Putting Web Tables into Context

Katrin Braunschweig, Maik Thiele, Elvis Koci and Wolfgang Lehner
Database Technology Group, Department of Computer Science, Technische Universitat Dresden, Dresden, Germany

Keywords: Information Extraction, Web Tables, Text Tiling, Similarity Measures.

Abstract: Web tables are a valuable source of information used in many application areas. However, to exploit Web tables it is necessary to understand their content and intention which is impeded by their ambiguous semantics and inconsistencies. Therefore, additional context information, e.g. text in which the tables are embedded, is needed to support the table understanding process. In this paper, we propose a novel contextualization approach that 1) splits the table context in topically coherent paragraphs, 2) provides a similarity measure that is able to match each paragraph to the table in question and 3) ranks these paragraphs according to their relevance. Each step is accompanied by an experimental evaluation on real-world data showing that our approach is feasible and effectively identifies the most relevant context for a given Web table.

1 INTRODUCTION

The Web has developed into a comprehensive resource not only for unstructured or semi-structured data, but also for relational data. Millions of relational tables embedded in HTML pages or published in the course of Open Data/Open Government initiatives provide extensive information on entities and their relationships from almost every domain. Researchers have recognized these Web tables as an important source of information for applications such as factual search (Yin et al., 2011), entity augmentation (Eberius et al., 2015; Yakout et al., 2012), and ontology enrichment (Mulwad et al., 2011).

These Web tables are very heterogeneous, often with ambiguous semantics and inconsistencies in the quality of the data. Consequently, inferring the semantics of these tables is a challenging task that requires a designated table understanding process. Since Web tables are typically very concise, additional contextual information is needed to understand their content and intention. On the Web, we encounter context in the form of headlines, captions or surrounding text. Text referring to a table can provide a summary of the content or conclusions drawn from it. It also frequently offers a more detailed description of various table entries to clarify terms or indicate restrictions on attributes (Hurst, 2000). However, not all information mentioned in potential context sections is actually relevant to the table. For instance, a query term that appears in the context of a table does not guarantee that the answer to the query is contained in the table. The verbosity of the context, especially when considering large texts, often introduces noise that leads to incorrect interpretations (Pimplikar and Sarawagi, 2012). Consequently, evaluating the relevance of potential context segments as well as establishing an explicit link to the table content is essential in order to reduce noise and prevent misinterpretations. In this paper we take a closer look at the context available for tables on the Web in order to improve its impact on Web table understanding. Therefore, we focus on the relevance of different context sources with respect to providing useful additional information on table content. Our objective is to identify measures that enable the evaluation of context information regarding its connection to table content and the reduction of noise based on these measures. By reducing the noise, we expect table understanding approaches based on table context as well as information extracted from it to be less ambiguous.

Problem Statement. We view the challenge of reducing the noise in the table context as a paragraph selection problem. Consequently, the objective is to identify paragraphs in long text segments which are semantically related or relevant to a table. We can split this problem into three subtasks, as illustrated in Figure 1, which also determine the structure of the paper.

1. First, the text is decomposed into topically coherent paragraphs. Selecting the best segmentation

Braunschweig, K., Thiele, M., Koci, E. and Lehner, W.
Putting Web Tables into Context.
DOI: 10.5220/0006034701580165
In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR*, pages 158-165
ISBN: 978-989-758-203-5

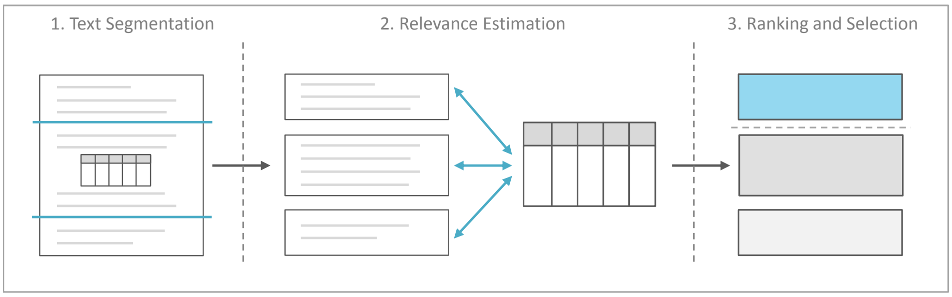


Figure 1: Overview of the paragraph selection task.

granularity is important, as the paragraph size can affect the selection process. If paragraphs are too large, we run the risk of retaining noise in the context. Yet, if paragraphs are too small, they do not provide enough content to make an informed decision regarding their relevance to the table.

