Knowledge and Information Systems (2023) 65:1713–1734
https://doi.org/10.1007/s10115-022-01815-0

REGULAR PAPER

Matching news articles and wikipedia tables for news augmentation

Levy Silva[1] · Luciano Barbosa[1]

Received: 3 June 2022 / Revised: 13 August 2022 / Accepted: 12 December 2022 /
Published online: 27 December 2022

Abstract

Nowadays, digital-news understanding is often overwhelmed by the deluge of online information. One approach to cover this gap is to outline the news story by highlighting the most relevant facts. For example, recent studies summarize news articles by generating representative headlines. In this paper, we go beyond and argue news understanding can also be enhanced by surfacing contextual data relevant to the article, such as structured web tables. Specifically, our goal is to match news articles and web tables for news augmentation. For that, we introduce a novel BERT-based attention model to compute this matching degree. Through an extensive experimental evaluation over Wikipedia tables, we compare the performance of our model with standard IR techniques, document/sentence encoders and neural IR models for this task. The overall results point out our model outperforms all baselines at different levels of accuracy and in the mean reciprocal ranking measure.

Keywords Web table retrieval · Neural information retrieval · News understanding · News augmentation · Table matching

1 Introduction

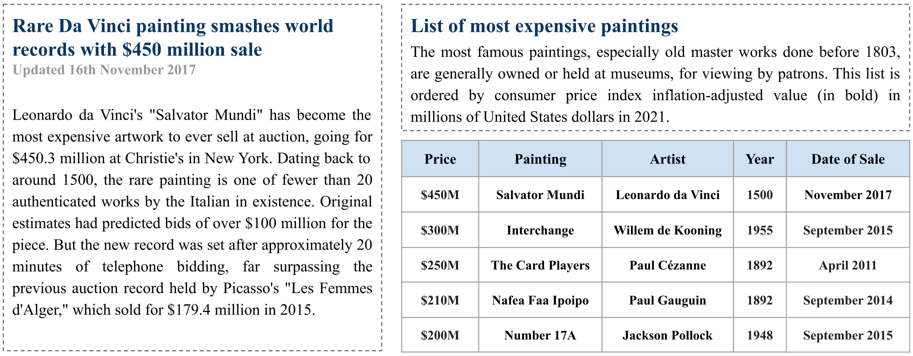
Web tables are a huge and rich corpus of relational data from the Internet [8].[1] Beyond representing complex data, tables also enable quick understanding of entity relationships due to their well-organized structure. In short, web tables are a valuable tool to categorize and publish real-world information [42]. As a result of that, over the last years, a growing body of work has begun to explore web tables for several downstream applications [4]. For instance, tables have been widely utilized for Question Answering (QA), where the goal is to retrieve a table that answers a query from a table collection [6, 32, 46]. Not limited to table retrieval, other studies focus on table argumentation, extraction and interpretation [57].

[1] The Web has over 14.1B tables Cafarella et al. [3].

✉ Levy Silva
lss9@cin.ufpe.br

Luciano Barbosa
luciano@cin.ufpe.br

1 Universidade Federal de Pernambuco, Recife, Pernambuco, Brazil



Rare Da Vinci painting smashes world records with $450 million sale

Updated 16th November 2017

Leonardo da Vinci's "Salvator Mundi" has become the most expensive artwork to ever sell at auction, going for $450.3 million at Christie's in New York. Dating back to around 1500, the rare painting is one of fewer than 20 authenticated works by the Italian in existence. Original estimates had predicted bids of over $100 million for the piece. But the new record was set after approximately 20 minutes of telephone bidding, far surpassing the previous auction record held by Picasso's "Les Femmes d'Alger," which sold for $179.4 million in 2015.

List of most expensive paintings

The most famous paintings, especially old master works done before 1803, are generally owned or held at museums, for viewing by patrons. This list is ordered by consumer price index inflation-adjusted value (in bold) in millions of United States dollars in 2021.

Price	Painting	Artist	Year	Date of Sale
$450M	Salvator Mundi	Leonardo da Vinci	1500	November 2017
$300M	Interchange	Willem de Kooning	1955	September 2015
$250M	The Card Players	Paul Cézanne	1892	April 2011
$210M	Nafea Faa Ipoipo	Paul Gauguin	1892	September 2014
$200M	Number 17A	Jackson Pollock	1948	September 2015

(a) CNN News **(b) Wikipedia Table**

Fig. 1 Improving news understanding by matching a correlated web table. A concrete example of the news/table matching - (adapted from the original web page) (**a**) https://edition.cnn.com/style/article/da-vinci-salvator-mundi-sale-christies (**b**) https://en.wikipedia.org/wiki/List_of_most_expensive_paintings

At the same direction, digital news has gained popularity. News reading habits have progressively moved from conventional media such as newspapers or TV to the Internet [40], where millions of articles are published every day [15]. However, given the today's news deluge, online readers can be overwhelmed to fully understanding the content of a news story [23]. One approach to cover this gap is to outline the news by highlighting the most important facts. For example, recent studies sum up news articles by adopting representative headlines [12, 15, 55]. Likewise, other papers utilize sentence summarization strategies for creating document representations [1, 29, 36]. For both fronts, the goal is just to capture relevant data of the news.

In this paper, we focus on news augmentation and argue news understanding can also be enhanced by surfacing contextual data relevant to the article, as structured web tables. For instance, popular services such as Google News[2] or Microsoft News[3] could benefit from this *News-Table Matching* by providing associated content to the news articles for their readers. Specifically, we aim to automatically find tables related to news articles. Figure 1 shows a concrete example of this news/table match: Fig. 1a presents an article about a rare world painting, and Fig. 1b depicts a Wikipedia table that lists the most expensive arts in the world. In this example, the table provides additional information about the central topic of the story, i.e, the rare Da Vinci painting. Moreover, by looking at the table, a reader could answer potential questions related to the topic, e.g., what is the second most expensive painting in the world? By looking at the table, the answer is Interchange by Willem de Kooning. Lastly, we can also confirm the selling price of this art by connecting the news and the table (i.e., $450 million in these two sources).

Furthermore, from a fake news perspective, this linking can also improve the credibility of articles and help in preventing rumor spread, since we can verify their facts across two different sources of information. For example, by matching the table in Fig. 1b with the related news article, *Most Expensive Paintings: A Look at the World's Most Valuable Paintings*,[4] we can verify that they diverge with respect to the price of the *Nafea Faa Ipoipo? painting by*

[2] https://news.google.com.

[3] https://news.microsoft.com.

[4] https://artincontext.org/most-expensive-paintings.

Paul Gauguin, since the article informs that it was sold for around $300 million, while its value in the Wikipedia table is $210 million.

In fact, similar research has shown the news consumption experience enhancement by linking sentences in the article with table cells [21]. In addition, people can achieve higher recall by jointly reading text and tables than by consuming text alone [14]. Lastly, web traffic from recent studies has demonstrated that online readers also explore tables inside Wikipedia pages after looking at news articles [23].

The task of matching news articles and web tables is quite similar to table retrieval for QA. However, it brings novel challenges. News stories can include different entities, categories and objects in the same article. Furthermore, articles are a mixture of unstructured text represented by several aspects, e.g., title, full content, main passage, keywords and so on. In contrast, table retrieval for QA focus on specific intent queries, usually single queries defined by a sequence of few words [46, 56], which limits the application of previous solutions for QA to our problem since they need to handle distinct news features at the same time.

The core challenge of this task is how to construct a robust *News-Table* matching model for computing this similarity degree. Lees et al. [23] address this problem by introducing a BERT-based bi-encoder model which uses features like entities and hypernyms. We go further and propose a cross-encoder matching model for this task that encodes both article and table over the same semantic space. Our model learns a joint-representation from a ⟨*news*, *table*⟩ tuple by applying recurrent networks, attention mechanisms and recent transformers architectures from BERT.[5] Therefore, in our solution we opted for a cross-encoder strategy instead of a bi-encoder one because previous work has shown the former brings superior results for text matching tasks [47].

