A FRAMEWORK FOR ESTABLISHING SIMILARITY BETWEEN WEB TABLES

Filipe Roberto Silva *
*Universidade Federal de Santa Catarina **
*Florianópolis – SC - Brazil **

Ronaldo dos Santos Mello *
*Universidade Federal de Santa Catarina **
*Florianópolis – SC - Brazil **

ABSTRACT

The Web is a huge information source. Large amounts of data are published daily and great part of them is available as HTML tables. Some works have proposed approaches to extract and integrate Web tables' content in order to make it more accessible for human consumption. However, this is a complex task and still an open issue given that Web tables do not have a unique representation pattern. Besides, the use of synonyms and abbreviations become hard the comparison of tables' content. Given that, we propose a framework to determine similarity between Web tables which is able to deal with distinct structures and synonym terms. Related works do not deal, at the same time, with both problematics. Preliminary experiments show that our solution is promising.

KEYWORDS

Web tables, Matching, Synonyms, Similarity

1. INTRODUCTION

The Web is a huge information source. Large amounts of data are published daily and great part of them is available as HTML tables. Cafarella et al. (2008) show that from 14.1 billion of these tables extracted from the Web, 154 million of them hold high quality relational data, i.e., they comprise *relational tables* with useful data, in a certain knowledge domain. This volume is certainly higher today.

We consider here HTML tables characterized by tag `<table>` and, as any other table, they can have data, headers, lines and columns. Several works have been studied ways to extract and integrate Web tables' content in order to make it more accessible for human consumption [Sardi Mergen et al. 2010; Venetis et al. 2011; Embley et al. 2011, Lautert et al. 2013, Fan et al. 2013]. However, the lack of schema, patterns and metadata in Web makes hard to identify which tables are similar. Even that, the identification of similar Web tables has several benefits. It could be useful, for example, for data integration processes, for filling out incomplete tables using similar tables' data, and also for Web table search engines. However, this is a complex task and still an open issue topic in the literature.

Crestan and Pantel (2011) estimate that 88% of HTML tables are used to provide structure to Web pages and do not have relational data. In an information retrieval process, this kind of table should be filtered out, since it does not hold relevant data. The tables indicated in Figure 1 by dashed rectangles are examples, i.e., their purpose is to properly display HTML elements at screen.

Besides these so called *layout tables*, relational useful tables do not have a unique representation pattern. This is one of the main problems for their correct retrieval, since it is necessary to detect where and how data are presented. Figure 2 shows some examples. Table in Figure 2(a) has tuples in horizontal orientation, while table in Figure 2(b) holds tuples in vertical orientation. Lautert et al. (2013) present a study about different structures of Web tables and classify them in 3 primary groups and 5 secondary groups. More details are given in Section 2.

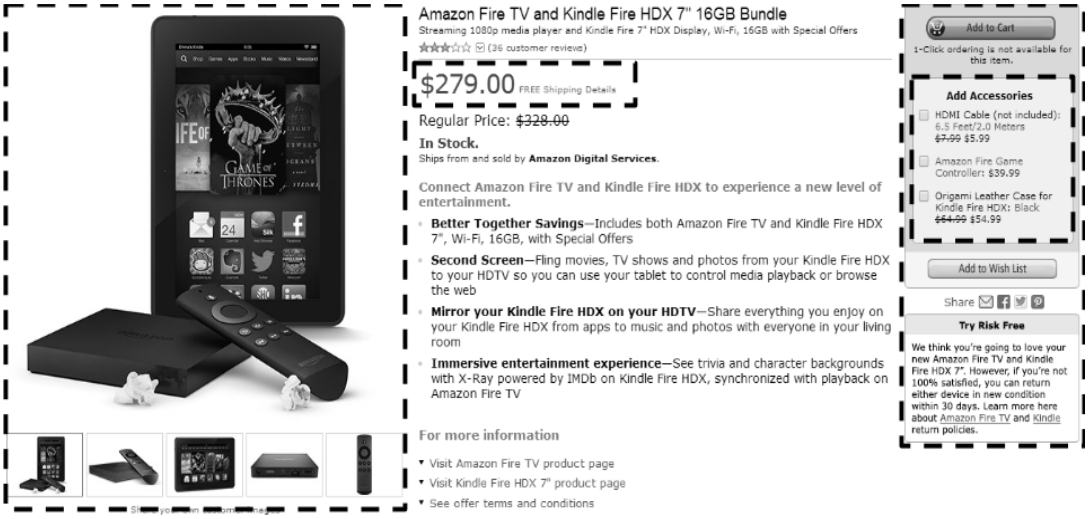


Figure 1. Irrelevant tables in a Web page.