2. Second, a similarity measure is used to match each paragraph to the table in question in order to evaluate its relevance. Since the overlap between table content and context information is very limited, it is important to select an appropriate measure to estimate the relevance of the paragraphs.
3. And finally, all paragraphs are ranked according to their relevance and irrelevant, noisy paragraphs are filtered out.

2 TEXT SEGMENTATION

The objective of the text segmentation step is to split long text sections into semantically coherent segments or paragraphs. various text segmentation algorithms have been proposed in the literature, including algorithms addressing lexical cohesion (Hearst, 1997), topic detection and tracking algorithms (Allan, 2002), as well as probabilistic models (Beeferman et al., 1999). In this paper, we employ a linear text segmentation approach similar to *TextTiling* (Hearst, 1997), which detects shifts in topics by measuring the change in vocabulary within the text. Using a sliding window approach, vocabulary changes are detected by measuring the coherence between adjacent text sections. Significant changes in coherence determine the position of break points, as they indicate topic shifts. In detail, the approach we adopted works like follows:

1. **Coherence Scores:** To measure the coherence, the text is first tokenized and split into smaller units. Common units are sentences or pseudo-sentences, i.e. token sequences of fixed length. While sentences provide for more natural boundaries, pseudo-sentences ensure that sections of equal size are compared. Two adjacent blocks of size b (in text units) are used to measure the change of vocabulary, as illustrated in Figure 2. A text similarity measure, such as the cosine similarity of the term frequency vectors, determines the coherence score s_c at the gap between both blocks. Sliding through the text with a step size of one text unit (sentence or pseudo-sentence), the coherence is measured throughout the text, resulting in a sequence of coherence scores. Low scores indicate potential topic shifts.
2. **Smoothing:** Before identifying the break points in the text, the sequence of coherence scores is smoothed to reduce the impact of small variations in coherence. In this paper, an iterative moving average smoothing is applied.
3. **Depth Scores:** To identify suitable break points, the gaps of interest are the locations of the local minima of the coherence sequence. The significance of a topic shift is indicated by the depths of a local minimum compared to the coherence of neighboring text sections. This depth score s_d is defined as the sum of the differences in coherence between local minimum i and the closest local maxima before (m_-) and after (m_+) the minimum, respectively.

$$s_d(i) = s_c(m_-) + s_c(m_+) - 2 \cdot s_c(i) \qquad (1)$$

4. **Break Points:** As only significant vocabulary changes are likely to represent topic shifts in the text, a threshold is defined to filter out insignificant changes. Only gaps with a depth score $s_d \geq \mu - t \cdot \sigma$ are selected as break points. t is an adjustable threshold parameter, while μ and σ are the mean and standard deviation of the depth scores, respectively. In some cases, the resulting break point requires further adjustment. If pseudo-sentences are used and a break point lies within a sentence, the next sentence break before or after the break point is used instead. Similarly, if the source text contained structural information,

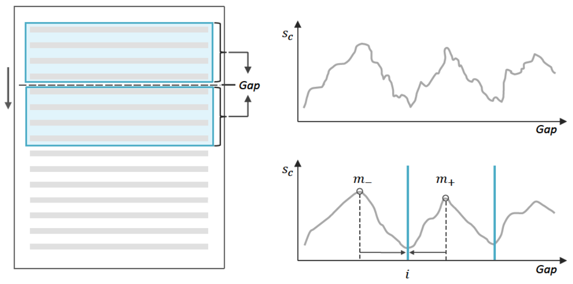


Figure 2: Linear text segmentation in *TextTiling* algorithm.

such as paragraphs, break points can be adjusted to fall on paragraph breaks nearby.