Specifically, our end-to-end solution for matching news articles and web tables has two cascaded steps. First, similar to previous work [41–43, 46], we retrieve a set of candidate tables by using a standard Information Retrieval (IR) approach whose goal is to efficiently find the highest number of relevant tables for the matching model. Afterward, we use the proposed model to re-rank the candidates in order to obtain the best matching tables to a news story. Our work goes beyond the previous studies that apply BERT for retrieval, in addition to fine-tuning it to the target task [8, 23, 30], since we also consider matching information from attention matrices over the inputs. Finally, we compare the proposed model with standard IR techniques, document/sentence encoders and neural IR models for text matching. Moreover, a statistical hypothesis test confirms our method statistically outperforms all baselines in terms of Mean Reciprocal Ranking ($MRR@50$). Concerning accuracy, our model achieves near 55% accuracy@1 as opposed to the best baselines varying between 13 and 48%. Such results demonstrate our model re-ranks the best matching table at the first ranking positions. In summary, our contributions are as follows:

- We introduce the problem of collocating news articles with structured web tables as a novel ranking task. Furthermore, we also formalize the most used matching features for this task (Sect. 3);
- We present the first news-table corpus from literature. By crawling Wikipedia pages, we collected 275,352 news articles and 298,792 web tables. In addition, our ground truth contains 93,818 matching pairs created by distant supervision strategies (Sect. 5);[6]
- We evaluate previous approaches for table retrieval and table matching in the context of our task, also assessing both single and multi-field (document) ranking methodologies in the experiments. (Sect. 2);

[5] We demonstrate its performance in our ablation study (see results in Table 8).

[6] https://github.com/levysouza/News-Table-Matching.

- We propose a novel BERT-based attention model for computing the similarity degree between news stories and structured tables (Sect. 4);
- We compare the performance of our solution with standard IR techniques, document/sentence encoders, text matching models and neural IR approaches for this task. (Sect. 5).

The remainder of this paper is organized as follows. We begin by covering the related work in Sect. 2. Section 3 formalizes the *news-table* matching problem. Afterward, we introduce the proposed model in Sect. 4. Section 5 describes the experimental setup, and relevant results are in Sect. 6. Finally, we conclude the paper in Sect. 7.

2 Related work

Many previous strategies have been applied to the task of table retrieval such as traditional IR methods [3, 25, 26, 32, 41], probabilistic approaches [33], semantic models [45, 50, 56], machine learning algorithms [2, 6] and neural networks [8, 23, 42, 44, 48]. As follows, we briefly discuss and point out relevant studies.

Liu et al. [25] introduce TableRank: an algorithm adapted from TF-IDF which weighs terms by using three levels: Table Term Frequency - Inverse Table Term Frequency (TTF-ITTF), Table Level Boost-ing (TLB) and Document Level Boosting (DLB). TTF-ITTF measures term frequency over a table metadata, and TLB goes over table-level features such as table-frequency. Concerning DLB, it addresses query-independent features as the overall importance of a document where a table appears. Lastly, the final vector is computed by aggregating the query/table terms in TTF-ITTF, TLB and DLB levels. The matching score between queries/tables is computed using the cosine similarity.

While basic solutions for table retrieval focus on lexical matching (i.e., query/table terms overlap), other studies go beyond by exploring the semantic association between queries and tables. For example, Zhang and Balog [56] propose a semantic similarity model in which the query and the table are represented by semantic spaces (bag-of-concepts and embeddings), as well as words and entities present in the query/table. The similarity between queries and tables is calculated according to two strategies: early fusion, which encodes query and table as a single representation; and late fusion, which computes the pairwise similarity between all query/table terms, and an aggregation function consolidates the similarity (e.g, *max*).

When human-curated data are available, previous studies have been applying machine learning algorithms for ranking the tables. For example,Cafarella et al. [3] train a linear regression model by using query-dependent features such as query/terms overlaps (e.g., the number of times that a query-term occurs in a table column), and query-independent features: page rank, page views and in/out reference links [2].

Regarding neural network approaches, Zhang et al. [54] propose *Table2Vec*: a neural language modeling for embedding tabular data into distinct vector spaces. Specifically, the authors introduce four types of table embedding: *Table2VecW*, *Table2VecH*, *Table2VecE* and *Table2VecE**. Such embeddings encode words in the page title, section title and table elements as well as those entities that appear inside the table cells. Another study, presented by Sun et al. [46], proposes a set of manually designed characteristics and neural network features for table raking. The first one considers word, phrase, and sentence-level aspects, while the network-based features applies bidirectional Gated Recurrent Units (GRUs) to extract query/table context vectors. In the same direction, other solutions have applied transformer-architectures to represent queries and tables for table retrieval [8, 13].

In summary, all cited studies focus on retrieve tables to answer a single query, i.e., a sequence of few words. Our study differs from them since we consider matching news articles to web tables.

Specifically for our task, Lees et al. [23] devise a News-Table matching model based on a pre-trained BERT for news articles. First, the authors pre-train BERT from scratch over a large news corpus (NewsBERT). It uses as features the title, hypernyms and content of the news article, and the page that contains the table. To fine-tune the model, they apply a pairwise learning strategy over a triple composed of a news article, a positive table and a negative table. The resulting model encodes news articles and tables into vectors. Tables are ranked based on cosine similarity (i.e., their BERT-based model is only used as a bi-encoder for the inputs). Their approach, however, has some reproducibility issues since NewsBERT is not public available, and their training data come from private Google resources - the authors create news/table matching pairs by utilizing user activity logs from Web sessions (i.e., the training data are hard to access). In addition, they employ proprietary technology for the extraction of topical entities and hypernyms.

Our work differs from them as we only fine-tune a public BERT in our network architecture, since pre-training a BERT-like model requires enormous computation resources [7].[7] Furthermore, similar to Lees et al. [23], our work also applies a fine-tuning BERT model for this task but, in addition, we further employ an attention/RNN network to produce a different contextual representation of the input. Our model joints recurrent networks, attention mechanisms and transformers architectures. Lastly, we utilize BERT as a cross-encoder, which perform full-attention over the input pair. Recent studies have shown BERT cross-encoders achieve higher performance when compared to bi-encoder ones [47].

3 Background

Problem statement In this paper, we target the task of matching news articles (A) and web tables (T) as follows: given an article a_i and a set of web tables $T = \{t_1, t_2, t_3, ..., t_n\}$, the goal is to find the most relevant table t_i to a_i. The notion of relevance is broad. As we show in Fig. 1, a web table can overview news stories, bring contextual data or answer upcoming questions. Formally, we assume *News-Table matching* as a ranking task, i. e., our goal is to learn a scoring function $f : a_i \times t_j \mapsto \mathbb{R}$ that scores tables in T for a given article a_i to rank them based on news-table aspects.

News-table aspects We represent a news article by three aspects: *title*, *main passage* and *keywords*. The title expresses the central topic of the story. Instead of using the entire news content, we consider the main passage as it compiles the story in a few sentences. For this feature, we collect the meta-description tag from HTML page. Moreover, the keywords represent the most frequent words from the article. Regarding the web tables, which are contained in web pages, we consider the following aspects: *page title* (HTML title), *page main passage* (article's short description), *page keywords* (most frequent words in the article), *table caption, table headers, table body*. The first three ones describe information around the table (i.e., surrounding text) and have been widely used in table retrieval approaches [8, 26, 32, 50, 54, 56]. The header indicates the properties of the column and helps to describe its meaning. The body (i.e., cells) contains all the table content, and the caption describes the subject of the table. Both article and table aspects are represented by words in natural language, and

[7] The BERT architecture has over 340 million parameters.

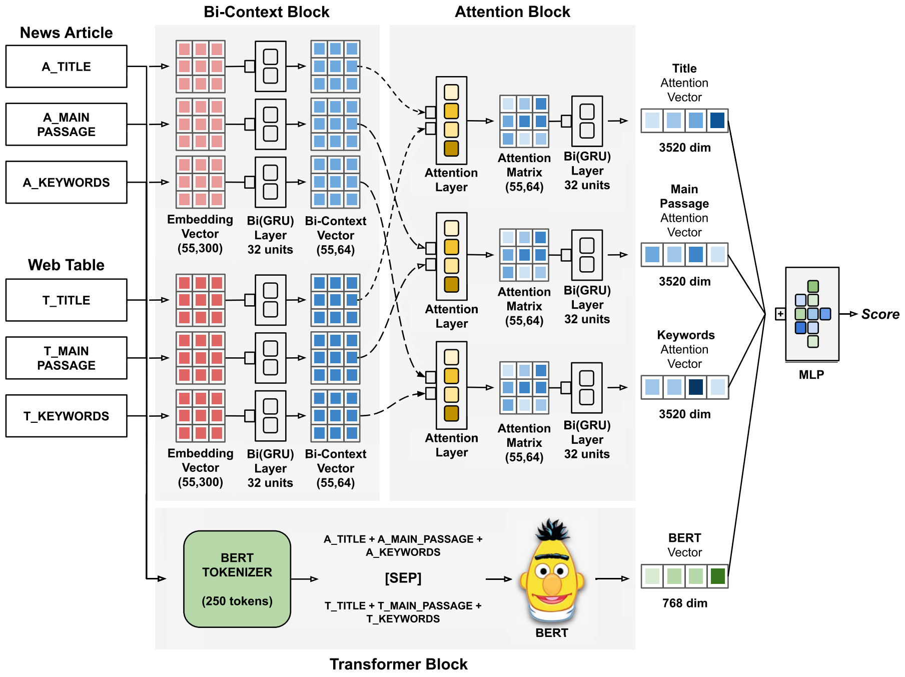


Fig. 2 An overview of the News-Table Matching Model. Our model learns a joint-representation for a ⟨*news*, *table*⟩ tuple by applying three network blocks: Bi-Context, Attention and Transformer, in which *embedding vector* is the FastText representation from a pre-trained corpus, *bi-context vector* is the contextual vectors learned from the input data, *attention matrix* is the matching degree between news and table aspects and *attention vectors* is the final matching signals for each pair of inputs. Moreover, An MLP architecture captures relevant matches and produces the similarity score on its top layer

