Collocating News Articles with Structured Web Tables

Alyssa Lees, You Wu, Flip Korn, Cong Yu
{alyssalees,wuyou,flip,congyu}@google.com
Google Research

Levy Silva, Luciano Barbosa*
{lss9,luciano}@cin.ufpe.br
Universidade Federal de Pernambuco

ABSTRACT

In today's news deluge, it can often be overwhelming to understand the significance of a news article or verify the facts within. One approach to address this challenge is to identify relevant data so that crucial statistics or facts can be highlighted for the user to easily digest, and thus improve the user's comprehension of the news story in a larger context. In this paper, we look toward structured tables on the Web, especially the high quality data tables from Wikipedia, to assist in news understanding. Specifically, we aim to automatically find tables related to a news article. For that, we leverage the content and entities extracted from news articles and their matching tables to fine-tune a Bidirectional Transformers (BERT) model. The resulting model is, therefore, an encoder tailored for article-to-table match. To find the matching tables for a given news article, the fine-tuned BERT model encodes each table in the corpus and the news article into their respective embedding vectors. The tables with the highest cosine similarities to the news article in this new representation space are considered the possible matches. Comprehensive experimental analyses show that the new approach significantly outperforms the baselines over a large, weakly-labeled, dataset obtained from Web click logs as well as a small, crowdsourced, evaluation set. Specifically, our approach achieves near 90% accuracy@5 as opposed to baselines varying between 30% and 64%.

CCS CONCEPTS

• **Information systems** → **Structured text search**; Document collection models; **Document representation**.

KEYWORDS

WebTables, Structured Data, Knowledge Graph, Bidirectional Encoders with Transformers, Encoders

ACM Reference Format:
Alyssa Lees, You Wu, Flip Korn, Cong Yu and Levy Silva, Luciano Barbosa. 2021. Collocating News Articles with Structured Web Tables. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3442442.3452326

1 INTRODUCTION

Web Tables have generated much interest as an invaluable resource for structured data on the Web [1]. The organized representation of complex data enables quick understanding of entity relationships. At the same time, digital news content has expanded as an increasing number of readers consume the majority of their news online [10]. Given the volume of information online, news understanding could be enhanced by surfacing contextual data relevant to the article. Indeed, analysis of Web traffic has confirmed that many readers search for Wikipedia pages directly after reading news articles[1]. Complementary research in [5] also shows how linking sentences in the article with data from table cells can further enhance the news consumption experience given a matching article and table, that is, once the article and table have been collocated. Thus, we argue that the growing abundance of data tables on the Web should be leveraged for news understanding and exploration.

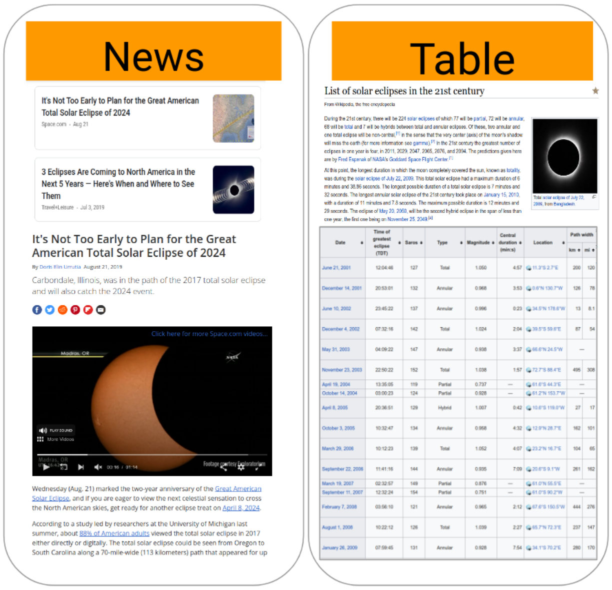
In this paper, we focus on high quality data tables from Wikipedia as the source of structured data for the *article-to-table collocation task* (see example in Fig. 1) for three reasons: (1) higher quality of tables compared to the average data table quality on the Web thanks to the Wikipedia community ecosystem; (2) rich contextual metadata that can be leveraged; and (3) accurate entity recognition for linked entity mentions based on direct mapping of entities' MIDs and their Wikipedia URLs in Freebase. While the problem of collocating data tables with news articles is novel, we explored several approaches based on existing techniques for text encoding [2] leveraging the structure of the table itself,

*Work partially done while visiting Google Research

[1] https://en.wikipedia.org/wiki/Wikipedia:Pageview_statistics

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8313-4/21/04.
https://doi.org/10.1145/3442442.3452326



It's Not Too Early to Plan for the Great American Total Solar Eclipse of 2024
Space.com · Aug 21

3 Eclipses Are Coming to North America in the Next 5 Years — Here's When and Where to See Them
Travel+Leisure · Jul 3, 2019

It's Not Too Early to Plan for the Great American Total Solar Eclipse of 2024

By Doris Elin Urrutia August 21, 2019

Carbondale, Illinois, was in the path of the 2017 total solar eclipse and will also catch the 2024 event.

Click here for more Space.com videos...
Madras, OR
PLAY SOUND
More Videos
00:16 / 01:14
Footage courtesy Exploratorium

Wednesday (Aug. 21) marked the two-year anniversary of the Great American Solar Eclipse, and if you are eager to view the next celestial sensation to cross the North American skies, get ready for another eclipse treat on April 8, 2024.