There are several parameters in the text segmentation algorithm that impact the quality of the returned segments, including the unit and block sizes, l and b, as well as the threshold parameter t. In addition, the selection of an appropriate similarity measure as well as smoothing technique also influence the number and location of break points. The optimal parameters depend on the characteristics of the text corpus.

3 RELEVANCE ESTIMATION USING WORD- AND TOPIC-BASED SIMILARITY

After splitting the context into smaller topical sections, our next goal is to evaluate the relevance of each segment with respect to the table content. Treating both the table and the context as a bag of words, i.e. assuming independence between words, we first focus on word-based similarity measures to estimate the relevance. The assumption is that if words from the table content, such as attribute labels or cell entries, are frequently mentioned in the context, it is very likely that the text describes the table. Processing tables as a loose collection of words is a common approach, for example to find related tables (Cafarella et al., 2009) or to retrieve tables that match a user query (Pimplikar and Sarawagi, 2012). Incorporating the table structure, which often provides implicit information about the dependencies between table entries, is much more difficult, because table structures are very heterogeneous and there are no general rules that apply to all tables.

Word-based similarity measures generally consider the frequency of terms as well as their significance to evaluate the similarity between text segments. Regarding a table as a loose collection of words, we face several issues. First of all, tables present information in a compact format, with most textual entries limited to words or short phrases and some of the information represented implicitly through the semantics of the table structure. Compared to text, tables contain significantly less explicit information, leading to very sparse term vectors. Second, the frequency of terms in a table is not representative of their importance regarding the tables main topic. Attribute labels, which are designed to describe the table content, generally only appear once, while some attribute values can appear numerous times due to redundancy in the table.

These characteristics are very similar to the characteristics of keyword queries in text retrieval systems. Compared to a longer text document, the term vector of a query is also very sparse, with little or no repetition of terms. Consequently, we can regard a table as a long keyword query. In the literature, a wide range of retrieval functions has been proposed, which score documents based on query relevance and sparsity. These retrieval functions present one option to identify relevant context segments for Web tables.

Considering that we also have a significant number of large tables on the Web, which feature a term count similar to that of the average context paragraph, we can consider text similarity as another option to evaluate context relevance.

3.1 Retrieval Functions

First, we consider several state-of-the-art retrieval functions to estimate the relevance of context segments with respect to a table. The objective of a retrieval function is to rank documents based on their relevance to a query, which generally is a list of keywords or phrases. Different retrieval functions use different techniques to address the importance of query terms and the length of the document. As retrieval functions we consider the following established techniques: As the first retrieval function, we use the *TF-IDF* score. As we only match the table terms to the paragraphs of the table's context, not to paragraphs in the context of other tables, we define the IDF score per table context, as the number of paragraphs in the context divided by the number of paragraphs that contain term t. We refer to this score as the *Inverse Paragraph Frequency (IPF)*. As a second retrieval approach, we consider language models, probabilistic models that reflect the distribution of terms in documents (Ponte and Croft, 1998). We consider unigram models, which assume independence between terms. Retrieval based on language models ranks documents based on the likelihood of the query (or table T) given the document model M_D. Additionally we apply smoothing techniques such as Jelinek-

Mercer smoothing and Dirichlet smoothing to avoid issues for query terms that do not appear in a document. As the last group of retrieval functions, we consider *Okapi BM25*, a probabilistic retrieval function frequently used for text retrieval (Robertson et al., 1996). To score a document, the scoring function takes into account the frequency of a term both in the query and the document as well as the inverse document frequency of the term. In addition, the size of the document $|D|$ as well as the average document size of the collection *avgdl* are included to correct the score depending on the size of the document at hand.