some table cells can contain numerical values. Lastly, like previous studies [2, 41, 45, 48], we also focus on Wikipedia tables since they are rich corpus of relational information.

4 News-table matching model

In this section, we introduce the *News-Table* matching model. The core of our contribution is a novel cross-encoder model for this task that performs full attention over the input pair by combining RNN, attention layers and BERT architecture. As a result, we introduce a new way of applying existing neural blocks in the context of news-table matching. In contrast to Lees et al. [23], which uses BERT as a bi-encoder method, our model learns a joint-representation from a ⟨*news*, *table*⟩ tuple and predicts a similarity score to rank the tables. Our ablation study demonstrates that we obtain better results for table retrieval by merging these blocks in a single network. (see results in Table 8). We present the proposed model in Fig. 2. It produces two types of representations of the ⟨*news*, *table*⟩ input: one based on cross-attention and another based on BERT. Our goal is to capture relevant matching signals from both sides of the input by using different attention mechanisms. For the attention branch, the input is the embeddings of the words present in the news and table aspects. The network then applies a bidirectional recurrent network (bi-context block) on them to produce contextual vectors.

These vectors are passed to the attention block that combines the aspects of the news article and the table, and outputs an attention vector for each one of them. In the other network's branch, we utilize BERT to obtain another type of contextual representation based on self-attention from the $\langle news, table \rangle$ input. The outputs of two branches are concatenated and passed to an MLP, which computes the matching score on its top layer. The whole network is trained using backpropagation and a binary cross-entropy loss function. We provide details of the model in the remaining of this section.

4.1 Input data representation

As aforementioned, the model's input is a $\langle news, table \rangle$ tuple. From the news side, we consider the aspects: $article\_title$, $article\_main\_passage$ and $article\_keywords$. Regarding the table, unlike traditional table retrieval where the answer is inside the table (i.e., headers, caption and body), the most relevant information for *News-Table* matching are those aspects around the table (i.e., its surrounding text): $table\_page\_title$, $table\_page\_main\_passage$ and $table\_page\_keywords$. We verify this by running previous experiments using a standard retrieval approach whose the table content obtained the worst results for this task (see results in Table 3). Lastly, we represent each of those aspects as a sequence of words and utilize a word embedding approach to get word vectors, similar to previous neural IR studies [10, 18, 31]. The goal here is to generate a dense representation for each news/table token (*Embedding Vector*).

4.2 Bidirectional context

In the context of our task, Recurrent Neural Networks (RNN) have been applied both in IR tasks as well as in table retrieval approaches [46, 51]. Overall, these neural models learn contextual information from the sequential data. Since both sides of our input contain consecutive tokens (i.e., article and table aspects are defined by a sequence of words in natural language), we apply RNN to both to get their semantic connections. The goal here is to produce a new representation for the initial embeddings based on the context of the words. In our network, we apply a Gated Recurrent Unit (GRU) [9] to map each word to a fixed-length vector.[8] As a result, our model learns long text dependencies from each news/table input. Lastly, we also consider a bidirectional GRU to obtain the representation of each word from both directions, and use the concatenation of each hidden state as the final word representation, i.e., the fixed-length vector $= [\overrightarrow{h_{t-1}}, \overleftarrow{h_{t-1}}]$. By doing this, our network learns bidirectional contextual information of each news/table aspect (*Bi-Context Vector*), in order to be sensitive to word order such as reversing or shuffling.

4.3 Attention

The core challenge of this task is how to compute the similarity degree between distinct news/table features. Overall, news stories are described by several aspects including title, main passage, headlines, keywords and so on. Our intuition is that we can get relevant matching signals from both sides of the inputs by using different attention mechanisms in our network architecture. Based on that, inspired by previous approaches that try to capture relevant association between query/document [18, 24, 28, 31], our model applies attention

[8] We also try Long Short Term Memory (LSTM) for this step but GRUs achieved better results.

networks to learn interaction features between article and table, i.e., we use a cross-attention methodology for this network block. Attention mechanisms have been applied in many similar tasks including question answering and text matching [53, 58]. In the context of our task, a matching model needs to identify significant correlation between table aspects and news features such as title, main passage and keywords, in order to capture the best matching information. For that, we use scaled dot-product attention to weigh news aspects based on table aspects [49]. Specifically, we compute the attention between co-related aspects: title, main passage and keywords, as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{1}$$

where Q corresponds to the query, K is the key, V the value and $softmax$ is a normalization function. For the title aspect, for instance, we consider $article\_title$ as Q and V, and $table\_page\_title$ as K. By doing this, we create a novel representation for $article\_title$ weighted by $table\_page\_title$ (*Attention Matrix*). The other aspects go similarly. In summary, for each pair of them, we generate one attention matrix which represents the matching degree between article and table features. Lastly, the attention matrices fed a Bi-GRU, whose output is flattened, producing the final bi-vectors (*Attention Vectors*).

4.4 Transformer

In addition to the attention block, which computes relevant matching for correlated attributes (e.g., article title and table title), we also employ BERT to capture significant interactions between them.[9] BERT is a novel bidirectional sentence encoder based on transformer blocks and multi-head attention mechanisms [44]. It contains multiple attention heads attending to distinct parts of the sequence at the same time (e.g., longer-term dependencies versus shorter ones). As a result, BERT produces a different representation for each word according to the text sequence where it appears. For example, if we consider the news story in Fig. 1, the word *Vinci* has a vector representation when it appears in the news title and another one when it is in the news description since its neighboring words are distinct. Based on that, similar to a previous study that employs BERT for the table retrieval [8], we apply it to create a contextual vector representation of the ⟨*news*, *table*⟩ pair. To fine-tune BERT, we use the task of sentence pair classification, i.e., a pair of sentences is classified as match or non-match. For that, we assume the news-table aspects as a single-field (text) document and apply BERT tokenizer to generate $input\_ids$, $attention\_masks$ and $token\_types$. In addition, the token $[SEP]$ separates news-segments from table-segments. Lastly, we adopt the final hidden state h of the first token $[CLS]$ as the whole ⟨*news*, *table*⟩ representation, similar to Chen et al. [8] (*BERT Vector*).

4.5 MLP

On the top of our network, the attention vectors produced by the attention block are concatenated with the BERT vector and fed a Multi-Layer Perceptron Architecture (MLP) to learn matching features and predict a similarity score. The goal here is to capture relevant match signals from both learned vectors. Moreover, the *News-Table* matching score is generated by

[9] Our ablation study shows we can improve the model performance by joining such two attention methodologies (see results in Table 8).

a sigmoid function over the last MLP neuron. We use this score to rank the tables. Lastly, we use the binary cross-entropy loss function for training the whole network according to Equation 2.

$$Loss = \frac{1}{n}\sum_{i=1}^{n} y_i \ . \ log \ \hat{y}_i + (1 - y_i) \ . \ log(1 - \ \hat{y}_i) \tag{2}$$

where $\hat{y}_i$ is the $i - th$ value in the model output, y_i is the corresponding target value, and n is the number of values in the model output.

5 Experimental setup

5.1 Baselines

As suggested in previous studies for table retrieval [42], we compare our solution to several state-of-the-art baselines including traditional IR methods, document/sentence encoders, neural IR models and dense passage retrieval strategies.