According to a study led by researchers at the University of Michigan last summer, about 88% of American adults viewed the total solar eclipse in 2017 either directly or digitally. The total solar eclipse could be seen from Oregon to South Carolina along a 70-mile-wide (113 kilometers) path that appeared for up

List of solar eclipses in the 21st century

From Wikipedia, the free encyclopedia

During the 21st century, there will be 224 solar eclipses of which 77 will be partial, 72 will be annular, 68 will be total and 7 will be hybrids between total and annular eclipses. Of these, two annular and one total eclipse will be non-central,[1] in the sense that the very center (axis) of the moon's shadow will miss the earth (for more information see gamma).[2] In the 21st century the greatest number of eclipses in one year is four, in 2011, 2029, 2047, 2065, 2076, and 2094. The predictions given here are by Fred Espenak of NASA's Goddard Space Flight Center.[1]

At this point, the longest duration in which the moon completely covered the sun, known as totality, was during the solar eclipse of July 22, 2009. This total solar eclipse had a maximum duration of 6 minutes and 38.86 seconds. The longest possible duration of a total solar eclipse is 7 minutes and 32 seconds. The longest annular solar eclipse of the 21st century took place on January 15, 2010, with a duration of 11 minutes and 7.8 seconds. The maximum possible duration is 12 minutes and 29 seconds. The eclipse of May 20, 2050, will be the second hybrid eclipse in the span of less than one year, the first one being on November 25, 2049.[4]

Total solar eclipse of July 22, 2009, from Bangladesh.

Date	Time of greatest eclipse (TDT)	Saros	Type	Magnitude	Central duration (min:s)	Location	Path width	
							km	mi
June 21, 2001	12:04:46	127	Total	1.050	4:57	11.3°S 2.7°E	200	120
December 14, 2001	20:53:01	132	Annular	0.968	3:53	0.6°N 130.7°W	126	78
June 10, 2002	23:45:22	137	Annular	0.996	0:23	34.5°N 178.6°W	13	8.1
December 4, 2002	07:32:16	142	Total	1.024	2:04	39.5°S 59.6°E	87	54
May 31, 2003	04:09:22	147	Annular	0.938	3:37	66.6°N 24.5°W	—	
November 23, 2003	22:50:22	152	Total	1.038	1:57	72.7°S 88.4°E	495	308
April 19, 2004	13:35:05	119	Partial	0.737	—	61.6°S 44.3°E	—	
October 14, 2004	03:00:23	124	Partial	0.928	—	61.2°N 153.7°W	—	
April 8, 2005	20:36:51	129	Hybrid	1.007	0:42	10.6°S 119.0°W	27	17
October 3, 2005	10:32:47	134	Annular	0.958	4:32	12.9°N 28.7°E	162	101
March 29, 2006	10:12:23	139	Total	1.052	4:07	23.2°N 16.7°E	104	65
September 22, 2006	11:41:16	144	Annular	0.935	7:09	20.6°S 9.1°W	261	162
March 19, 2007	02:32:57	149	Partial	0.876	—	61.0°N 55.5°E	—	
September 11, 2007	12:32:24	154	Partial	0.751	—	61.0°S 90.2°W	—	
February 7, 2008	03:56:10	121	Annular	0.965	2:12	67.6°S 150.5°W	444	276
August 1, 2008	10:22:12	126	Total	1.039	2:27	65.7°N 72.3°E	237	147
January 26, 2009	07:59:45	131	Annular	0.928	7:54	34.1°S 70.2°E	280	170

Figure 1: Example match between a news articles and a Wikipedia table

matching Knowledge Graph (KG) entities, and text-based document matching.

The core challenge with this problem is developing a high quality table to news article matching model. To address this, online reader behavior is leveraged to construct a weakly supervised training dataset of ⟨news article, web table⟩ pairs based on news article follow-up visits to Wikipedia table pages over a large aggregate of anonymized users. We explore features extracted from the Web Tables and articles: title, content and Knowledge Base entity categories derived from the table columns. We employ such features in a model based on Bidirectional Transformers (BERT) [2] explicitly tailored for the article-to-table matching task: the model is treated as an encoder from text input and pre-trained on a news corpus and then fine-tuned with features from co-visited news articles and Wikipedia tables. To perform the matching of a given news article, the fine-tuned BERT model encodes the news article into a dense vector, as well as the candidate tables, and the tables with the highest cosine similarities to the news article in the model's encoding space are considered as possible matches.

In our experimental evaluation, the model significantly outperformed baselines on human-rated article-table pairs, achieving top-5 accuracy of almost 90% versus an array of baselines that achieved no more than 64%. Our contributions are as follows:

(1) Introduction of a novel technical problem of collocating/ranking data tables for news articles (Section 3)
(2) Exploration of applicable approaches that rely on prior information retrieval or document matching techniques (Section 2)
(3) A novel BERT-style matching model for encoding a news article and a data table in the same semantic space for matching, as well as the novel incorporation of table column entity hypernym in representation (Section 4)
(4) A method for automatically generating large scale weakly supervised training data set with reasonably high quality (Section 5)
(5) Comprehensive empirical evaluation to demonstrate the advantage of the proposed approach (Section 6)

2 RELATED WORK

In exploring related work to the News-Table matching task, we review work on leveraging the structured nature of Web table retrieval as well text encoders.