3.2 Similarity Metrics

In addition to various document retrieval functions, we compare various symmetric text similarity measures. Probably the most common text similarity measure is the cosine similarity of weighted term vectors, which has also been used frequently in the Web table recovery literature. Besides this measure, we also consider two alternative measures, proposed by (Whissell and Clarke, 2013), which represent symmetric variants of popular retrieval functions. In detail, we consider the following text similarity measures in our study: The cosine similarity *Cosine TF-IDF* represents a similarity measure frequently used in connection with the vector space model, which models texts or documents as vectors of term weights, one for each term in a dictionary (Salton and Buckley, 1988). The most common weighting functions for the terms in the documents include the TF-IDF score and its variants. We further consider a symmetric similarity measure based on language models proposed by (Whissell and Clarke, 2013). Using language models, it is often very likely for two long documents to feature many similarities simply due to terms that are very frequent in the language and, thus, appear in most documents. To account for this scenario, (Whissell and Clarke, 2013) incorporate $\log P(*|C)$ to model chance, where C is the collection or background model that reflects the general term frequency in the language. (Whissell and Clarke, 2013) also propose a symmetric variant of the Okapi BM25 retrieval function. To ensure symmetry, the first factor of the retrieval function is replaced by a factor that equals the second factor, utilizing the same parameters k_1 and b. Again, we can use different variants of IDF or omit the score.

3.3 Topic-based Similarity

If the vocabulary in the documents is large, word-based similarity measures operate in a very high-dimensional space, as the size of the vocabulary determines the dimensionality. In practice, computing the similarity in such a high-dimensional space can be computationally very expensive and impractical. Furthermore, the analyzed data becomes very sparse in such a high-dimensional space, which can impact the ability to identify similar documents. To address this dimensionality issue associated with word-based similarity measures, we consider *topic modeling* as an alternative. Instead of comparing tables and context segments at word level, where the size of the vocabulary determines the dimensionality, we compare the *topics* associated with the words, instead. In addition to reducing the dimensionality, topic modeling also enables the identification of implicit matches between a table and the associated context, where both describe the same topic, but do not explicitly use the same terms to describe it. Such relations between tables and context cannot be detected via word-based matching.

Topic modeling aims to identify abstract topics in a document collection and to model each document based on its association with one or more of these topics. In general, the presence of a topic is determined from the frequency and co-occurrence of terms in the document, and, in most cases, a topic is represented as a probability distribution over all terms in the vocabulary. In this paper, we focus on LDA to model topics in tables as well as their context.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model to model topics in text collections (Blei et al., 2003). LDA is based on a generative process that allows for documents to cover multiple topics. The model incorporates topics K, documents D and words from a fixed vocabulary N. $W_{d,n}$ denotes the n-th word in document d. The topical structure in a corpus is modeled by random variables ϕ_k, θ_d and $Z_{d,n}$. Each topic k is described by a multinomial probability distribution ϕ_k over the term vocabulary. Furthermore, each document d is associated with a multinomial distribution θ_d which describes the topic proportions for the document. As each document can describe multiple topics, this distribution reflects the mixture of topics in the document. Finally, $Z_{d,n}$ denotes the topic assignment for word n in document d. Variables α and β are hyper-parameters.

Similarity Measures. For each document or document section, the LDA inference generates a vector of *topic proportions* $\widehat{\theta}_d$, which represents a discrete probability distribution over all topics. Consequently, in order to measure the similarity between the topical representations of two documents, we can apply measures that quantify the similarity between probability distributions. (Blei and Lafferty, 2009) suggest the *Hellinger distance*.

Another common measure is the *Kullback-Leibler (KL) divergence*, which, in its original form, is not strictly a metric, as it is not symmetric. The KL divergence of probability distribution P_2 from P_1 is denoted as $D_{KL}(P_1||P_2)$. As a symmetric variant of the KL divergence we use th sum of $D_{KL}(d_1||d_2)$ and $D_{KL}(d_2||d_1)$.