Traditional IR methods One can think of the news article as a (long) keyword query and therefore apply traditional IR techniques or QA solutions. We use two different strategies to represent the inputs: news article and Web table. In the multi-field approach, similar to Zhang and Balog [56], we compute the similarity score over each aspect separately (e.g., *news title* and *table page title*). Then, we rank the tables based on the mean of similarity/score from all aspects. In the single-field strategy, following Cafarella et al.[4], we represent both news and table aspects as a single-field (text) document by concatenating their aspects.

- *Cos(TF-IDF)* [37]. It is a basic IR method that represents query/documents based on term-frequency and inverse document frequency. It ranks the tables based on cosine similarity over TF-IDF.
- *BM25* [35]. It is an IR algorithm based on the probabilistic relevance framework that uses term-frequency weighting and document length for ranking.

Document/sentence encoders We evaluate pre-trained sentence/document encoders to represent the news-table aspects. We consider both single and multi-field document approaches, and the cosine similarity is used to score and rank the tables.

- *Doc2Vec* [22]. It is an unsupervised approach that encodes sentences, paragraphs or documents by using neural networks, similar to Word2Vec.
- *USE* [5]. It is a transformer-based network that learns sentence embeddings by using attention. The model was pre-trained on similarity-related tasks such as textual entailment and question/answering.
- *Public-BERT* [11]. It is a sentence encoder that uses bidirectional transformer networks for language representations, pre-trained on a large corpus from the Web.
- *Fine-tuned BERT.* We fine-tune BERT to the task of sentence pair classification. For that, we concatenate both news and table aspects as a single-field text document.
- Lees et al. [23]. Given its reproducibility issues, whose both NewsBERT and their training data are not publicly available, we implement their network architecture over our news-table corpus. More specifically, we fine-tune BERT by using a pairwise hing loss function over cosine distance, whose goal is to learn that negative (article, table) pairs should have lower similarity than positive pairs.[10] The input for the model is a triple composed of a

[10] https://tinyurl.com/ranking-loss.

news article, a positive table and a negative table. They share the same set of parameters for the BERT-encoder.[11]

Neural IR approaches We also evaluate a set of widely used text match neural models for IR. These models have been applied for downstream tasks including question answering [51], paraphrase identification [31], ad-hoc search [52] and document retrieval [16]. We use the public Matchzoo Toolkit to train each model [17]. Specifically, we represent the news-table aspects as a single-field (text). Then, we use the matching score produced by the respective neural model to rank the tables.

- *DSSM* [19]. It maps query/document to a low-dimensional space by using TF-IDF, N-Grams and nonlinear layers, and the cosine distance is the query/document relevance.
- *ARCI* [18]. It is a siamese model that learns semantic representations by using 1D-convolution and max-pooling layers.
- *ARCII* [18]. Unlike previous model, it first builds an interaction space between two sentences and applies 2D-convolution and max-pooling layers to encode high-level representations.
- *MVLSTM* [51]. It applies LSTM layers to learn sentence representations, and a max-pooling interaction step extracts relevant match from both sentences.
- *DRMM* [16]. It extracts matches from sentences by word histogram, feed-forward and term gating networks.
- *MATCH PYRAMID* [31]. It is based on image recognition models which learns sentence similarities by using matching matrix, convolution and dynamic pooling.
- *KNRM* [52]. It uses an RBF kernel pooling to learn query-document features from a translation matrix.
- *CONV-KNRM* [10]. Unlike previous model, it generates a continuous vector for query/document terms by using word embeddings. Next, convolutional layers construct n-gram representations of the text. Last, a K-Gaussian kernel pooling counts soft-matches and generates the match score.
- *DUET* [28]. It is a deep learning model for matching queries and documents by using local and distributed representations of text.

Dense passage retrieval strategies We complete our baselines by considering novel dense passage retrieval methods. We generate dense representations for both news-table aspects (as a single-field (text) document). Then, we compute their similarity by applying cosine distance for SBERT and DistilBERT or dot-product for DPR.

- *Dense Passage Retrieval (DPR)* [20]. It is a two independent BERT-based encoder that employs dot-product similarity as a ranking function for retrieval.
- *Sentence BERT (SBERT)* [34]. It is a BERT-based model that uses siamese and triplet networks to derive semantically sentence embeddings that can be compared by utilizing cosine-similarity. We also evaluate SBERT by using DistilBERT: a smaller, faster, cheaper and lighter version of BERT [38].

5.2 Datasets

Train-validation data Since we are not aware of any public labeled data for this task, we implemented a distant supervision strategy to build a news-table matching dataset. For that,

[11] We try the following similarity thresholds for the cosine distance over positive and negative pairs: 0.3, 0.4, 0.5, 0.6, and 0.7, in which 0.3 achieves the best results in our validation dataset.

Table 1 A sample of news-table matching pairs in our training and validating corpus. We show the news-title, the Wikipedia page title for the table and the labels for each one, in which 1 means a matching pair and 0 is a non-match one

News title	Wikipedia page	Label
Folk Music Awards Nominees Announced	Canadian Folk Music Awards	1
Patty Andrews, Leader Of The Andrews Sisters, Dies	Andrews Sisters	1
Facts About Mexico's Education System	Education in Mexico	1
One Love by David Guetta Reviews	(I'll Never Be) Maria Magdalena	0
The Prime Minister's Official Hub	Megan Hilty	0
Fire Interviews Fraser	Filmfare Award for Best Actress	0

we leverage the links in the reference section on the Wikipedia page that contains the tables. Specifically, we selected only reference links belonging to news web sites. Here, we assume that those news articles are likely related to the table content as they are collocated in the same Wikipedia page. For this evaluation, we gathered 275,352 news articles and 298,792 web tables by adopting Newspaper API.[12] To generate the matching pairs, we index the tables by using a multi-field approach using the Elastic Search (ES) API.[13] Then, for each article, we search over the index by considering its aspects as a single-field query. Lastly, we consider the table with the highest cosine similarity over TF-IDF as the match. By doing this, we collected 93,818 news-table matching pairs that we use for evaluating the proposed model as well as the neural IR approaches. We split this corpus into 84,436 (90%) examples for training and 9382 (10%) for validating. To the best of our knowledge, this is the first public dataset for the task of matching news articles and web tables in literature. Finally, to demonstrate the reliability of our data, we manually check 100 random samples of matching and non-matching pairs in our train/validation dataset. This evaluation showed that 92% of them contain correct labels. We illustrate some of them in Table 1.

Test data We use as a test set a dataset released by Lees et al. [23]. The authors use an internal production crowd evaluation platform to construct a public dataset comprising 148 news-table pairs created by human labelers. Each pair is labeled by 3 to 5 labelers, and labels are obtained from the majority of ratings. The relevance judgment determines the quality of tables paired to news articles and whether the table provides additional context for, or insight into, the article. In addition to answer whether the table is relevant to the article ("Yes", "So-so", "No"), a question about the table's level of clarity is used to ensure high enough table quality to enable assessment of relevance. Finally, to simulate a real scenario of table retrieval, the authors also released a table corpus containing 53K Wikipedia tables. We use both the table corpus and the ground-truth news-table pairs in our experimental evaluation.

Data preprocessing We removed pages with empty values for *title*, *main passage* or *keywords*, and also special characters and stopwords from the text. In addition, we padded long/short sentences based on the average of tokens for each aspect. Table 2 shows a statistical analysis of them on the training, validation and test datasets. As one can see, the aspects have at least 37 tokens on average from the news side (i.e., by jointing news title, description and keywords), which limits the application of QA table retrieval solutions that usually assume few words in the query.

[12] https://newspaper.readthedocs.io/en/latest/.

[13] https://www.elastic.co.

Table 2 A statistical overview for each news-table aspect over the train, validation and test dataset. We point out the minimum and maximum values of the tokens, average of words and standard deviation for each of them

Aspects	Train/Validation data				Test data			
	Min	Max	Mean	STD	Min	Max	Mean	STD
News Title	2.0	28.0	6.6	2.7	2.0	23.0	6.6	2.9
News Main Passage	2.0	931.0	18.0	22.0	2.0	327.0	18.8	18.4
News Keywords	1.0	22.0	13.0	2.0	6.0	20.0	13.4	2.1
Table Title	1.0	13.0	2.9	1.3	1.0	13.0	3.7	1.4
Table Main Passage	1.0	674.0	78.4	40.7	1.0	756.0	68.1	63.9
Table Keywords	1.0	18.0	10.4	1.5	1.0	20.0	11.3	2.3

5.3 Methodology

Following previous work [41–43, 46], we assume there is a pool of initial candidate tables for re-raking. Hence, similar to Shraga et al. [42], we index the table corpus on Elastic Search API by using a multi-field document approach. Moreover, for each news article, we use ES to obtain the top $k = 100$ candidates tables with the highest BM25 score (by using its default parameter setup). The recall at $k = 100$ using this strategy is 0.9122, which means that for 8.78% of the news articles, the match table is not present in the 100 table candidates. Finally, we re-rank the candidate tables by applying each baseline as well as our proposed model.

