DEIM Forum 2021 F14-3

Toward Description Generation for Tables in Scientific Articles

Junjie H. XU†, Wiradee IMRATTANATRAI††, and Makoto P. KATO††

† Graduate School of Comprehensive Human Sciences, University of Tsukuba
1–2, Kasuga, Tsukuba, Ibaraki, 305–8550, Japan
†† Faculty of Library, Information and Media Science, University of Tsukuba
1–2, Kasuga, Tsukuba, Ibaraki, 305–8550, Japan
E-mail: †s2021705@s.tsukuba.ac.jp, ††{wiradee,mpkato}@slis.tsukuba.ac.jp

Abstract The task of description generation for tables in scientific articles requires not only table contents but also retrieval and a combination of other information outside the table . The table is sometimes too consolidated and informative, challenging to understand if only use the table. This paper proposes generating table descriptions by combining the text body of a paper and table information through the learning to rank (LTR) method to find sentences related to the table. We introduced a labeling strategy for the training data, proposed LTR features concerning table structures, and investigated these features' effectiveness. We used the DocBank dataset to evaluate the effectiveness of the proposed method in the experiment.

Key words Scientific article, table, learning to rank, description generation

1 Introduction

The pace of academic research has been accelerated more than ever in recent years [1]: the number of scientific articles archived on the Internet has already exceeded 100 million, and the number of scientific articles still increases steadily. As the total number of scientific articles increases, researchers need to read more scientific articles than ever. It becomes difficult for them to read the whole scientific article's content and need efficient strategies to keep tracking the scientific trends.

It is common for authors to use tables to present quantitative data obtained through experiments. Tables take account for providing a better view of showing findings and helping readers quickly understand the content in scientific articles. Hence, the table is considered an essential part of scientific articles. However, it is usually not enough to understand the quantitative data in the table only with the table's content itself. For example, it is not easy to find out the original meaning of abbreviations or academic terms without referring to a scientific article's body. If we use only the table for description generation, such a description could not address the problem mentioned earlier, such as abbreviations of academic terms, and can deviate from the author's original intention expressed by the table. Therefore, we proposed that it is necessary to extract the sentences related to the tables in the paper to produce a description that provides a better understanding of a table in the scientific article.

Another problem of extracting sentences from the paper's text body is to ensure such sentence is relevant to the table. Otherwise, such a sentence might harm the description generation, which leads the generated description ambiguous and unclear. As the example shown in Figure 1, assumes only using the table information for generating a description of the table: the word "Method" located in the row header, which is considered an essential part of table structure. However, it is arguable that "Method" is ambiguous and highly irrelevant to the author's intention of using such table, so such information is not favorable to be taken. Also, the word "Ablation Study" is a word in the table's caption written by the author him/herself, while it is not in the table. Its existence in the caption shows a clear that such word has high relevance to the author's intention of using this table and is positively related to this scientific article's general idea. However, we also notice that it is also possible to generate sentences that unrelated to the author's intentions of using the table, the different meaning in such sentences, another word "Ablation Study" in our example in Figure 1 appears quite a few and so many sentences could be used to refer. Finding out the importance of the word "Ablation Study" could be solved by referring to the body text sentences. Although the meaning of "Ablation Study" also could be easily found in the body text of the example scientific article, if such sentence is not strongly related to the table, the meaning of the sentence might deviate from the intention only by using this table, which may cause ambiguity in the generated description. Here, we argue that in the case of table description generation, the sentences extracted should have high rele-

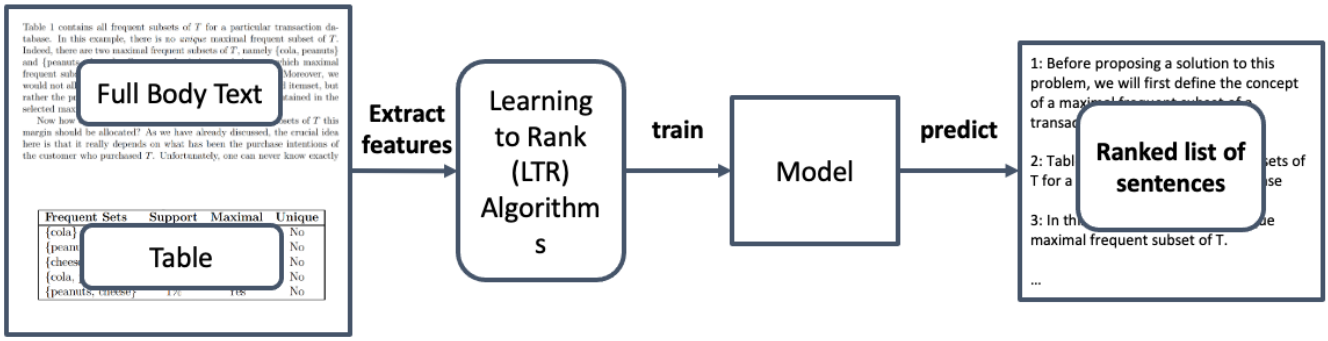

Figure 1 Overview of our approach (the full body text and the table in this example are derived from [2]).