3.4 Evaluation and Comparison

All presented approaches present viable options for table-to-context matching. Considering the diverse characteristics of Web tables, it is difficult to favor one approach over the other. To gain a better understanding of the functionality and behavior of these measures with respect to Web tables, we conduct a comparative study, analyzing different measures proposed in the literature for a test set of Web tables. For the evaluation, we use a set of 30 tables extracted from the English Wikipedia along with their respective pages. To retrieve context paragraphs, we utilize the original structure of the Wikipedia articles, considering headlines as natural topic boundaries. We manually judged each resulting paragraph as either *relevant* or *not relevant*. After applying the different measures to score the context paragraphs, we use Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP) to evaluate the suitability of the similarity measures.

To determine the suitability of these retrieval functions, we evaluate each retrieval function (with different parameter settings, if applicable) on the test set. To analyze the sensitivity of each approach to table size, we split the test set into small, medium sized and large tables, with table sizes of less than 20 terms, between 20 and 200 terms and more than 200 terms, respectively. In Figure 3, we present the best overall MRR and MAP scores for each approach.

Overall, all retrieval functions achieve high scores with only little variance across the different functions. For our test set, the highest MMR and MAP scores are achieved using retrieval functions based on language models with Dirichlet smoothing (see Figure 3(a)). The results indicate that more sophisticated retrieval functions such as language models or BM25 are better suited for the identification of relevant context than the simple weighting scheme. The symmetric similarity measures show only little variance between the different measures.

However, comparing the cosine and BM25 similarity measures, we can see that for our test set the cosine similarity with simple TF weights outperforms the more complex weighting scheme of BM25 (see Figure 3(c)). For both approaches, we can also observe that inverse paragraph frequency (IPF) performs better than the other IDF variants.

The analysis of the various symmetric similarity measures for different table sizes shows slightly more variation. As expected, all measures achieve the highest scores for larger tables with more than 200 terms. Overall, symmetric text similarity measures produced results of high quality.

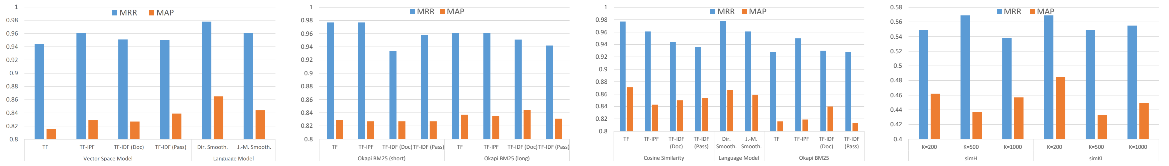
To evaluate the suitability of LDA to estimate the relevance of table context, we trained the LDA model using a corpus of $1,000$ English Wikipedia articles (from which the test tables were selected). For the hyper-parameters, we use the settings often recommended in the literature (Wei and Croft, 2006), with $\beta = 0.01$ and $\alpha = \frac{50}{K}$, where K is the number of topics considered in the model. We varied the number of topics in our experiments, but limit the results presented here to $K \in \{200, 500, 1000\}$.

For each table and each context segment, we infer a topic distribution using the trained LDA model and measure the similarity between the topic distribution of the table and the topic distribution of each segment in the context of the table. Figure 3(d) shows the overall MRR and MAP scores. The scores achieved on our test set with different topic counts indicate that topic modeling is significantly less effective in estimating the context relevance, compared to word-based matching techniques. We observe only very little variation for different topic counts.

In our analysis, we can identify two possible reasons for the significantly lower results. The first issue is the table size. It appears that most tables in our test set are too small, i.e. they contain too few terms, in order to enable a meaningful inference of topic distributions. An analysis of tables of different sizes confirms this assumption, as both MRR and MAP scores improve with increasing table size.

The second issue is the topic count and granularity. In an open domain scenario, such as the Web, we face a huge number of possible topics, which is replicated, although on a slightly smaller scale, on Wikipedia. Using very general topics reduces the number, however, in order to distinguish between the topics of paragraphs of the same document, we require very detailed topics. Consequently, it is very difficult to model the subtle differences between paragraphs, if the overall corpus is heterogeneous and diverse.

For our test set, directly applying LDA to Web tables and their respective context segments does not offer any benefits compared to the word-based similarity measures. Therefore, we do not further consider this approach in the remainder of this paper.