2.1 Web Table Retrieval

Much work has been conducted on question-answer tasks using Web tables. Yin et al. [17] introduced a neural-network-based model for answering natural language (NL) questions from content on Web tables. Given an NL query, the model outputs a ranked list of entities from the tables that answer it. This is done by jointly learning representations of queries and tables using a multi-layer attention network that creates table annotations at different levels. Similar to [17], Yan et al.[13] propose a method to answer natural language queries using Web tables but instead of returning entities as response, they aim to find whole tables that match the query. For this, they implemented a two-step approach. First, they use a standard IR model (BM25) that selects k candidate tables on the table dataset, and then apply machine learning methods on the k candidate tables to rank them, using as features similarity scores between the query with different parts of the table (head, row, column and caption). The work [18] has a similar goal but they try to match keyword queries to tables using a learning-to-rank approach. For that, they compute different similarities from raw words, entities, word embeddings and entity embeddings present in both query and table and feed them into a Random Forest model to calculate their similarity. Although in this work we are also interested in the problem of table matching as these previous approaches, our goal is to match tables with news articles which have very different characteristics than NL or keyword-based queries.

2.2 Text Encoding

Different approaches have been proposed to perform text enconding in order to best represent words, sentences or even documents to capture underlying meanings and relationships which has been shown helpful in tasks such as text classification or matching. Universal Sentence Encoder (USE) [15] is an encoder that learns sentence representations primarily for matching tasks namely paraphrasing detection or entity resolution. It does so by jointly trained the encoder on three different tasks: question and answering, textual entailment and sentence similarity with the previous and next sentences. Doc2Vec [7] is another type of encoder which adapts the well-known word2vec [9] approach for paragraphs or documents, trying to capture the underlying concept on them.

State-of-the-art results in a variety of language related tasks, including question answering have been achieved based on Bidirectional Encoder Representations with Transformers (BERT)[2]. BERT models rely on pre-trained deep bidirectional layers which can be fine-tuned on specific tasks. An adaptation of BERT for News data was proposed by [14] for multi-doc headline generation. This pretrained model is the basis of the work conducted in this paper.

3 PROBLEM DEFINITION AND SOLUTION OVERVIEW

In this section, we formalize the problem we aim to tackle in this work and present an overview of our solution.

3.1 Problem Definition

Given a news article A and a set of data tables $\mathcal{T}$, the goal is to identify the most relevant table(s) in $\mathcal{T}$ to A. Formally, we define table recommendation for news articles as a ranking problem as follows:

Problem. ***[News-Table Ranking]*** *Given a news article A, a set of candidate tables $\mathcal{T}$, rank tables in $\mathcal{T}$ by their relevance to A.*

The notion of relevance is defined broadly. A table could be providing an organized summary of the news article, e.g., a *filmography of Marvel Cinematic Universe Films*[2] table for a news article on the *history of the Marvel Cinematic Universe*[3]. A table could also be providing additional background information related to the news article for readers to explore, e.g., a list of *the most expensive paintings ever sold*[4] for a news article reporting *the sale of Leonardo da Vinci's "Salvator Mundi"*[5]. We let user behavior on news-to-table exploration drive our definition of relevance, as we shall describe in Section 5.3.

3.2 Solution Overview

To perform the news-table ranking task, our goal is to identify a scoring function $f : \mathcal{A} \times \mathcal{T} \mapsto \mathbb{R}$ that scores (and thus ranks) tables in $\mathcal{T}$ for the given article A, based on content features extracted from both A and tables in $\mathcal{T}$. To create this function, we explore features in news articles and tables.

News articles features. We extract the following features from the news article. First, textual features including *HTML title* and *main content* of the news article, which is the article's content without the repeated information in the website's template (boilerplate content). We leverage boilerplate detection algorithms[6] to ignore boilerplate content. Second, semantic features including *entities* and their most salient *hypernyms* from the news article; hypernyms are the IS-A categories for the base entities, e.g., "LeBron James"→"basketball player".

Table features. Similar to the news article, we extract the following textual features from the table and its accompanying page: *HTML title* and (non-structured) textual *content*, which are directly extracted from the page containing the table; and (structured) *table content*, which are cell contents flattened in its natural order; we assume row-major, which is appropriate in almost all cases. We also extract textual features unique to the table structure: *caption* and *header*, which are meta textual content describing the table that are usually identifiable from HTML tags. Finally, for semantic features we focus on *column-aggregated entity hypernyms* to take advantage of the table structure further. Specifically, we filter for the maximum-agreement KG entity hypernym from entities within a column.

Both textual and semantic features, regardless of whether they are from news articles or tables, are represented as a sequence of word tokens as the raw input. Once extracted, the features serve as input into the scoring function f, and our goal is to develop the most effective model for the scoring function.