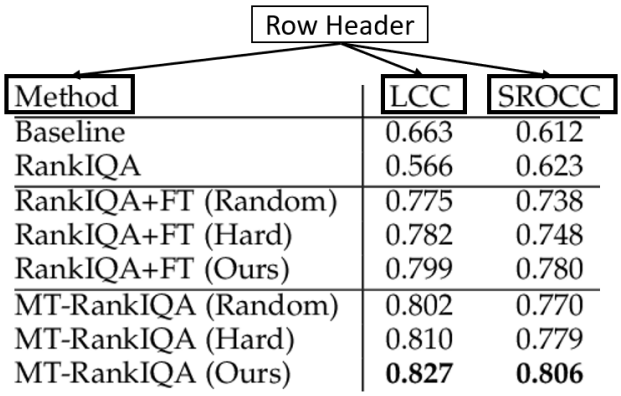

Figure 2 Example of a table derived from [5].

vance to the table, which ensures such context information does not differ from the author's intention using this table, so the generated description would not deviate from the original intention of the author as well as the general meaning of the paper. It could be expected that the generated description could help with a better understanding of the table content for the scientific article reader using additional information provided by such a relevant sentence. In this context, we believe that the additional embedding of these sentences into LSTM [3] or Transformer [4] based generator could improve their performance.

Therefore, to extract the sentences mentioned above, in this paper, we propose an approach that uses the Learning to Rank (LTR) method to extract such sentences in scientific articles related to the table and rank them based on relevance to the table. The overview of our approach is given in Figure 1. LTR can rank the documents according to their relevance score with regards to the query in information retrieval [6]. It is also noted that in general ranking problems utilizing LTR methods, human-labeled relevance scores are given to the pair of document-queries. In the problem described above, in the context of extracting relevant sentences of such table for table description generation, it is possible to utilize the LTR method to learn a ranked-list of sentence-table (query-document) in the scientific article from the given relevance scores. However, as mentioned before, readers need proper expertise to understand a scientific article to understand scientific articles and label these tables and sentences. Creating such a dataset is considerably costly. In this work, we use the alternative solution to label the sentence-table pair using caption by the authors near the table and use BM25 [7] score, which is a continuous type of relevance score of sentence-caption to generate relevance scores. More specifically, the relevance between the table's caption and sentence in the scientific article's body text is given as relevance scores. We tested our idea and on the dataset DocBank [8], which contains 500,000 document pages (19,638 pages have the table(s)) with 12 types of semantic units included tables, captions of tables, and text body in the scientific articles. The comparison and discussion of effectiveness among three types of labeling table-sentence was given in Section 4.

The contributions in this paper are summarized as follows: (1) a method using the caption of the table as the label to train an LTR model that capable of extracting sentences related to the table in a scientific article, (2) we proposed features for LTR of the table-sentence pair (corresponds to the query-document pair in LTR, the features from a group (ii) to (v), see Table 1), (3) we further tested our idea on an algorithm in LTR called ListNet to conducted some experiments evaluating the effectiveness of our proposed methods described in (1) and (2), the results derived from our experiment could be used as a reference for our future works on description generation for a scientific article. This paper is organized as follows. In the following section, we provide a comprehensive review of the literature about table structure recognition. Then we introduce the methodology about utilizing LTR on our problem is given in Section 3. The details of conducted experiments are given in section 4. Finally, the

conclusion of this paper and our future works are given in Section 5.

2 Related Work

2 1 Table Structure Detection and Understanding

As a structured form of data, using a table could show much more quantitative data than textual information. Since the first introduction of the table dataset in ICDAR 2013 table competition [9], image-based table detection has gained more research interest. Recently, various researches have been conducted using table data, such as tables on the Internet (such as Wikipedia's Infobox), tables containing financial information (such as asset estimation tables), and technical specification tables (such as electrical product manuals). For tables in scientific articles, recently proposed DeepFigures [10], TableBank [11], DocBank [8] enable access to the research on understanding tables in scientific articles.