(a) Retrieval Functions: Vector Space and Language Model

(b) Retrieval Functions: Okapi BM25 short and long

(c) Similarity Measures: Cosine Similarity, Language Model and Okapi BM25

(d) Latent Dirichlet Allocation

Figure 3: Evaluation of retrieval functions, similarity measures and LDA to estimate the relevance of context segments.

4 CONTEXT SELECTION - FILTER AND THRESHOLD

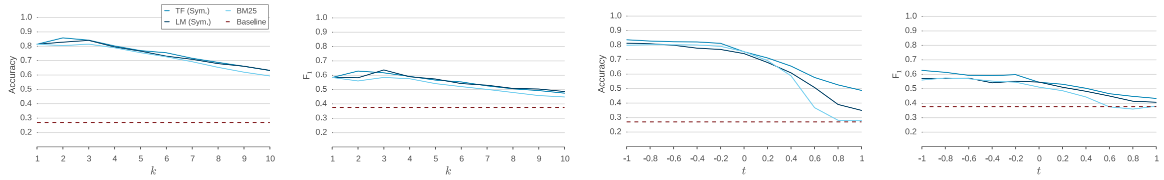
After evaluating the relevance of each context section with respect to the table, using a retrieval function or symmetric text similarity measure, we can retrieve a ranked list of context sections. In the final step, we need to decide which context sections to keep for subsequent processing and which sections to discard as irrelevant or noisy. Consequently, we require a threshold for the relevance score.

Finding the optimal threshold for a large collection of tables and their respective contexts is very challenging, as the Web pages can have very different characteristics. In some cases, only a small section on the Web page is related to the content of a table, while in other cases the entire Web page can be regarded as relevant. Furthermore, the similarity measures are not always able to make a clear distinction between relevant and irrelevant context. Thus, when selecting a relevance threshold, we face a trade-off between eliminating noise in the context and missing potentially relevant information. To address this trade-off, we consider two alternative threshold specifications: A *rank-based threshold* is a popular selection approach in retrieval systems. Instead of considering the value of the relevance score, context segments are regarded as relevant based on their position in the ranked list of all context sections. Only the top k sections are retrieved.In contrast, the *score-based threshold* is not associated with a fixed position in the ranked list, and, instead, takes the variance of the relevance scores across the context sections into account. In particular, the threshold is defined as follows: $\theta_{score} = \mu - t \cdot \sigma$, where μ and σ are the mean and standard deviation of the relevance scores, respectively.

While the rank-based threshold returns the same number of context sections for each table, the score-based threshold is different for each table. For each approach, we can adjust the threshold by varying the parameters k and t, respectively. Using the test set of 30 Web tables and associated context sections, we can analyze the characteristic behavior of each threshold approach. Varying the threshold parameters, we measure the accuracy as well as the F_1 measure, averaged across all tables. *Accuracy* measures the percentage of correctly identified context sections, i.e. the number of relevant sections that have been retrieved as well as the number of irrelevant sections that have been discarded. The F_1 measure only considers the retrieved sections and takes into account precision and recall. The precision states how many of the retrieved sections are actually relevant to the table, while recall states how many of the relevant sections have been retrieved. Figure 4 shows the results. In our evaluation, we consider three different measures to compute the relevance scores of the context sections. As the different retrieval functions and similarity measures we studied in the previous section all produce very different relevance scores and rankings, the choice of a scoring function can influence the quality of the retrieved context sections. For the experiments, we consider the cosine similarity of TF scores, a symmetric similarity score based on language models with Dirichlet smoothing (LM) as well as the BM25 retrieval function. As a baseline, we measure the accuracy and F_1 for the case where all context sections are retrieved. Consequently, a higher score indicates an improvement achieved through context selection.

Figures 4(a) and 4(b) show the results for the rank-based threshold approach for k in the range $[1, 10]$. There are only little differences between the various scoring functions. We can see an obvious improvement over the baseline, which is decreasing as more context is retrieved. The less significant improvement in F_1 measure indicates the weakness of a fixed rank-based threshold. With a fixed number of retrieved context sections, this approach does not adapt very well to the different characteristics of Web table context, where some tables have significantly less relevant context sections than others.