4 THE MATCHING MODEL

Bidirectional transformer models trained on natural text such as BERT have demonstrated huge advances over previous generalized language models [2]. Inspired by the success for deeper understanding with text embeddings, we incorporate the technique and tailor the architecture and training tasks for our news article to table collocation task. The BERT framework has two key steps: *pre-training*, where the model learns text encoders from language model tasks on a text corpus; and *fine-tuning*, which specializes the model encoder

[2] https://en.wikipedia.org/w/index.php?title=List_of_Marvel_Cinematic_Universe_films&oldid=938683141

[3] https://www.cnet.com/features/watch-every-marvel-movie-and-tv-show-in-the-perfect-order/

[4] https://en.wikipedia.org/w/index.php?title=List_of_most_expensive_paintings&oldid=938427546

[5] https://www.wsj.com/articles/leonardo-da-vinci-painting-salvator-mundi-sells-for-450-3-million-1510794281

[6] https://github.com/fhamborg/news-please

on a specific task. In our solution, we pre-train a BERT model on a large news corpus and fine-tune it on table-article pairs. The resulting model is used to perform the news-table matching by encoding a given an incoming news article and the tables in the corpus into their respective embedding vectors. The matching score is the cosine similarity between the article and the table in the model's encoding space. One can consider the tables with the highest scores as the potential matches.

4.1 Pre-Training with News Corpus

We incorporate the same pre-training architecture ($L = 12$ layers and $H = 768$ dimension size) and tasks (Masked Language Model and Next Sentence Prediction) of BERT, but perform the pre-training specifically on news corpus and a vocabulary derived from the news corpus, based on our observation that model performance is correlated to the size and the underlying vocabulary. We leveraged the work in [14], to obtain a BERT model pre-trained explicitly for news (NewsBERT). The vocabulary for BERT is also tailored to news: we extract a $50K$ vocabulary from a corpus of $50M$ online news documents using a a WordPiece tokenizer [12]. The model was then pre-trained using the same corpus of $50M$ news articles generating 1.3 billion sentences in total.

Note that we only apply the pre-training task on the news corpus: the number of documents containing high quality data tables available in our dataset (see Section 5) is much smaller comparing to the news corpus, and is therefore not suitable for pre-training. The underlying assumption we make here is that any semantic gap between the news corpus and the table corpus can be rectified in the fine-tuning stage, which will capture the intrinsic correspondences between news articles and tables.

4.2 Fine-Tuning with News-Table Pairs

Leveraging the pre-trained NewsBERT model as an encoder, we further fine-tune the embeddings corresponding to the textual and semantic features from both news articles and tables. For that, we introduce a novel fine-tuning task that captures the article-to-table relevance based on aggregated user behavior, and design a pipeline to identifying weak supervision training examples at scale.

Fig. 2 depicts the model architecture for our fine-tuning stage. Each block in the figure corresponds to a Transformer network. We use the same Transfomer encoder architecture as in pre-training for encoding a news article or a table represented as a sequence. The encoder for news article and for table share the same set of parameter, and they are initialized with the pre-trained model checkpoint.

The model is trained using as input a triple composed of a news article A, a positive table T^+ (table associated to A), and a negative table T^- (table not associated to A), for which we aim to ensure:

$$f(A, T^+) > f(A, T^-),$$

where $f(A, T)$ represents the cosine similarity between the embedding vectors of news article A and table T. This is implemented in the training by minimizing the hinge loss:[7]

$$L(A, T^+, T^-) = \max(0, f(A, T^-) - f(A, T^+) + \beta) \quad (1)$$

The training goal is therefore to learn that negative (article, table) pairs should have lower similarity than that of the corresponding positive pairs. Section 5.3 describes the details of the training data selection for the fine-tuning task.

The input for each Transformer in our architecture is the sequence representation of a news article or a table. Specifically, the raw token sequences from the news side include *title*, *content*, *entities and hypernyms*, whereas the token sequences from the table side include *title*, *(page) content*, *(column header and) hypernyms*. The input sequence for a news article or a table to the encoder is formed as a concatenation of the sequence features.

$$\langle \text{[CLS]}\ \textit{title}\ \text{[SEP]}\ \textit{hypernyms}\ \text{[SEP]}\ \textit{content} \rangle$$

The [CLS] token indicates the beginning of the sequence and [SEP] separates the different feature vector components. The semantic features of news article is a comma-separated list of ⟨*entity name,hypernym*⟩ pairs, ordered by the entities' topicality to the article. Semantic features of a table is a comma-separated list of ⟨*column header, aggregated entity, hypernym*⟩ tuples, occurring in the same left-to-right order of columns as they appear in the table. Such ordering is correlated with table editors' perception of the importance of columns for the table. We impose a maximum token length of the entire input sequence, with a budget allocated for each individual segment. Specifically, we adopted a budget of 50 tokens for *title*, 100 tokens for column entity semantics, and 150 tokens for *content*. The entire sequence is limited to 300 tokens.[8] Any unused budget of a segment is distributed to other segments (if needed) prioritized by *title* > *column-semantics* > *content*. For ablation, we also experimented with the same model architecture, with *column-semantics* de-activated, in which case the budget for *content* is increased to 250, while the budget for *title* and the maximum token length for the entire sequence remain unchanged.

4.3 Ranked Retrieval

The fine-tuned BERT model is then used to encode each table in our table corpus into an embedding vector based on the token sequences generated from the textual and semantic

[7] Hinge loss is particularly appropriate for learning-to-rank applications; see https://tinyurl.com/ranking-loss.