Year	Title
1925	*The Freshman*
1931	*Maker of Men*
1932	*Horse Feathers*

(a) Horizontal table

Robert De Niro	
Born	August 17, 1943 New York, NY
Nationality	American
Occupation	Actor and director

(b) Vertical table

Figure 2. Tables with different structures [Lautert et al. 2013].

Despite the challenge of dealing with distinct representations, the use of synonyms and abbreviations makes also difficult the comparison of tables' content. One example is two tables with car descriptions, where one of them has a column "make" and other has a column "manufacture". Both have the same schema, but with different words.

Based on this problematic, we propose an approach to determine a similarity score between Web tables able to deal with distinct table structures and synonym terms. Related works do not deal, at the same time, with all existing Web table representations and synonym treatment. This is the main contribution of this work. The intention is to define a simple and effective strategy to compare Web tables, using, for this purpose, string similarity metrics available in the literature, and their combination and/or adaptation. We consider two Web tables as similar if they hold data in a same knowledge domain, like Tables Table 1 and Table 2, which maintain information about movies.

Table 1. Movies from January 2014

Title	Studio	Genre
The Lego Movie	Warner Bros.	Action, comedy
Barefoot	Roadside Attractions	Romantic comedy, drama
The Monuments Men	Columbia Pictures / 20th Century Fox	Drama, war

Table 2. Movies from February 2014

Name	Distributor	Genre
Jamesy Boy	Phase 4 Films	Crime, Drama
Paranormal Activity: The Marked Ones	Paramount Pictures	Horror
Dumbbells	GoDigital	Comedy

The rest of this paper is organized as follows. Section 2 is dedicated to related work. Section 3 details our approach, which is called *WTMatcher*. Section 4 presents an experimental evaluation and Section 5 is dedicated to the conclusion.

2. RELATED WORK

This section presents a brief description of recent works related to Web table matching.

Lautert el al. (2013) propose a Web table taxonomy. The work uses a primary classification proposed by Crestan and Pantel (2011) presented in Figure 3(a). This classification is mutually exclusive and divides primarily into relational tables and layout tables. Layout tables are organized as follows:

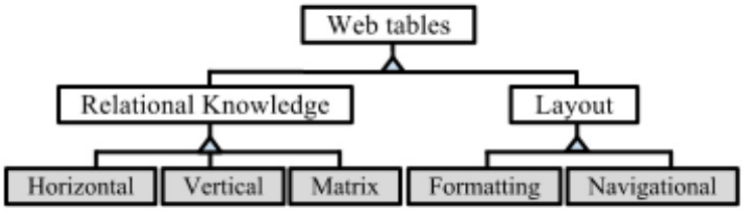
- **Navigational:** tables used to navigate the Web site;
- **Formatting**: tables that position the elements in Web page.

On the other hand, relational tables are organized as follows:

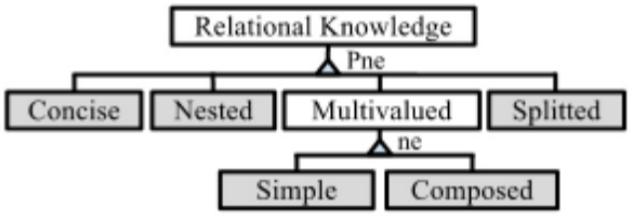
- **Vertical**: tables in which tuples are disposed vertically;
- **Horizontal**: tables in which tuples are disposed horizontally;
- **Matrix**: tables where each value that are not in the first row/column is associated to a pair of attribute values presented in row header and column header.

In addition to this primary classification, the work proposes a secondary classification, not mutually exclusive, that better specifies relational tables. This classification, shown in Figure 3(b), defines the following groups:

- **Concise**: tables with occurrence of merged cells;
- **Nested**: tables with inner tables;
- **Splitted**: tables that, for layout purposes, are splitted vertically or horizontally, and their pieces are disposed side by side or one below the other;
- **Simple Multivalued:** tables with multivalued cells of the same domain;
- **Composed Multivalued:** tables with multivalued cells of multiple domains.