The results of the score-based threshold are presented in Figures 4(c) and 4(d), for parameter t in the range $[-1, 1]$. Here we can clearly see the impact of



(a) Accuracy of Rank-based Threshhold (b) F_1 of Rank-based Threshhold (c) Accuracy of Score-based Threshhold (d) F_1 of Score-based Threshhold

Figure 4: Average accuracy and F_1 measure of context selection using rank-based and score-based thresholds.

the scoring function, especially for higher values of t. Overall, the maximum accuracy and F_1 values that can be achieved with this threshold approach are very similar to those achieved with a rank-based threshold. The score-based threshold assumes some variation in the relevance scores of the context sections. However, if all context sections are equally relevant and receive very similar scores, the threshold discards some of the relevant sections, which is reflected by F_1.

5 RELATED WORK

A table in a document is generally not an independent object, but one of many components that carry information content. Table recovery research has identified the potential for other document components, such as titles or text, to affect how table content is interpreted (Embley et al., 2006). Therefore, various work related to table recognition and table understanding, as well as applications that utilize document tables, take the context of a table into account, e.g. the table retrieval system *TINTIN* or the question answering system *QuASM* (Pinto et al., 2002) that consider text that is close to or between rows of a table (Pyreddy and Croft, 1997). To improve the quality of the process involved in table and context identification and extraction, the authors of (Pinto et al., 2003) proposed a classification approach based on conditional random fields. Again, they focus solely on context that is located directly before, after or within the boundaries of the table.

In his thesis (Hurst, 2000), Hurst also includes text segments that are not co-located with the table, such as headings and the main text, and studies formats of references to tables in the text. Such references can be *explicit*, including an index for the table, as in “shown in Table 2.2”, or *implicit*, without a unique string, as in “in the following table”. Hurst focuses mainly on the extraction of tables and context information, but does not consider the relevance of context segments with respect to a table, or the utilization of contextual information to interpret the table content.

Contextual information is taken into account in various applications. These include the identification of a semantic relation between tables (Yakout et al., 2012), establishing the relevance of a table in response to a search query (Limaye et al., 2010; Pimplikar and Sarawagi, 2012), as well as the detection of hidden attributes (Cafarella et al., 2009; Ling et al., 2013). The selection of contextual information that is considered suitable differs significantly amongst the individual approaches. While many approaches do not define a specific selection of context and simply consider all available information, others limit the amount of information considered. In (Yakout et al., 2012) the context is restricted to text that directly surrounds the table on the Web page. In (Cafarella et al., 2009) the authors further reduce the contextual information by taking only significant terms into account, specifically the top-k terms based on *TF-IDF* scores. A more elaborate context selection technique is proposed by (Pimplikar and Sarawagi, 2012) and subsequently applied by (Sarawagi and Chakrabarti, 2014). Relevant context segments are selected based on their position in the DOM tree of the document. Considered context types include heading and text segment. Starting from the path between the table node and the root of the DOM tree, all text nodes that are siblings of nodes on the path are included. In order to estimate its relevance to the table, each of these nodes is then scored based on its distance from the table, its position relative to the path as well as the occurrence of formatting tags such as *bold* or *italics*. However, as (Pimplikar and Sarawagi, 2012) skip further details about the extraction of context segments, the suitability of this technique is difficult to evaluate.

For search applications, the similarity between the search terms and the table with its context is computed, whereas integration applications measure the similarity between two tables and their respective context segments. In (Yakout et al., 2012) *TF-IDF* is applied to identify conceptually related tables, considering the similarity between both context sections as well as a table-to-context similarity between one table and the context associated with the other table (Yakout et al., 2012). This simple measure is further extended by (Pimplikar and Sarawagi, 2012) to enable collective matching that incorporates the table and context into a single similarity score.

In summary, the context of a table is frequently recognized as an important resource in table understanding. However, the relevance of specific context paragraphs with respect to a table has received only limited attention so far.