[8] The 300 token length limit was constricted by our resource configuration.

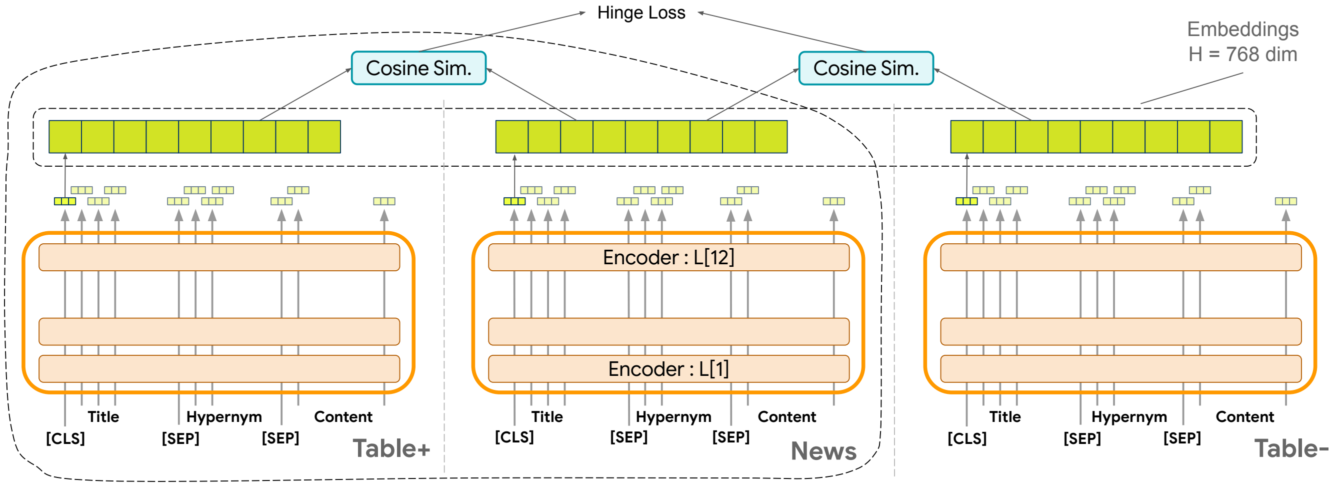


Figure 2: Fine tuning overview.

features. Given a news article, we apply the same encoding process to the token sequences from the news article to produce its embedding vector, which is then used to query the set of table vectors to retrieve those tables with the highest cosine similarities to the news article.

5 TRAINING DATA COLLECTION

In this section, we describe how we collected the news articles and table data, the features we extracted from them, and how we generated a large set of (weakly) labeled pairs for the fine-tuning step.

5.1 News Articles

For the pre-training task, we collected about 50 million of English news articles on the Web published between May 2018 and May 2019. As mentioned in Section 3.2, we extract the title and content of each article as features for training, as well as k most topical entities[9]and their respective hypernyms. While we used a proprietary knowledge base augmented with an IS-A database on its entities, a public alternative is DBpedia's `rdf.type` field for each entity.

5.2 Wikipedia Tables

For table data, we use processed Wikipedia tables[10]and limit the corpus to English list-like pages since the page title is often a good description of the content of the table(s) in the page. We ended up working with a total of about 53K Wikipedia pages having at least one large enough ($\geq$ 5 rows) horizontal table; "List(s) of ..." pages without a qualified horizontal table are not included in our training or evaluation.

For each qualified table page, we extract the page-level features as described in Section 3.2, and the table feature from the *representative table* of the page (if there exists multiple tables on the page). The representative table is chosen as the largest table (in number of cells), unless:

a. There exist *superlative table(s)* on the page,[11] in which case the first superlative table in the page is chosen as the representative table.
b. The page contains chronologically ordered tables, in which case the most recent one is chosen as the representative table.

The representative tables in our corpus compose our candidate table universe $\mathcal{T}$ of 53K Wikipedia tables.

5.3 News-Table Pairs

To establish the connection between the news articles and the Wikipedia tables, we utilize users' collective behavior of table exploration following visits to news articles. More specifically, for the fine-tuning task, positive training labels were obtained from Web session logs between August 2017 and December 2019 suitably aggregated and anonymized to satisfy strong differential privacy guarantees. From these, we obtained $\langle A, T \rangle$ co-visit counts such that the Wikipedia page containing table T was visited within 10 minutes after A was visited for a minimum percent of z sessions; in our experiments we set $z = 0.3\%$. This yielded a total of 130K positive pairs of $\langle A, T^+ \rangle$.

[9] Each named entity recognized by the proprietary entity extractor used for this is associated with a score in $[0, 1]$ indicating how topical it is to the article. See [4] for an example of a method that does this.

[10] This can be obtained from a WebTables corpus such as at http://webdatacommons.org/webtables/.

[11] A superlative table has a rank column and/or a column with rank-ordered numerical values, as defined in [6].

Negative training labels $\langle A, T^{-} \rangle$ were selected from the entire corpus of tables $\mathcal{T}$, using two different selection approaches: random and hard. In random selection, for each news article we initially removed its corresponding k matching tables from $\mathcal{T}$ and then select k random tables as negative examples.