(a) Primary classification



(b) Secondary Classification

Figure 3. Web table taxonomy [Lautert et al. 2013]

Fan et al. (2013) present a semi supervised system to recover Web tables' concepts. The purpose is to discover these concepts and use them to compare Web tables. This process is the same as to recover tables' schema and perform schema matching.

To find the concepts from a group of tables, they use the Freebase knowledge base [Bollacker et al. 2008], and associates concepts from the knowledge base to each column of the tables. The system creates a bipartite graph that keeps these relationships. The graphs' edges have weights obtained from comparisons between columns' values and concepts' instances. The system analyses the difficulty degree to discover the concepts of each column from these weights. Columns with more than one similar weight are harder to be associated to their real concept. In these cases, human intervention is needed.

Once discovered the concepts from each table, it is possible to determine which tables are similar. For this, they just compare the tables columns' concepts. A match is generated when two columns have the same concept. The work does not give details about the number of matchings necessary to determine if two tables are similar.

Pawlik and Augsten (2011) propose RTED, an algorithm to compute tree edit distances. The distance is calculated by the sequence of editing operations with minor cost that transforms one tree into another. The following operations are considered:

- Node exclusion;
- Node insertion;
- Node label renaming.

Each operation receives a cost and the edit distance is the sum of these costs to transform one tree into another. Their algorithm compares general labeled trees and can be adapted to compare HTML trees that represent tables. However, it just compares their structures and do not consider their contents.

Sarma et al. (2012) propose a framework to compare Web tables. The purpose is to find out candidate tables to join and union. The work uses three knowledge bases to label tables: Freebase, WebIsA [Venetis et al. 2011] and its own base. These labels and the Web tables' data are compared to identify which tables are similar.

Table 1 shows a comparison of the approaches. It includes our proposed solution, which is called *WTMatcher*. Lautert el al. (2013) is not included here because it does not make web tables matching. It was considered only as a basis to identify Web tables heterogeneity. As we can see, most of the approaches deals only with horizontal Web tables. The expection is the work of Pawlik and Augsten (2011), but it does not have synonym treatment.

Table 3. Comparison of approaches

Work	Kind of processed tables	Processing	Synonym Treatment	Strategy
[Fan et al. 2013]	Horizontal	Semi supervised	Yes	Knowledge base Similarity functions
[Das Sarma et al. 2013]	Horizontal	Automatic	Yes	Knowledge base Similarity functions
[Pawlik and Augsten 2011]	Any type	Automatic	No	TED
WTMatcher	Any type	Automatic	Yes	Knowledge base Similarity functions

The limitations of these works motivate the development of our approach. The intention is to propose a solution to compare Web tables that considers synonyms and, in addition, be able to deal with heterogeneous Web tables' structures, according to Lautert's taxonomy [Lautert et al. 2013], during the matching process. Our approach is detailed in the following.

3. WTMATCHER

Our proposed solution is called *WTMatcher* (*Web Table Matcher*). WTMatcher is a Web table matching process that determines their similarity considering distinct structures and synonyms. A general overview of the process is shown in Figure 4. The system receives two HTML tables as input. They are processed in order to find out their main structures, as headers and data (*Structure Recovery* component). After that, tables are compared using a Web table matching algorithm, which is the focus of this work (*Matching* component). This matching algorithm works with a knowledge base (KB). We intend to use available and domain organized KBs like Freebase [Bollacker et al. 2008] and/or WordNet [Miller 1995]. At the end, a similarity score is generated. This score is a value between 0 and 1. As nearer to 1 the score is, more similar the tables are.

The process considers Web tables already extracted, identified and classified. They are extracted by a crawler proposed by Sheidt and Dorneles (2013). It extracts Web tables from a HTML page and uses some

heuristics like the number of lines or columns, the position of the headers and the presence of merged cells to classify tables according to the taxonomy proposed by Lautert et al. (2013).