In terms of the semantic meaning of table structure, literature in table structure detection reveals that understanding semantic meaning such as entities in the column as well as row header are essential to understand the table [12]. Moreover, Deng et al. [13] about table embedding pointed out that the different meanings of row and column headers and the context of table embedding row and column headers should be separately considered when applying word embedding to table.

2 2 Description Generation

In recent years, models based on Neural Networks have shown significant progress in generating more expressive text generation, such as captions for images, machine translation, etc. Impressive performance has been shown on generating short descriptive texts on the tables on the website [14]. Since 2017, Wiseman et al. [15] point out that table description generation is the next-problem of generating a more informative description extracting structured data within tables. To tackle such a problem, they introduced a dataset called Rotowire, which contains standard statistical tables of basketball games around 2017 and description texts created by professional commentators; text generation method based on deep learning has also been studied. Some research has been conducted [16], [17] achieved some success. As the differences in table domains, we argue our task on table description generation in a scientific article is complicated. For example, (1) title information such as header in the basketball game is fixed, but there are many types of tables in scientific articles, varies among academic fields, (2) it is impossible to understand if no expert knowledge is included due to a large number of academic terms within these tables, (3) the difference between the intention of using the table in the scientific articles. Tables in scientific articles are used to show the findings that the author has obtained. Even if some of the information in the table is bold or italic-styled, it is hard to say such pointed information has perfectly matched the paper's content. Therefore, due to the insufficient information in the table for generating descriptions in scientific articles, the methods proposed in the above research only focus on tables. Therefore, to tackle the problem of table description generation in scientific articles, it is clear that access to information outside the tables is required.

3 Methodology

This section first explains why the sentences are needed to be ranked and defines the problem addressed in the paper. Then we describe features used for LTR, and lastly, we explain the LTR algorithm used in our experiments and labels for training LTR models.

3 1 Why are Rankers needed for description generation?

The term "Description" mentioned in this paper can be considered more generalized to summarize the described object, namely in this work. So it could be seen as a summarizing task that summarizes such an object with a variety of sources, even forms of data (e.g., not only using the information in the table but also involving sentences in the paper body in this work). As the task of document summarization, for tasks of generating description, it is more likely to suffer from the slow and wrong encoding of very long input of information as the same problem of encoding of the very long document occurs in document summarization tasks since more information is considered and likely to be included [18].

3 2 Problem Definition

The problem tackled in this work is defined as, given a table in a paper, the paper's ranking of sentences to the relevance to the table. Formally, the input consists of a table q and a set of sentences D, both of which are derived from the same paper. The output of our problem is a ranked list of sentences, i.e. $\mathbf{r} = (r_1, r_2, \ldots, r_{|D|})$ where $r_{\pi(i)} = d_i$ and $\pi(i)$ denotes the rank of the i-th document in D. For example, when a pointwise LTR model $f(d)$ is applied to this problem, $f(d)$ is considered as an estimated relevance, and, accordingly, π is defined as a function such that $\pi(i) < \pi(j)$ if and only if $f(d_i) > f(d_j)$. When a pairwise LTR model $f(d_i, d_j)$ is used in our problem, π is defined as a function such that $\pi(i) < \pi(j)$ when $f(d_i, d_j)$ is large.

3 3 Features for Learning to Rank

We introduce not only standard features used for LTR but also some features unique in our problem setting. We compute features for three variants of the table-related features concerning table structures. A brief description of these fea-

tures is given in Table 1. Those table parts are denoted as W (whole table), R (row headers) and O (table contents except for R, see Figure 1) [13]. Each feature is characterized by whether it is dependent on the only table (**Q**), only sentence (**D**), and both table and sentence (**Q**–**D**). Besides, details of features are given below:

3 3. 1 Semantic features

Follows Standard LTR features [6], We considered four lexical matched features (term frequency (TF, fID: 1-4), inverse document frequency (IDF, fID: 5-8), TF*IDF (fID: 9-12) and BM25 [7] (fID: 13-16)) and they was computed as follows:

$$tf(q_i, d) = f_{q_i,d} \tag{1}$$

$$idf(q_i) = \log(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}) \tag{2}$$

$$tfidf(t, d, D) = f_{q,d} \times idf(q_i) \tag{3}$$

$$\text{BM25}(Q, D) = \sum_{i=1}^{n} \text{IDF}\,(q_i) \cdot \frac{f\,(q_i, D) \cdot (k_1 + 1)}{f\,(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{argdl}}\right)}, \tag{4}$$

