically significant improvements compared to CopyNet. T5 is the best model for this task.

Table 2 shows a T5-generated summary for the table "List of number-one albums of 2012 (Finland)" (corresponding to the example in Table **??**). It is seen that the machine-generated summary is in fluent natural language. The model is also capable of picking valuable attributes, e.g., the "Album" and "Artists" columns, while ignoring less valuable attributes, e.g., the "Reference" column. However, we found several mistakes in the machine-generated summaries when manually comparing the human and machine-generated texts. We here conduct an error analysis on all tables and classify the most frequent errors into the following categories:

- *Wrong Quantity* (found in 84% summaries): T5 was prone to making errors in mathematical calculations. For example, given the table "Indiana Mr. Basketball Award Winners,"[6] the generated summary mentions "Purdue won 4 consecutive in the 1960s," but in fact Purdue only won 3 consecutive.
- *Wrong Reference* (found in 38% summaries): Some summaries fail to refer to the right attribute. For example, the generated summary mentions "the highest mountain in France is Mont Blanc, at 4,810 meters..." for table "List of Alpine peaks by prominence."[7] This summary is based on the "Elevation" column of the table, but the question asked for "Prominence."

In summary, current state-of-the-art text generation models are capable of generating natural language summaries from tables even with a small amount of training examples. However, they do not work well for knowledge reasoning or question answering when complex calculations are involved. To address the above problems, a possible research direction for future work is to combine semantic parsing with natural language generation methods.

4 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have introduced the task of table summarization in conversational search and developed a test collection using crowdsourcing. The test collection consists of 200 questions, each with a corresponding result table. Each table comes with five manually created candidate summaries, along with both sentence-level and summary-level quality assessments. We have employed three neural language generation models as SOTA baselines, performed an experimental comparison of them, and identified two main classes of errors made by these methods.

This paper represents an important first step towards tabular data presentation in conversational search, where the focus was on developing a benchmark test collection. For baselines, we have focused exclusively on the newest generation of abstractive summarization methods. Based on the dataset, it would be interesting to test how more traditional summarization methods [1, 5, 6, 11] perform on this task. Additionally, it can be used to design novel approaches that consider the unique characteristics of this task.

Acknowledgments. This work was partially supported by the National Science Foundation (NSF) grant IIS-1815528.

Table 2: A manually-written summary and a summary generated by T5. Both manage to answer the question correctly and provide extra information contained in the table.

Question q	What album had the most sales in 2020 in Finland?
Result table T	List of number-one albums of 2012 (Finland)[8]
Manual	I saw a table showing sales figures of around 10 albums of 2012. Of these "vain elamaa" by various artists ranked highest regarding sales. It is performed by various artists. The second rank is held by "koodi" by robin.
T5-Generated	I found a table of the top 10 albums of 2012. The most sold album was "vain elÃďmÃď" by various artists. It sold 164,119 copies. The next highest album was "koodi" by robin.

REFERENCES

[1] Jieying Chen, Jia-Yu Pan, Christos Faloutsos, and Spiros Papadimitriou. 2013. TSum: Fast, Principled Table Summarization. In *Proc. of ADKDD '13.*
[2] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proc. of Workshop for NLP Open Source Software (NLP-OSS).*
[3] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proc. of ACL '16.*
[4] Braden Hancock, Hongrae Lee, and Cong Yu. 2019. Generating Titles for Web Tables. In *Proc. of WWW '19.*
[5] Dino Ienco, Yoann Pitarch, Pascal Poncelet, and Maguelonne Teisseire. 2013. Knowledge-Free Table Summarization. In *Proc. of DWKD '13.*
[6] Parag Jain, Anirban Laha, Karthik Sankaranarayanan, Preksha Nema, Mitesh M. Khapra, and Shreyas Shetty. 2018. A Mixed Hierarchical Attention Based Encoder-Decoder Approach for Standard Table Summarization. In *Proc. of NCACL '18.*
[7] Ming-Ling Lo, Kun-Lung Wu, and Philip S. Yu. 2000. TabSum: A Flexible and Dynamic Table Summarization Approach. In *Proc. ICDCS '00.*
[8] Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *Proc. of ACL-IJCNLP '15.*
[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners (Preprint).
[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683
[11] K. Selçuk Candan, Huiping Cao, Yan Qi, and Maria Luisa Sapino. 2009. AlphaSum: Size-Constrained Table Summarization Using Value Lattices. In *Proc. of EDBT'09.*
[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proc. of NIPS '17.*
[13] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, RÃľmi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771
[14] Shuo Zhang and Krisztian Balog. 2020. Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Trans. Intell. Syst. Technol.* (2020).

[6] https://en.wikipedia.org/wiki/Indiana_Mr._Basketball#Award_winners
[7] https://en.wikipedia.org/wiki/List_of_Alpine_peaks_by_prominence