Since random negative examples can not help the models to discriminate well between positive and negative matches, we decided to increase the complexity of the negative examples by choosing non-matching tables *closer* to the article in the embedding space. For that, we generate a ranked list of tables for an article A and conservatively sample between the 100th and 300th item in the ranked list. The ranked list were generated using cosine similarity with the baseline NewsBERT encoders. The table and article are both encoded using *title* and *content* as features. In the end, we were able to generate about 120K triples of $\langle A, T^{+}, T^{-} \rangle$ as inputs for fine-tuning.

6 EXPERIMENTS

6.1 Implementation Details

The pre-training task for NewsBERT was conducted using a 128-core slice from a TPU v3 Pod with a total of 4 TiB of High Bandwidth Memory. Hyperparameters for training consisted of standard BERT configurations including 12 hidden layers, with 768 encoding dimension. Pre-training required a couple of days with the above configuration to complete.

Our custom NewsTableBERT model was fine-tuned using a 8-core slice from a TPU v2 pod. Hyperparameters consisted of standard BERT configurations including 12 hidden layers and a learning rate of $5e^{-5}$, with 768 encoding dimension and 256 batch size. Fine-tuning with this configuration took a few hours to complete.

6.2 Evaluation Data

Since we are not aware of any dataset providing "ground truth" labels for the matching quality of some set of table-article pairs, we manually created our evaluation data by using an internal production crowd evaluation platform. We selected candidate pairs with the goal of obtaining labels with a balance between positives and negatives. The expected-positive pairs were chosen based on the same scheme we used for generating training data: if a large number of users visited a Wikipedia page containing a table shortly after visiting a news article. While random selection of an article and table is likely to result in an "obvious" negative, expected-negative pairs were chosen to be less obviously negative by taking the source Wikipedia page for the table as a 'news article', running it through NewsBERT, and finding a similar but not most similar cosine distance between the page and news article embeddings.

We selected 300 article-table pairs using this method and asked labelers to determine the quality of tables paired to news articles and whether the table provides additional context for, or insight into, the article. Labelers were told to read the news article and then browse the table, consisting of a title, a header row with column names, and a sequence of rows with values. In addition to answering whether the table was relevant to the article ("Yes", "So-so", "No"), a question about the table's level of clarity was used to ensure high enough table quality to enable assessment of relevance. Each pair was labeled by 3 to 5 labelers and each rater was restricted to evaluating at most 6 items in total. Labels were obtained from the majority of ratings; any pairs not having a majority were ignored. The resulting ground truth evaluation set consisted of about 128 labeled positive pairs. While this approach is not scalable enough to provide enough volume for training, it is useful both to validate our weakly-labeled data used in training as well as for testing the model.

Towards reproducibility of our experiments, WebTable data is publicly available[12], with all the necessary information on feature extraction for tables. We used proprietary technology for the extraction of news article content and its topical entities & hypernyms. We recommend **news-please**[13] and **Newspaper3k**[14] as publicly accessible replacements. We also used a proprietary version of Freebase and technology for Named Entity Recognition, Linking, and Typing. We point readers to **DeepType**[15] as a public replacement. We have applied for the release of the positive and negative news-table pairs curated by human rater. Unfortunately, we cannot release training data set from user activity logs. As replacement, we encourage readers to use Wikipedia references to news article from both within a table and within the page of a table (more noisy), which are also available from the WebTable data.

6.3 Evaluation Metric

We evaluate all baselines and our proposed model as a ranked retrieval task, on the data table corpus described in Section 5.2, using accuracy@k (a.k.a. top-k accuracy) that measures the percentage of news articles in the test set correctly matched to at least one of the top-k returned tables. Accuracy@k is a metric widely used in the evaluation of question answering tasks (e.g., [3]) and recommendation systems (e.g., [8]) for which, like for our problem, recall is limited and there are one or few known results of each query. We perform our evaluation by ranking, for each one of the 128 news article in the labeled positive pairs, the most similar

[12] http://webdatacommons.org/webtables/

[13] https://github.com/fhamborg/news-please

[14] https://newspaper.readthedocs.io/

[15] https://github.com/openai/deeptype

tables in our dataset of 53K Wikipedia tables, and calculating the accuracy@k for $k \in \{1, 5, 10, 100\}$ with all articles.

6.4 Baselines

We implemented the following baselines, which treat both the article and table as bags-of-words or flat documents (depending on which baseline):

- BM25 [11] is a standard information retrieval algorithm based on the probabilistic relevance framework that takes into consideration term weighting (term and document frequency) and document length for ranking. In these experiments, we used the implementation available on Elasticsearch (ES)[16].
- TF-IDF uses the standard tfidf weighting scheme and calculates the document score based on cosine similarity. We used the implementation on ES.
- Universal Sentence Encoder (USE) [15] is a sentence encoder jointly trained on three different tasks: Q&A, textual entailment and sentence similarity with the surrounding sentences. To perform text enconding, we used the implementation on Tensor Flow Hub[17].
- Doc2Vec [7] applies a language model learning strategy, similar to word2vec, to represent a sentence or document in a low-dimensional space. We used the implementation available with gensim python library[18].
- BERTpublic [2], as aforementioned, is a sentence encoder that applies a bidirectional transformer architecture for language modelling. We used the bert-as-service python package[19] to generate the encoders based on the pre-trained model BERT-Large-Uncased available on Google Research github[20].
- NewsBERT [14] contains a model specifically customized for the News space. Using the same public dataset, the model was constructed with a News specific vocabulary and pre-trained on a News Corpus.