6 CONCLUSION

To exploit the rich information stored in billions of Web tables, additional contextual information is needed to understand their content and intention. In the most general case the overall document containing the Web table could be considered to support table understanding. However, since most of the context will not be related to the Web table at all this introduces to much noise. Therefore, we proposed a novel contextualization approach for Web tables based on text tiling and similarity estimation to evaluate the relevance of context information. We performed a detailed analysis of state-of-the-art retrieval functions such as TF-IDF, language models, Okapi BM25, and LDA and applied them on the Web table in question as well as the different context paragraphs. Our evaluation showed that language models with Dirichlet smoothing deliver excellent results with an MRR score of almost 0.98. We finally studied different ranking schemes that enable us to effectively identify the most relevant context paragraphs for a given Web table.

REFERENCES

Allan, J. (2002). Introduction to topic detection and tracking. In *Topic Detection and Tracking*, pages 1–16. Kluwer Academic Publishers.

Beeferman, D., Berger, A., and Lafferty, J. (1999). Statistical models for text segmentation. *Machine Learning - Special Issue on Natural Language Learning*, 34(1-3):177–210.

Blei, D. M. and Lafferty, J. D. (2009). Topic models. *Text Mining: Classification, Clustering, and Applications*, 10:71.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Cafarella, M. J., Halevy, A. Y., and Khoussainova, N. (2009). Data integration for the relational web. *Proceedings of the VLDB Endowment*, 2:1090–1101.

Eberius, J., Thiele, M., Braunschweig, K., and Lehner, W. (2015). Top-k entity augmentation using consistent set covering. In *SSDBM'15*, SSDBM '15, pages 8:1–8:12, New York, NY, USA. ACM.

Embley, D. W., Hurst, M., Lopresti, D., and Nagy, G. (2006). Table-processing paradigms: a research survey. *IJDAR'06*, 8(2-3):66–86.

Hearst, M. A. (1997). Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Hurst, M. (2000). *The Interpretation of Tables in Texts*. PhD thesis, University of Edinburgh.

Limaye, G., Sarawagi, S., and Chakrabarti, S. (2010). Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3:1338–1347.

Ling, X., Halevy, A. Y., Wu, F., and Yu, C. (2013). Synthesizing union tables from the web. In *IJCAI'13*, pages 2677–2683.

Mulwad, V., Finin, T., and Joshi, A. (2011). Generating linked data by inferring the semantics of tables. In *VLDS'11*, pages 17–22.

Pimplikar, R. and Sarawagi, S. (2012). Answering table queries on the web using column keywords. *Proceedings of the VLDB Endowment*, 5(10):908–919.

Pinto, D., Branstein, M., Coleman, R., Croft, W. B., King, M., Li, W., and Wei, X. (2002). Quasm: A system for question answering using semi-structured data. In *JCDL'02*, pages 46–55. ACM.

Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003). Table extraction using conditional random fields. In *SIGIR'03*, pages 235–242. ACM.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR'98*, pages 275–281. ACM.

Pyreddy, P. and Croft, W. B. (1997). Tintin: A system for retrieval in text tables. In *JCDL'97*, pages 193–200. ACM.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1996). Okapi at trec-3. In *TREC'96*, pages 109–126.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5):513–523.

Sarawagi, S. and Chakrabarti, S. (2014). Open-domain quantity queries on web tables: Annotation, response, and consensus models. In *SIGKDD'14*, pages 711–720.

Wei, X. and Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *SIGIR 2006*, pages 178–185. ACM.

Whissell, J. S. and Clarke, C. L. A. (2013). Effective measures for inter-document similarity. In *CIKM'13*, ACM, pages 1361–1370.

Yakout, M., Ganjam, K., Chakrabarti, K., and Chaudhuri, S. (2012). Infogather: Entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD'12*, pages 97–108. ACM.

Yin, X., Tan, W., and Liu, C. (2011). Facto: A fact lookup engine based on web tables. In *WWW'11*, pages 507–516.