3 3. 2 Non-semantic features

Given a table as query, document length (DL) denotes the count of semantic units within the sentence, namely the length of a document. Query length (QL) [6] is the number of words in the table. Contents in the table contain either numeral or character words, so we proposed two simple indicators: Numeral frequency (NF) denotes the percentage of words containing numeral content, and word frequency (WF) denotes the percentage of words containing character content vice versa. Features in the group (*ii*) represent the document's length where the sentence is located (fID:13). Features in the group (*iii*) represent the total count of both numbers and words, namely the total count of numbers or words in the table's corresponding field (fID:14-16). Features in the group (*iv*) represent the ratio of the frequency of numbers and words (fID:17-19) and vice versa (fID:20-22) in the corresponding field of the table. Features in the group (*v*) represent the ratio of the frequency of numbers (fID:23) and words (fID:24).

3 4 Learning to Rank

An LTR model is a model to rank documents in response to a query and is trained with training data that consist of query-document pairs with relevance. The LTR model's generalized goal is to learn π for ordering a set of given documents. The function π is considered appropriate if the output ranking can achieve high retrieval effectiveness in terms of a particular evaluation metric (e.g., nDCG).

In general ranking problem utilizing Learning to Rank methods, a human-labeled relevance score is given to the query-document pair. We note that there is no publicly available ground truth in our problem. It is not very clear what are relevant sentences in our problem. We hypothesized that relevant sentences to a table are those relevant to table captions and examined several approaches that provide pseudo labels to sentences base on this hypothesis. In this work, we treat a table as a query and a sentence as a document. We label the table sentence pair with the BM25 score of the sentence and the table's caption as the relevance score for training the ranking model.

4 Experiments

This section first introduces the dataset used in the experiments, i.e., DocBank, and describes the experimental settings and evaluation metrics. Finally, we explain comparative methods, including various LTR models, and present experimental results.

4 1 Dataset

DocBank [8] is a dataset containing over 500,000 scientific papers, of which 19,638 pages are having at least one table, in which each part of the paper is categorized into such as tables, bodies, and table captions, etc. While this dataset was originally proposed for image-based table recognition tasks, it can be used for our purpose since it also contains text information of tables and table captions.

We note that the DocBank dataset was designed for image-based table recognition tasks, not NLP tasks. Specifically, though DocBank was designed for where and what an entity is, we extract its meaning symbolically while hard to extract its meaning semantically.

4 2 Experimental Settings

For experiments in this paper, we extract 500 pages of scientific articles with one table for each, which are formatted in one-column and having only one table and its caption, refer to the original image corresponding to the dataset. As a result, we could extract data while avoiding most of the mismatching caused by varying table structures, even only using the simple rule-based method in this work.

We labeled our dataset using continuous labels (BM25 score of the sentence and the caption of the table) and compared the performance trained our rankers. Statistics details of the data used in our experiments are given in Table 2. The continuous label is simply defined as the raw BM25 score, where $\text{BM25}(q, d)$ is the BM25 score of a sentence d for a table q, where the table and sentence are treated as a query and a document, respectively:

$$s_{\text{cnt}} = \text{BM25}(q, d). \tag{5}$$

Table 1 Features for LTR

fID	Name	Feature description	Category	Group
1	TF-W	TF (Term Frequency) in the whole table	**Q–D**	(*i*)
2	TF-R	TF in the row headers	**Q–D**	(*i*)
3	TF-O	TF in the table w/o headers	**Q–D**	(*i*)
4	IDF-W	IDF (Inverse Document Frequency) in the whole table	**Q**	(*i*)
5	IDF-R	IDF in the row headers	**Q**	(*i*)
6	IDF-O	IDF in the table w/o headers	**Q**	(*i*)
7	TF*IDF-W	TF*IDF in the whole table	**Q–D**	(*i*)
8	TF*IDF-R	TF*IDF in the row headers	**Q–D**	(*i*)
9	TF*IDF-O	TF*IDF in the table w/o headers	**Q–D**	(*i*)
10	BM25-W	BM25 of the whole table	**Q–D**	(*i*)
11	BM25-R	BM25 of the row headers	**Q–D**	(*i*)
12	BM25-O	BM25 of the table w/o headers	**Q–D**	(*i*)
13	DL-Sentence	DL (Document Length): length of the sentence	**D**	(*ii*)
14	QL-W	QL (Query Length) of the whole table	**Q**	(*iii*)
15	QL-R	QL of the row headers	**Q**	(*iii*)
16	QL-O	QL of the table w/o headers	**Q**	(*iii*)
17	NF-W	NF (Numeral Frequency) in the whole table	**Q**	(*iv*)
18	NF-R	NF in the row headers	**Q**	(*iv*)
19	NF-O	NF in the table w/o headers	**Q**	(*iv*)
20	WF-W	WF (Word Frequency) in the whole table	**Q**	(*iv*)
21	WF-R	WF in the row headers	**Q**	(*iv*)
22	WF-O	WF in the table w/o headers	**Q**	(*iv*)
23	NF-Sentence	NF of the sentence	**D**	(*v*)
24	WF-Sentence	WF of the sentence	**D**	(*v*)