5.4 Evaluation measures

We evaluate our model and baselines by considering a table retrieval re-ranking task. Like previous IR work [10, 51], we employ Mean Reciprocal Rank (MRR) at cutoff $k = 50$ to assess the average position in which a correctly table-answer appears in the ranking, where the first positions are preferred and, therefore, receive higher scores. In addition, we use accuracy@k (a.k.a. top-k accuracy) at cutoffs $k \in \{1, 5, 10, 20, 50\}$ to measure the percentage of news articles in the test set correctly matched to at least one of the top-k ranked tables. Accuracy@k is a metric widely used in the evaluation of question answering tasks [39] and recommendation systems [27], for which, like for our problem, there are one or few relevant results for each query. We also run a paired Wilcoxon test at 95% confidence level to measure the results' significance. Finally, regarding prediction/latency time of each model, we measure the runtime to retrieve the top-100 candidate tables, produce the similarity scores and obtain the top-20 matching tables for each article. Based on that, we compute the average runtime per article in the test set. We replicate this experiment ten times for each baseline as well as for the proposed model and report the mean.

5.5 Implementations details

We implement the proposed model by using Python 3.6 and TensorFlow 2.2.0. To encode the *news-table* aspects, we use a FastText corpus with 1 million word vectors trained on English Wikipedia pages.[14] Regarding the IR methods, we use TfidfVectorizer from Sklearn for TF-

[14] https://fasttext.cc/.

IDF,[15] and Rank-BM25 API for BM25.[16] In relation to the document/sentence encoders, we consider a pre-trained Gensim model for DOC2VEC.[17] Moreover, we import the 4th version of Universal Sentence Encoder (USE) from TensorFlowHUB.[18] For public BERT, we adopt a online version of Bert-as-Service,[19] and consider TFBertModel from Hugging Face to the fine-turning task.[20] Concerning the neural IR models, we use the Matchzoo Toolkit to train each of them.[21] For dense passage retrieval methods, we use Sentence Transformer API.[22] Finally, we perform the experiments by using a Titan XP GPU and Ubuntu 16.04 LTS.

6 Results

Top-k Algorithm The objective of the top-k algorithm is to retrieve the highest number of relevant tables for the matching model (re-ranking). Based on that, we ran a set of distinct index settings to investigate which table aspects obtain the best candidate set, and also evaluate different news features as input queries.[23] We then apply Elastic Search API and BM25 algorithm for retrieval. Such approach has been widely used as a strong baseline for several open-domain retrieval tasks [20]. Table 3 shows the results for each index field and news aspects in terms of $Acc@100$.

Unlike table retrieval for QA, in which the table-content can improve the similarity degree since most answers are inside the table, for news-table matching, the most relevant tables are found by matching the text around the table instead of its content. For example, by combining all news aspects as the input query and the surrounding text of the table as indexes - *page title*, *page main passage* and *page keywords* - the pool of candidates contains over 90% of the ground-truth tables in the set ($Acc@100 = 0.9122$). In contrast, when we use table aspects as indexes like headers or caption, it achieves the lowest results for the same metric. *Table caption* and *table headers* achieve the worst values for $Acc@100$ (only 30% of the matching tables are in the candidate set). Therefore, we do not include these aspects as inputs for the proposed model.

Finally, we adopt the best combination of them as the top-k algorithm (*line 20*) and retrieve a pool of candidate tables by querying over the index at cutoff $k = 100$. We use this subset to evaluate both baselines and the proposed method over the ranking step as follows.

Ranking results We now discuss the core results of our study, i.e., the ranking step, which we present in Table 4. We first examine the ranking accuracy of the approaches then we focus on their average prediction time. Given the pool of candidate tables, we re-rank them by applying each baseline as well as our proposed model. Overall, our model outperforms all baselines for all evaluation metrics. In fact, we run a paired Wilcoxon Test at 95% confidence level between the MRR score of our model and each baseline which showed that its MRR value is statistically different than all baselines (See Table 5 for the p-values comparisons).

[15] https://scikit-learn.org.

[16] https://pypi.org/project/rank-bm25.

[17] https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html.

[18] https://tfhub.dev/google/universal-sentence-encoder/4.

[19] https://github.com/hanxiao/bert-as-service.

[20] https://huggingface.co/transformers/model_doc/bert.html.

[21] https://github.com/NTMC-Community/MatchZoo.

[22] https://www.sbert.net/docs/pretrained-models/dpr.html.

[23] We do not combine the table-content and its surrounding text since the table-aspects achieve the worst results for the evaluated metric.

Table 3 Searching over the index by using Elastic Search API and BM25 algorithm (Top-k Algorithm). We evaluate distinct approaches for news-table aspects. Bold values represent the best combination for the candidate set in terms of $Acc@100$, and the symbol $(*)$ points out the worst results for the same metric (when we use table aspects as indexes)

#	News aspects(Query)	Table aspects(Indexes)	Acc@100
1	Title	Title	0.6959
2	Title	Main Passage	0.6622
3	Title	Keywords	0.7279
4	Title	*Table Caption**	0.0680
5	Title	*Table Headers**	0.1216
6	Title	*Table Body**	0.3176
7	Main Passage	Title	0.5682
8	Main Passage	Main Passage	0.5000
9	Main Passage	Keywords	0.5676
10	Main Passage	*Table Caption**	0.0621
11	Main Passage	*Table Headers**	0.1284
12	Main Passage	*Table Body**	0.3041
13	Keywords	Title	0.7343
14	Keywords	Main Passage	0.6892
15	Keywords	Keywords	0.7162
16	Keywords	*Table Caption**	0.0816
17	Keywords	*Table Headers**	0.2905
18	Keywords	*Table Body**	0.5608
19	Title, Main Passage	Title, Main Passage	0.7838
20	**Title, Main Passage, Keywords**	**Title, Main Passage, Keywords**	**0.9122**

In terms of $ACC@1$, our model correctly ranks over 55% of the ground-truth tables, and at the top-five ranking positions ($ACC@5$), it achieves accuracy of 77%. Comparing its results, for instance, with *multi-field USE* (a pre-trained matching encoder), our model is at least 36% more effective in the re-ranking step. Regarding the *Fine-tuned BERT* (the strongest baseline), our model surpass it by over 13%. In contrast to Lees et al.[23], our model surpasses their network architecture by a large margin for all evaluated metrics. In fact, previous studies have demonstrated that BERT bi-encoders, used by Lees et al.[23], usually have lower performance in comparison with BERT cross-encoders [47], which we apply in our solution. Another possible reason for their poor performance is that in their results NewsBERT attains over 45% of the matching tables for $ACC@1$. But, since NewsBERT is a private Google resource, we implemented their approach using a public BERT. Lastly, concerning the dense passage retrieval methods such as SBERT and DPR, our solution outperforms them by over 20% in terms of $MRR@50$.