Cosine similarity was used to measure the similarity between the vectors generated by the text encoding models USE, Doc2Vec, BERT and NewsBERT.

6.5 Proposed Matching Models

We executed the following models that implement our proposed solution:

- **N**ews**T**able-BERT (NT-BERT) was constructed by fine-tuning the NewsBERT model explicitly to the News-Table matching task, with 150K $\langle A, T^+, T^- \rangle$ triplets. The $\{T^-\}$ were randomly sampled. The model was pre-trained using just *Title* from the article and *Title, Content* from the data table as feature vectors.
- **N**ews**T**able**H**ypernym-BERT (NTH-BERT) additionally exploits the novel WebTable column entity hypernyms to fine-tune the model. The sequence representation of news articles and tables are as described in Section 4.2.

6.6 Results

We present in Table 2 the results of our proposed models and in Table 1 the performance of the best baselines. We evaluate all strategies with different combinations of the input features. The numbers show that NTH-BERT obtains the highest values of accuracy@k for all values of k. At $k = 10$, for instance, NTH-BERT correctly matches 95.3% of all articles in the test set. NT-BERT also outperforms all baselines but is inferior to NTH-BERT, which shows that the addition of hypernyms yielded a substantial improvement in the results of NTH-BERT. This confirms that our solution is in fact effective to correctly retrieve matching tables for news articles, which means that the use of a BERT style model, pre-trained on a News corpus, enhances the deep text understanding on the article side and provides context for a table within the Wikipedia page. Similarly, harnessing relevant Knowledge Graph entities from the structured nature of table columns offers a powerful tool for matching concepts.

Regarding the baselines, NewsBert's accuracy@k in Table 1 demonstrates that this model outperforms USE, Doc2Vec and BERT for accuracy @1, @5, @10, and is slightly worse for some combinations @100. It is interesting to note the great difference in performance between BERTpublic and NewsBert, showing the importance for this task of having a pre-trained model built on a news corpus.

7 DISCUSSION

Achieving much higher accuracy at $k = 1$ may be quite difficult without changing the evaluation mechanism and/or over-fitting a model. First, there are inherent flaws in the underlying assumption that the News-Table matching problem can be treated as a ranked retrieval problem. For some articles, achieving accuracy@1 for a corresponding matched table is nearly impossible due to multiple equally applicable topics. For example, an article discussing the relationship between economic growth and poverty rates in U.S. states arguably has equally good matches to the tables *U.S. states and territories by economic growth rate*[21] and *U.S. states and*

[16] https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-similarity.html

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

[18] https://radimrehurek.com/gensim/models/doc2vec.html

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

[20] https://github.com/google-research/bert

[21] https://en.wikipedia.org/w/index.php?title=List_of_U.S._states_and_territories_by_economic_growth_rate&oldid=928542351

Table 1: Results for baselines (BM25, TF-IDF, USE, Doc2Vec, BERT, NewsBERT) on human verified evaluation set.

Model	Table	News	acc.@k=1	5	10	100
BM25	⟨Title,Content⟩	⟨Title,Content⟩	**.426**	.574	.622	.831
	⟨Title,Content,Table Content⟩	⟨Title,Content⟩	.372	**.588**	**.703**	**.838**
TF-IDF	⟨Title,Content,Table Content⟩	⟨Title,Content⟩	**.453**	**.642**	**.730**	**.892**
USE	⟨Title,Content⟩	⟨Title⟩	**.250**	.439	.507	.669
	⟨Title,Content⟩	⟨Content⟩	.237	.419	.520	**.851**
	⟨Title,Content⟩	⟨Title,Content⟩	.243	**.466**	**.561**	.838
Doc2Vec	⟨Title⟩	⟨Title⟩	**.297**	**.507**	**.581**	**.824**
	⟨Title,Content⟩	⟨Title,Content⟩	.223	.378	.487	.737
BERTpublic	⟨Title,Content⟩	⟨Title⟩	**.155**	**.291**	**.372**	**.574**
NewsBERT	⟨Title,Content⟩	⟨Title,Content⟩	**.458**	**.725**	**.779**	**.824**

Table 2: Results for specialized BERT based model, NT-BERT and NTH-BERT, on human verified evaluation set. Compared to the baseline of NewsBERT.