Table 2 Statistics of the data used in the experiments.

The number of extracted tables (pages)	500
The number of table-sentence pairs	7,689
Max. of BM25 score	60.236
Min. of BM25 score	0
Avg. of BM25 score	3.263
The average number of sentence terms	21.16
The number of table-sentence pairs with BM25 $\neq 0$	5,014

The statistics of the extracted data are shown in Table 2. Five-fold cross validation was used in our experiments. We use three parts for training in each fold, one part for validation, and the remaining part for the test, respectively.

Our baseline features are the standard lexical matching features (fID: 1-12), which are commonly extracted for each query-document pair in most LTR tasks. We proposed features from the feature group (*ii*) to group (*v*), which is the count of semantic units of the sentence or the table, and the number/word ratio of the sentence, or the table, respectively in our experiment. For the LTR model, we used ListNet [19]: a listwise LTR model based on neural networks.

The number of epochs was set to 5. The implements of these ranking models are based on PTRanking [20]. We used nDCG@10 to evaluate the ranking performance across the sets of features, which evaluates the top-10 ranked sentences in the extracted page of the corresponding paper [21], as our primary evaluation metric for comparing the baseline methods with different relevance labeling strategies.

4 3 Experiment Results

We examined the results of our proposed method of using LTR approaches to rank the sentences in the paper using learning features extracted from the table contents and sentences, with relevance judgment given by BM25 score between table captions and sentence. We compared the effectiveness of using features of the contents with considering row headers and whole table excluding row headers of the table (W, W+R, W+O, W+R+O) by comparing ranking performance using ListNet [19], (3) the effectiveness of proposed features in each proposed feature group measured in nDCG@10 (features in the group (*ii*), (*iii*), (*iv*), (*v*)) respectively.

For each proposed feature group (features group (*ii*), (*iii*), (*iv*) and (*v*)), we add them to the features of the baseline ranking using only features in the group (*i*) and learn a ranking model to investigate the impact of adding features in proposed feature groups. We focus on features that bring the improvement of ranking performance. Among adding features in proposed feature groups (*ii*), (*iii*), (*iv*) and (*v*), regarding the improving features, we found that adding features in feature group (*ii*) brings significant improvement ranking performance (5% in W, W+O, W+R+O and 4% in W+R,

Table 3 Comparison of performance (measured in nDCG@10) using ListNet [19] including consideration of table structures (see Table 1), total counts of the features are given by number in parentheses after the nDCG@10 score.

	W	W + R	W + O	R + O	W + R + O
(*i*)	0.494(4)	0.509(8)	0.495(8)	0.502(8)	0.499(12)
(*i*)+(*ii*)	0.540(5)	0.549(9)	0.545(9)	0.543(9)	0.555(13)
(*i*)+(*iii*)	0.492(5)	0.495(10)	0.498(10)	0.500(10)	0.509(15)
(*i*)+(*iv*)	0.490(6)	0.502(12)	0.499(12)	0.501(12)	0.496(18)
(*i*)+(*v*)	0.516(6)	0.515(10)	0.512(10)	0.522(10)	0.520(14)
(*i*)+(*ii*)+(*v*)	0.547(7)	0.555(11)	0.551(11)	0.517(11)	0.550(15)
full	0.543(10)	0.549(17)	**0.556**(17)	0.549(17)	0.435(24)

R+O), and adding features in feature group (*iv*) slightly improves the ranking performance (2% in W, W+R, W+O, R+O and W+R+O and 1% in W+R). Two top groups ((*ii*), (*v*)) of proposed features were identified. The count of semantic units within a sentence and the ratio of numeral terms and natural language word terms are useful indicators for feature selection in LTR in this task. We also note that the group (*ii*) and (*v*) are in the D category, namely the document level feature. Following Macdonald et al. [22], a document level feature is more capable of increasing effectiveness than a query level feature because a query level feature has the same value across all documents in training query-document pairs of the same query under problem setting in this work.