We next analyze the performance of the single and multi-field baselines, neural IR models as well as the DPR approaches. Similar to traditional table retrieval strategies, news-table ranking results are improved by adopting a multi-field methodology. For example, $ACC@1$ improves from 0.1892 to 0.3041 (*DOC2VEC*), 0.2838 to 0.4054 (*USE*) and 0.2230 to 0.2432 (*PUBLIC-BERT*) when a multi-field approach is used. Indeed, each news or table aspect individually contributes to the ranking score. The neural IR models achieve the worst results: their $ACC@1$ is close to zero. Even when combining other news-table aspects, the neural

Table 4 Ranking results for the proposed model and baselines. News aspects: *Title*, *MainPassage*, *Keywords*. Table aspects: *Page Title*, *Page MainPassage*, *Page Keywords*. The symbol (*) means a statistically significant better result compared to all baselines, (SF) is a Single-Field approach and (MF) is a Multi-Field approach. We also show the average prediction time in seconds for each news article in the test set

APPROACH	MRR@50	Acc@k=1	5	10	20	Avg. time(s)
Cos(TF-IDF) - (SF)	0.4414	0.3176	0.5946	0.7297	0.7973	0.2777
BM25 - (SF)	0.4269	0.2973	0.5946	0.6959	0.7432	0.0456
DOC2VEC - (SF)	0.2734	0.1892	0.3649	0.4662	0.6284	1.2770
USE - (SF)	0.3837	0.2838	0.4932	0.5811	0.6622	0.4078
PUBLIC-BERT - (SF)	0.3313	0.2230	0.4459	0.5541	0.6824	3.1564
Cos(TF-IDF) - (MF)	0.4513	0.3649	0.5405	0.6081	0.7365	0.7624
BM25 - (MF)	0.4069	0.3041	0.5405	0.6351	0.6824	0.0594
DOC2VEC - (MF)	0.4248	0.3041	0.5203	0.6622	0.7500	1.3787
USE - (MF)	0.5166	0.4054	0.6419	0.7297	0.8108	0.4634
PUBLIC-BERT - (MF)	0.3373	0.2432	0.4459	0.5068	0.5811	8.4924
Fine-tuned BERT	0.5949	0.4865	0.7500	0.7905	0.8514	1.9803
Lees et al. [23]	0.1510	0.0743	0.1824	0.2905	0.5203	1.5733
DSSM	0.0348	0.0068	0.0338	0.0878	0.1959	0.2937
ARCI	0.0768	0.0270	0.0878	0.1554	0.3378	0.3044
ARCII	0.0667	0.0135	0.0946	0.1486	0.3311	0.6722
MVLSTM	0.0765	0.0068	0.1216	0.2432	0.4122	0.3518
DRMM	0.0486	0.0270	0.0473	0.1081	0.2365	6.5425
MATCH PYRAMID	0.0545	0.0068	0.0676	0.0878	0.1689	0.4956
KNRM	0.1285	0.0608	0.1689	0.2703	0.4392	0.4032
CONV-KNRM	0.1103	0.0608	0.1486	0.1959	0.2703	0.5717
DUET	0.1560	0.0608	0.2365	0.3784	0.5405	0.5013
DPR	0.4550	0.3311	0.6419	0.7568	0.8243	1.1242
SBERT	0.5135	0.3851	0.6824	0.7770	0.8514	0.3054
DistilBERT	0.5045	0.3986	0.6216	0.6824	0.8041	0.5081
OUR-METHOD	0.6369*	0.5541	0.7703	0.8176	0.8514	2.2760

Table 5 P-values results for Wilcoxon Test. We compare our model against each baseline in terms of $MRR@50$. Our approach statistically outperforms all baselines

Ranking method	P-value
Cos(TF-IDF) - single-field	$1.5329 * 10^{-08}$
BM25 - single-field	$1.8467 * 10^{-08}$
DOC2VEC - single-field	$6.7410 * 10^{-15}$
USE - single-field	$1.0348 * 10^{-09}$
PUBLIC-BERT - single-field	$3.5424 * 10^{-13}$
Cos(TF-IDF) - multi-field	$1.0992 * 10^{-07}$
BM25 - multi-field	$2.2055 * 10^{-09}$
DOC2VEC - multi-field	$5.0557 * 10^{-08}$
USE - multi-field	0.0001
PUBLIC-BERT - multi-field	$3.0637 * 10^{-14}$
Fine-tuned BERT	0.0077

Table 5 continued

Ranking method	P-value
Lees et al. [23]	$1.4215 * 10^{-19}$
DSSM	$1.2386 * 10^{-23}$
ARCI	$2.2968 * 10^{-21}$
ARCII	$1.8398 * 10^{-22}$
MVLSTM	$4.9307 * 10^{-22}$
DRMM	$4.8761 * 10^{-22}$
MATCH PYRAMID	$1.0299 * 10^{-22}$
KNRM	$8.2976 * 10^{-20}$
CONV-KNRM	$1.4310 * 10^{-20}$
DUET	$1.1611 * 10^{-19}$
DPR	$9.5979 * 10^{-7}$
SBERT	0.0002
DistilBERT	$4.0489 * 10^{-05}$

IR models do not outperform any other ranking method. A possible reason for the poor performance of these models is they were devised to answer short queries and, in the context of our task, most of the queries are long, i.e., the concatenation of the news aspects. Among the neural IR models, DUET obtains the best performance: $ACC@5 = 0.2365$. Finally, concerning dense passage retrieval models, *SBERT* and *DistilBERT* surpass *DPR* in terms of $Acc@1$ and $MRR@50$. Moreover, our study also confirms that these approaches attain better results for retrieval than traditional IR methods like Cos(TF-IDF) and BM25. For example, *SBERT* supasses *Cos(TF-IDF) - (MF)* by over 12% in terms of $MRR@50$.

On the whole, *USE* and *Fine-tuned BERT* are the strongest baseline as they achieve better results than the traditional IR methods, neural models and sentence encoders in terms of $ACC@1$ and MRR. In fact, dense retrieval approaches, which apply BERT as encoder, have shown comprehensive efficacy at several open-domain IR tasks [20]. Our results also confirm such hypothesizes for news-table matching. In addition, the IR methods, although simpler, achieve good results and are strong baselines. For example, in $ACC@1$, *Cos(TF-IDF) multi-field* surpasses both all neural IR models and sentence encoders such as *DOC2VEC*.

Lastly, similar to previous work Shraga et al. [43], we assume there is a pool of candidate tables in which our model applies re-ranking. As aforementioned, in this evaluation, this candidate pool with $k = 100$ has recall of 0.9122. To evaluate our approach in a 100% recall scenario, we added the correct table to the pools that do not contain it (8.78% of the news articles). Table 6 shows such results. Also in this scenario, our method outperforms the strongest baselines in terms of $ACC@1$ and MRR. In addition, even when we change the the top-k retrieval algorithm to collect the candidate pool (i.e., BM25 to classic TF-IDF), the proposed model outperforms the baselines. That result confirms our method correctly re-ranking the web tables regardless of the retrieval approach. Finally, over a maximum recall scenario, our method ranks 62.16% of the ground-truth tables at the first ranking position using Cos(TF-IDF) as the top-k algorithm.

Prediction time We now discuss the query prediction time of each ranking model. We measure the average runtime per news article in the test dataset as shown in Table 4 (over the last column). Overall, the algorithm *BM25* for both single and multi-field approaches has the

Table 6 News-table matching results by adopting a maximum recall scenario. The symbol (*) means a statistically significant better result compared to the other baselines

Top-K Algorithm	RE-RANK	MRR@50	Acc@k=1	5	10
BM25	Cos(TF-IDF) - (MF)	0.4530	0.3649	0.5473	0.6149
Cos(TF-IDF)	Cos(TF-IDF) - (MF)	0.4778	0.3919	0.5541	0.6351
BM25	USE - (MF)	0.5345	0.4122	0.6757	0.7703
Cos(TF-IDF)	USE - (MF)	0.4416	0.3176	0.5676	0.6892
BM25	OUR METHOD	0.6488*	0.5608	0.7838	0.8514
Cos(TF-IDF)	OUR METHOD	0.6959*	0.6216	0.8176	0.8649

smallest time, 0.0456 and 0.0594 seconds per query respectively, while the models *DRMM* and *PUBLIC-BERT - (MF)* have the longest ones (6.5425 and 8.4924 seconds respectively). In contrast, *PUBLIC-BERT* uses external web services which leads to higher latency. For *DRMM*, the model combines several components including matching histogram mapping, feed forward networks, terms gatting networks and term vector frequencies. As a result, it has a high runtime. Regarding single and multi-field approaches, the time per query is very similar. For example, *BM25*, *DOC2VEC* and *USE* have similar runtimes. In relation to the neural IR models such as *DUET* and *CONV-KNRM*, their execution time per query is close to 0.5 seconds, but they have poor performance in terms of accuracy. For the novel dense passage retrieval techniques, *DPR* is the slowest algorithm (1.1242 seconds). *SBERT* is over three times faster than *DPR*.

Lastly, our cross-encoder model has a prediction time of 2.2760 seconds per news article. Comparing its results, for instance, with *Fine-tuned BERT* (the strongest baseline), our model is over 0.2 seconds slower but over 13% more effective in terms of $ACC@1$. Such results also indicate that the combination of blocks, used in our network, does not put high latency on query prediction. *Bi-GRUs* and *attention layers* increase over 0.2 seconds in the final time compared to the *Fine-tuned BERT*, which only uses the BERT architecture. In contrast to Lees et al. [23], bi-encoder models are faster than ours but have lower accuracy [24]. Finally, a possible alternative to decrease the runtime of our model is to use a distilled version of BERT instead the original one in the *Transformer Block* since it shows a much smaller runtime.