Model	Table	News	acc.@k=1	5	10	100
NewsBERT	⟨Title⟩	⟨Title⟩	.422	.602	.656	.719
	⟨Title⟩	⟨Title,Content⟩	.438	.677	.746	.8
	⟨Title,Content⟩	⟨Title⟩	.176	.244	.313	.588
	⟨Title,Content⟩	⟨Title,Content⟩	.458	.725	.779	.824
NT-BERT	⟨Title⟩	⟨Title⟩	.545	.773	.833	.939
	⟨Title⟩	⟨Title,Content⟩	.53	.818	.871	.955
	⟨Title,Content⟩	⟨Title⟩	.545	.795	.841	.932
	⟨Title,Content⟩	⟨Title,Content⟩	.553	.856	.879	.947
NTH-BERT	⟨Title,Hypernyms,Content⟩	⟨Title⟩	.575	.811	.858	.961
	⟨Title,Hypernyms,Content⟩	⟨Title,Hypernyms⟩	.543	.803	.858	.953
	⟨Title,Hypernyms,Content⟩	⟨Title,Hypernyms,Content⟩	**.669**	**.898**	**.953**	**.976**

territories by poverty rate[22]. Another concern is that with longer articles covering manifold topics the *best* match is entirely subjective. In general, as the table corpus grows, the likelihood of multiple good matches increases. Our ad hoc analysis of top results showed that in numerous cases the top few results were all reasonable and difficult to rank in quality. For example, the article *GDP Is In, And Recession Is Out... Or Is It?*[23], discusses GDP but also the prospect of a major financial crisis. The top-3 tables retrieved were as follows:

(1) *List of recessions in the United States*[24]
(2) *List of countries by past and projected GDP (nominal)* [25]
(3) *List of countries by past and projected GDP (PPP)*[26]

8 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced the problem of collocating Web table and news articles to foster news understanding and exploration. Our proposed solution is based on the BERT framework, which first pre-trains a BERT encoder on a large news corpus, and then fine-tunes it based on a table-article pairs. Our evaluation shows that our model outperforms traditional text encoder strategies and information retrieval techniques, and having encoders trained in the domain corpus have a great impact on the model's performance.

In terms of evaluation, constructing a larger human curated training/evaluation dataset model may enhance the model and reader satisfaction. In terms of the model, multiple enhancements could be explored:

- Use the whole web table content - including embedding rows and incorporating an explicit hierarchical structured, similar to [16].
- Pre-train BERT model as a classification task for KG entities or a similar procedure to massage the model to learn associations between entity concepts. Such work has been explored in [19].
- Mask dates and numbers in pretraining, fine-tuning and inference, employing separate matching models for such instances.

This work can be a first step towards table-enhanced news intelligence. Possible future directions include but are not limited to: finer-granular association of web tables with snippets of text rather than a document as a whole; detection of most relevant view of a web table in the context of the news.

[22] https://en.wikipedia.org/w/index.php?title=List_of_U.S._states_and_territories_by_poverty_rate&oldid=939233580
[23] https://seekingalpha.com/article/4321877-gdp-is-in-and-recession-is-out-is
[24] https://en.wikipedia.org/wiki/List_of_recessions_in_the_United_States
[25] https://en.wikipedia.org/wiki/List_of_countries_by_past_and_projected_GDP_(nominal)
[26] https://en.wikipedia.org/wiki/List_of_countries_by_past_and_projected_GDP_(PPP)

REFERENCES

[1] Michael Cafarella, Alon Halevy, Hongrae Lee, Jayant Madhavan, Cong Yu, Daisy Zhe Wang, and Eugene Wu. 2018. Ten Years of Webtables. *PVLDB* 11, 12 (2018), 2140–2149.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.

[3] Cicero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *ACL*. 694–699.

[4] Michael Gamon, Tae Yano, Xinying Song, Johnson Apacible, and Patrick Pantel. 2013. Identifying salient entities in web pages. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 2375–2380. https://doi.org/10.1145/2505515.2505602

[5] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. 2018. Facilitating Document Reading by Linking Text and Tables. In *ACM UIST*. 423–434.

[6] Flip Korn, Xuezhi Wang, You Wu, and Cong Yu. 2019. Automatically Generating Interesting Facts from Wikipedia Tables. In *SIGMOD*. 349–361.

[7] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. 1188–1196.

[8] Suman Kalyan Maity, Abhishek Panigrahi, Sayan Ghosh, Arundhati Banerjee, Pawan Goyal, and Animesh Mukherjee. 2019. DeepTagRec: A Content-cum-User Based Tag Recommendation Framework for Stack Overflow. In *ECIR*. Springer, 125–131.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR*.

[10] Pews Research. [n.d.]. Newspaper Fact Sheet. *Pews Research* ([n. d.]). https://www.journalism.org/fact-sheet/newspapers/

[11] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *TRIR* 3, 4 (2009), 333–389.

[12] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *ACL*. 1715–1725.

[13] Yibo Sun, Zhao Yan, Duyu Tang, Nan Duan, and Bing Qin. 2019. Content-based table retrieval for web queries. *Neurocomputing* 349 (2019), 183–189.

[14] Jiawei Han Jialu Liu Hongkun Yu You Wu Cong Yu Daniel Finnie Jiaqi Zhai Xiaotao Gu, Yuning Mao and Nicholas Zukoski. 2020. Generating Representative Headlines for News Stories. In *WWW*. 1773–1784.

[15] Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning Semantic Textual Similarity from Conversations. In *Rep4NLP*. 164–174.

[16] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *HLT-NAACL*.

[17] Pengcheng Yin, Zhengdong Lu, Hang Li, and Ben Kao. 2016. Neural Enquirer: Learning to Query Tables in Natural Language. In *IJCAI*. 2308–2314.

[18] Shuo Zhang and Krisztian Balog. 2018. Ad Hoc Table Retrieval using Semantic Similarity. In *WWW*. 1553–1562.

[19] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*. 1441–1451.