Next, we compare the three variants of table-related features by comparing performance using each of two features concerning different parts of the table W+R (whole table and row headers), W+O (whole table and whole table excluding row headers), and R+O (row headers and whole table excluding row headers) feature pairs which share the same numbers of features. We observe no significant differences in nDCG@20 using W+R, W+O, and R+O feature pairs, which implies our method processing table cells by categorizing them into Row headers (R) or Others (O) in this work could not fully leverage the insight structured information of cells in tables, a more advanced table representation method is needed.

From the results comparing each of these proposed feature groups, we added both features in the group (*ii*) and (*v*) into the baseline features (*i*). Comparing to adding features in only one feature group and only baseline features, the ranking performance increases except for using R+O features with those using the whole table that decrease the performance, which does not deploy the whole table's features (W). This implies that if more features are included, it is necessary to consider features of the whole table to rank sentences in the paper using our approach.

Finally, we examine the difference of ranking performance listed in the table 3 among all settings. Moreover, using the W+O features of features in all groups (17 features in total) achieves the best performance measured in nDCG@10 score (0.556), which implies the optimum feature set in this work. We also observe that using all features in all groups achieves the lowest performance (0.435), whose performance decreases significantly with the increase in features (24 in total).

5 Conclusion and Future Work

In this paper, we introduced the problems and challenges of description generation for tables in scientific articles. We proposed a method to extract sentences related to a table in an academic paper using LTR to tackle this problem. In the experiment, we used ListNet, an LTR algorithm of the listwise approach, to evaluate the effectiveness of using different features by comparing the ranking performance among different sets of features in ranking sentences by the relevance to tables in scientific articles. From the results of conducted experiments, We found that (1) length of sentences and ratio of number/word of sentences are useful features for ranking the sentences related to the table, (2) the feature related to the sentences are more effective than the feature related to the table using our approach to extract sentences related to the table. (3) Our current approach on the feature representations of only separating row headers with others in this work could not fully leverage the insight relational information of table structure. A more advanced table representation method, such as extracting columns using positional information or using table embedding, is needed. Regarding the verification of each feature proposed in this paper and the verification of whether the sentences and tables related to the table extracted from the text of the paper can be used for table description using a sentence generation model such as encoder-decoder models.

Acknowledgement This work was supported by JSPS KAKENHI Grant Numbers JP18H03243 and JP18H03244, and JST PRESTO Grant Number JPMJPR1853, Japan.

References

[1] Madian Khabsa and C Lee Giles. The number of scholarly documents on the public web. *PloS one*, 9(5):e93949, 2014.

[2] Tom Brijs, Bart Goethals, Gilbert Swinnen, Koen Vanhoof, and Geert Wets. A data mining framework for optimal product selection in retail supermarket data: the generalized profset model. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 300–304, 2000.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[5] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1862–1878, 2019.

[6] Tie-Yan Liu. *Learning to rank for information retrieval*. Springer Science & Business Media, 2011.

[7] S. Robertson, S. Walker, Susan Jones, M. Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In *TREC*, 1994.

[8] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*, 2020.

[9] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1449–1453. IEEE, 2013.

[10] Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. Extracting scientific figures with distantly supervised neural networks. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*, pages 223–232, 2018.

[11] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. TableBank: Table benchmark for image-based table detection and recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1918–1925, Marseille, France, May 2020. European Language Resources Association.

[12] Shuo Zhang and Krisztian Balog. Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(2):1–35, 2020.

[13] Li Deng, Shuo Zhang, and Krisztian Balog. Table2vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*, pages 1029–1032. ACM, 2019.

[14] Braden Hancock, Hongrae Lee, and Cong Yu. Generating titles for web tables. In *The World Wide Web Conference*, pages 638–647, 2019.

[15] Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[16] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6908–6915, Jul. 2019.

[17] Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. A hierarchical model for data-to-text generation. In *European Conference on Information Retrieval*, pages 65–80. Springer, 2020.

[18] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, 2018.

[19] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[20] Hai-Tao Yu. Pt-ranking: A benchmarking platform for neural learning-to-rank. *arXiv preprint arXiv:2008.13368*, 2020.

[21] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[22] Craig Macdonald, Rodrygo LT Santos, and Iadh Ounis. On the usefulness of query features for learning to rank. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2559–2562, 2012.