Matching analysis We now present three matching examples for the test set. For each news story, we collect the $top - 5$ tables produced by our model, which we illustrate in Table 7 (note we also show their similarity degree). This results demonstrate our model re-ranks correlated tables for each of them in the first ranking positions. For example, regarding Article 1, which contains facts about *Cars*, *America*, *Chrysler* and *Ford*, our model points out tables such as *Chrysler Vehicles* and *Ford Vehicles*. In fact, a reader of this news may be interested in knowing which cars are manufactured by the Chrysler/Ford automakers. In addition, our approach also finds matching tables in which there is no term-overlap for the news-table titles, i.e., exacting matching (e.g., *Automobiles Manufactured in United States*). Such linking provides further information about the central topic of the story - *cars made in America* - beyond exploring specific places for this news as the United States and Ontario. Regarding the second article, *NASA's Moonwalking Apollo Astronauts*, the results are similar since our model finds tables like Apollo Missions or Astronauts, Spacewalks and Moonwalks (very relevant tables to this news). As a result, any reader can further explore the list of all

[24] The training time for cross and bi-encoder BERT-based models are similar as both of them are composed of the BERT architecture. In our experiments, their fine-tuning time is over 20min per epoch.

Table 7 A sample of news-table matching in the test set. We present the $top - 5$ tables for three evaluated news articles, beyond pointing out their similarity degree

Article 1 - Title: *Cars made in America? Chrysler, Ford no longer qualify*.

#	Table Title	Similarity
1	Chrysler Vehicles	0.9861
2	Ford Vehicles	0.9104
3	Toyota Vehicles	0.8727
4	Automobiles Manufactured in United States	0.8223
5	Automobiles Manufactured in Ontario	0.8023

Article 2 - Title: *NASA's Moonwalking Apollo Astronauts*.

#	Table Title	Similarity
1	Apollo Astronauts	0.9976
2	Apollo Missions	0.9910
3	Missions of the Moon	0.8966
4	Spacewalks and Moonwalks	0.8891
5	Spacewalkers	0.8875

Article 3 - Title: *The Best-Selling Video Games*.

#	Table Title	Similarity
1	Best Selling Video Games	0.9476
2	Best Selling Nintendo Video Games	0.9130
3	Best Selling Gamecube Video Games	0.9106
4	Games Gold Games	0.9068
5	The Simpsons Couch Gags	0.9022

Table 8 Ablation study of the proposed model. We evaluate the following components and their combinations: Bi-Context Block, Attention Block and Transformer Block

#	*Network block*	*MRR@50*	Acc@k=1	5	10
1	Bi-Context	0.0913	0.0270	0.1419	0.2568
2	Attention	0.1647	0.0811	0.1959	0.3176
3	Transformer	0.5949	0.4865	0.7500	0.7905
4	Bi-Context + Attention	0.3768	0.3108	0.4324	0.5405
5	Bi-Context + Transformer	0.6236	0.5270	0.7432	0.8108
6	Attention + Transformer	0.6193	0.5135	0.7703	0.8311
7	Full Model	0.6369	0.5541	0.7703	0.8176

Apollo missions or astronauts which landed on the Moon. Lastly, for Article 3, which relates to Best-Selling Video Games, our approach retrieves matching tables such as *Nintendo* and *Gamecube Video Games*, i.e., specific brands for games.

Ablation Study We conclude this section by presenting an ablation study of our model. As we show in Fig. 2, our network combines three main components: *Bi-Context Block*, which uses recurrent networks to learn contextual vectors from the input; *Attention Block*, whose goal is to compute the matching degree between article and table features; and *Transformer Block*, which applies multi-head attention layers based on BERT architecture. We evaluate each block individually as well as their combinations. Table 8 shows the results for each of them in terms of $Acc@k$ and $MRR@50$. Overall, if we only use *Bi-Context Block* (*line 1*) or *Attention Block* (*line 2*) for matching, the model achieves the worst results for this task (its $Acc@1$ is equal to 0.0270 and 0.0811 respectively). In contrast, by combining recurrent networks and attention layers (*line 4*), the model finds over 30% of the matching tables at the first rank position (almost four times better than these isolated blocks). Specifically, the *Transformer Block* achieves the best results for $MRR@50$ and $Acc@k$ compared to the other network components (*line 3*). Moreover, if we concatenate it with both recurrent networks or attention layers (*lines 5* and *6*), the results also increase for the same metric. For example, its results improve from 0.4865 to 0.5270 for $Acc@1$ (*line 5*). The results are similar to the combination of *Attention* and *Transformer*. Finally, by analyzing all blocks and their combinations, the *Full Model* attains the highest results for news-table matching (over 55% for $Acc@1$). Such results indicate our approach increases the performance by over 13% in terms of $Acc@1$ compared to the *Transformer Block* (the most isolated baseline). Moreover, we also confirm that recurrent networks and cross-attention layers can capture relevant match signals for news-table matching.

7 Concluding remarks

Matching news articles and web tables is a recent table retrieval problem. In this paper, we claimed news understanding can be enhanced by joining associated content from structured web tables. In fact, previous studies have demonstrated that online readers also explore tables inside Wikipedia pages after looking at news articles. Based on that, we focused on the task of the *news-table* matching. Our solution for that is a hybrid neural network that combines different encoders to better represent articles and tables for this task. Our intuition was that we can improve the similarity degree by using distinct attention approaches in the same network architecture. We performed an extensive evaluation that assessed the performance of our approach, comparing it with standard IR methods, document/sentence encoders and neural IR models. In comparison to Lees et al. [23], the most related baseline, our study provides further directions in the context of *News-Table* matching. The overall results point out our model outperforms the baselines for all evaluated metrics. As future work, we plan: (i) To explore the improvement of the news understanding brought by the top-ranked web tables; (ii) To estimate the number of news articles that may not be able to match off-the-shelf web tables; (iii) To add more semantic features into the model such as entities and categories and (iv) To explore the news-table matching problem in the context of fake news verification/checking by conducting novel experiments.

Acknowledgements Not Applicable.

Author Contributions These authors contributed equally to this work.

Funding This work is supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) from Brazil.

Availability of data and materials The datasets and source code are public available in: https://github.com/levysouza/News-Table-Matching.

Declarations

Conflict of interest The authors declare they have no conflict of interest.

Ethical Approval and Consent to participate Not Applicable.

Consent for publication The authors agree to the publication of this study.

References

1. Agarwal S, Singh NK, Meel P (2018) Single-document summarization using sentence embeddings and k-means clustering. In: Proceedings of the 2018 international conference on advances in computing, communication control and networking, IEEE, pp 162–165
2. Bhagavatula CS, Noraset T, Downey D (2013) Methods for exploring and mining tables on wikipedia. In: Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics. ACM, pp 18–26
3. Cafarella MJ, Halevy AY, Wang DZ et al (2008) Webtables: exploring the power of tables on the web. Proc VLDB Endow 1(1):538–549
4. Cafarella MJ, Halevy AY, Lee H et al (2018) Ten years of webtables. Proc VLDB Endow 11(12):2140–2149
5. Cer D, Yang Y, Kong SY, et al (2018) Universal sentence encoder. CoRR abs/1803.11175
6. Chakrabarti K, Chen Z, Shakeri S, et al (2020) Tableqna: Answering list intent queries with web tables. CoRR abs/2001.04828
7. Chen X, Cheng Y, Wang S, et al (2021) Earlybert: Efficient BERT training via early-bird lottery tickets. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing. Association for Computational Linguistics, Virtual Event, pp 2195–2207
8. Chen Z, Trabelsi M, Heflin J, et al (2020) Chen Z, Trabelsi M, Heflin J, et al (2020) Table search using a deep contextualized language model. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. ACM, Virtual Event, China, pp 589–598
9. Chung J, Gülçehre Ç, Cho K, et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR abs/1412.3555
10. Dai Z, Xiong C, Callan J, et al (2018) Convolutional neural networks for soft-matching n-grams in ad-hoc search. In: Proceedings of the 11th ACM international conference on web search and data mining. ACM, Marina Del Rey, USA, pp 126–134
11. Devlin J, Chang MW, Lee K, et al (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies. Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186
12. Gavrilov D, Kalaidin P, Malykh V (2019) Self-attentive model for headline generation. In: Proceedings of the 41st European conference on information retrieval research, Lecture Notes in Computer Science, vol 11438. Springer, Cologne, Germany, pp 87–93
13. Glass M, Canim M, Gliozzo A, et al (2021) Capturing row and column semantics in transformer based question answering over tables. CoRR abs/2104.08303
14. Govindaraju V, Zhang C, Ré C (2013) Understanding tables in context using standard NLP toolkits. In: Proceedings of the 51st annual meeting of the association for computational linguistics. the association for computer linguistics, Sofia, Bulgaria, pp 658–664
15. Gu X, Mao Y, Han J, et al (2020) Generating representative headlines for news stories. In: Proceedings of the web conference 2020. ACM / IW3C2, Taipei, Taiwan, pp 1773–1784
16. Guo J, Fan Y, Ai Q, et al (2016) A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM international conference on information and knowledge management. ACM, Indianapolis, USA, pp 55–64
17. Guo J, Fan Y, Ji X, et al (2019) Matchzoo: A learning, practicing, and developing system for neural text matching. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. ACM, Paris,France, pp 1297–1300
18. Hu B, Lu Z, Li H, et al (2015) Convolutional neural network architectures for matching natural language sentences. CoRR abs/1503.03244

19. Huang P, He X, Gao J, et al (2013) Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on information and knowledge management. ACM, San Francisco, USA, pp 2333–2338
20. Karpukhin V, Oguz B, Min S, et al (2020) Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 conference on empirical methods in natural language processing. Association for Computational Linguistics, Virtual Event, pp 6769–6781
21. Kim DH, Hoque E, Kim J, et al (2018) Facilitating document reading by linking text and tables. In: Proceedings of the 31st annual ACM symposium on user interface software and technology. ACM, Berlin, Germany, pp 423–434
22. Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the 31th international conference on machine learning, Beijing, China, pp 1188–1196
23. Lees AW, Yu C, Korn F, et al (2021) Collocating structured web tables with news articles. In: Proceedings of the 1st international workshop on news recommendation and intelligence co-located with the web conference 2021
24. Li J, Dou Z, Zhu Y et al (2020) Deep cross-platform product matching in e-commerce. Inf Ret J 23(2):136–158
25. Liu Y, Bai K, Mitra P, et al (2007a) Tablerank: A ranking algorithm for table search and retrieval. In: Proceedings of the 22nd AAAI conference on artificial intelligence, Vancouver, Canada, pp 317–322
26. Liu Y, Bai K, Mitra P, et al (2007b) Tableseer: automatic table metadata extraction and searching in digital libraries. In: Proceedings of the 7th ACM/IEEE joint conference on digital libraries. ACM, Vancouver,Canada, pp 91–100
27. Maity SK, Panigrahi A, Ghosh S, et al (2019) Deeptagrec: A content-cum-user based tag recommendation framework for stack overflow. In: Proceedings of the 41st European conference on information retrieval, Springer, Cologne, Germany, pp 125–131
28. Mitra B, Diaz F, Craswell N (2017) Learning to match using local and distributed representations of text for web search. In: Proceedings of the 26th international conference on world wide web. ACM, Perth, Australia, pp 1291–1299
29. Nallapati R, Zhou B, dos Santos CN, et al (2016) Abstractive text summarization using sequence-to-sequence RNNS and beyond. In: Proceedings of the 20th conference on computational natural language learning. ACL, Berlin, Germany, pp 280–290
30. Nogueira R, Cho K (2019) Passage re-ranking with BERT. CoRR abs/1901.04085
31. Pang L, Lan Y, Guo J, et al (2016) Text matching as image recognition. In: Proceedings of the 30th AAAI conference on artificial intelligence, Phoenix, USA, pp 2793–2799
32. Pimplikar R, Sarawagi S (2012) Answering table queries on the web using column keywords. Proc VLDB Endow 5(10):908–919
33. Pyreddy P, Croft WB (1997) TINTIN: A system for retrieval in text tables. In: Proceedings of the 2nd ACM international conference on digital libraries. ACM, Philadelphia, USA, pp 193–200
34. Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing. Association for Computational Linguistics, Hong Kong, China, pp 3980–3990
35. Robertson SE, Zaragoza H (2009) The probabilistic relevance framework: BM25 and beyond. Found Trends Inf Retr 3(4):333–389
36. Rush AM, Chopra S, Weston J (2015) A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 conference on empirical methods in natural language processing. The Association for Computational Linguistics, Lisbon, Portugal, pp 379–389
37. Salton G, Yang CS (1973) On the specification of term values in automatic indexing. Journal of Documentation
38. Sanh V, Debut L, Chaumond J, et al (2019) Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR abs/1910.01108. http://arxiv.org/abs/1910.01108
39. dos Santos CN, Barbosa L, Bogdanova D, et al (2015) Learning hybrid representations to retrieve semantically equivalent questions. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the Asian federation of natural language processing. The Association for Computer Linguistics, Beijing, China, pp 694–699
40. Santosh TYSS, Saha A, Ganguly N (2020) MVL: multi-view learning for news recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. ACM, Virtual Event, China, pp 1873–1876
41. Shraga R, Roitman H, Feigenblat G, et al (2020a) Ad hoc table retrieval using intrinsic and extrinsic similarities. In: Proceedings of the web conference 2020. ACM / IW3C2, Taipei, Taiwan, pp 2479–2485

42. Shraga R, Roitman H, Feigenblat G, et al (2020b) Web table retrieval using multimodal deep learning. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. ACM, Virtual Event, China, pp 1399–1408
43. Shraga R, Roitman H, Feigenblat G, et al (2020c) Projection-based relevance model for table retrieval. In: Proceedings of the web conference 2020. ACM / IW3C2, Taipei, Taiwan, pp 28–29
44. Sun C, Qiu X, Xu Y, et al (2019a) How to fine-tune Bert for text classification? In: Proceedings of the 19th China national conference on Chinese computational linguistics, Springer, Hainan, China, pp 194–206
45. Sun H, Ma H, He X, et al (2016) Table cell search for question answering. In: Proceedings of the 25th international conference on world wide web. ACM, Montreal, Canada, pp 771–782
46. Sun Y, Yan Z, Tang D et al (2019) Content-based table retrieval for web queries. Neurocomputing 349:183–189
47. Thakur N, Reimers N, Daxenberger J, et al (2021) Augmented SBERT: data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies. Association for Computational Linguistics, Virtual Event, pp 296–310
48. Trabelsi M, Davison BD, Heflin J (2019) Improved table retrieval using multiple context embeddings for attributes. In: Proceedings of the 2019 IEEE international conference on big data. IEEE, Los Angeles, USA, pp 1238–1244
49. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Proceedings of the 30th international conference on neural information processing systems, Long Beach, USA, pp 5998–6008
50. Venetis P, Halevy AY, Madhavan J et al (2011) Recovering semantics of tables on the web. Proc VLDB Endow 4(9):528–538
51. Wan S, Lan Y, Guo J, et al (2016) A deep architecture for semantic matching with multiple positional sentence representations. In: Proceedings of the 30th AAAI conference on artificial intelligence, Phoenix, USA, pp 2835–2841
52. Xiong C, Dai Z, Callan J, et al (2017a) End-to-end neural ad-hoc ranking with kernel pooling. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. ACM, Shinjuku, Japan, pp 55–64
53. Xiong C, Zhong V, Socher R (2017b) Dynamic Coattention networks for question answering. In: Proceedings of the 5th international conference on learning representations, Toulon, France
54. Zhang L, Zhang S, Balog K (2019) 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. ACM, Paris, France, pp 1029–1032
55. Zhang R, Guo J, Fan Y, et al (2018) Question headline generation for news articles. In: Proceedings of the 27th ACM international conference on information and knowledge management. ACM, Torino, Italy, pp 617–626
56. Zhang S, Balog K (2018) Ad hoc table retrieval using semantic similarity. In: Proceedings of the web conference 2018. ACM, Lyon, France, pp 1553–1562
57. Zhang S, Balog K (2020) Web table extraction, retrieval, and augmentation: a survey. ACM Trans Intell Syst Technol 11(2):1–35
58. Zhu M, Ahuja A, Wei W, et al (2019) A hierarchical attention retrieval model for healthcare question answering. In: Proceedings of the web conference 2019. ACM, San Francisco, USA, pp 2472–2482

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.