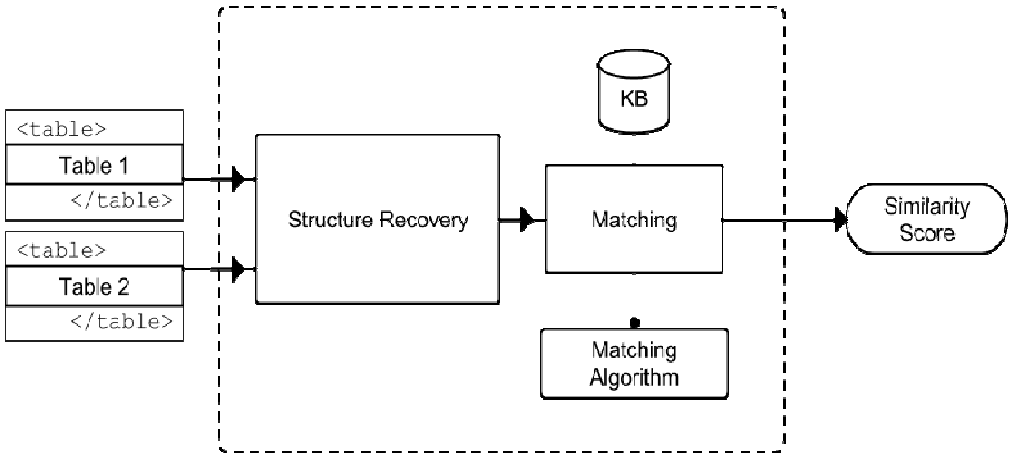


Figure 4. WTMatcher Architecture

By the time of this paper writing, we have developed and evaluated four table matching algorithms: *SimpleMatcher*, *SimpleMatcherPlus*, *WordNetMatcher* and *PureWordNetMatcher*. Their strategies compare pure HTML tables. They have basically two principles in mind: select data and attributes using simple heuristics, and perform string matching. These algorithms do not consider Web tables' classification yet. This treatment will be focus of future work. These algorithms are detailed in Subsections 3.1, 3.2 and 3.3.

3.1 SimpleMatcher

The SimpleMatcher algorithm basically compares headers and data using tags `<th>`, `<td>` and `<caption>` to detect tables' headers, data and title respectively. The intention of this detection using tags is to compare tables independently from their logic structure (vertical, matrix and so on). We also intend here to verify the viability of matching headers and data in this simple way.

The algorithm implementation uses the Java library *JSoup* [Hedley 2009] that allows to select accurately the HTML tags using CSS selectors. By using it, data sets are extracted from HTML tables. In order to compare these sets, we use the *SubSetSim* metric [Dorneles et al. 2004] that compares value sets. SubSetSim uses, in turn, string similarity metrics. Therefore, we have studied some known metrics for this purpose, as described in Section 3.4.

Equation 1 express the main similarity formula adopted by SimpleMatcher algorithm, where CS, HS and VS denote the Caption, Header and Value similarity, respectively, and CW, HW and VW their weights, where $\mathrm{CW} + \mathrm{HW} + \mathrm{VW} = 1$. In our experiments, these weights are all equal to 0.333.

$$SimpleMatcher = (CS * CW) + (HS * HW) + (VS * VW) \quad (1)$$

3.2 SimpleMatcherPlus

SimpleMatcherPlus is an extension of SimpleMatcher algorithm. It considers tags `<b>`, `<h1>` and `<h2>` for headers detection and also adds the matching of the attributes present in HTML tags, i.e. the attributes `class`, `id` and any other present attribute. Equation 2 shows the main similarity formula adopted by this algorithm, where `AS` is the attributes similarity and `AW` its weight. In our experiments, the weights for this algorithm are all equal to 0.25. In this first version, it compares only the attributes of the tag `<table>`.

$$SimpleMatcherPlus = (CS * CW) + (HS * HW) + (VS * VW) + (AS * AW) \quad (2)$$

3.3 WordNetMatcher

As stated before, we intend to deal with synonyms during the matching process. Thus, we used the lexical database WordNet [Miller 1995]. It contains English words, their synonyms and other semantic relations. We defined two algorithms based on WordNet: *PureWordNetMatcher* and *WordNetMatcher*. As we want to discover Web tables in the same domain, we focused on the WordNet matching in table headers. In this way, we can identify tables with similar schema.

The main similarity formula considered by both algorithms can be seen in Equations 3 and 4, where $WordNet(H_{t1}, H_{t2})$ is the header similarity between tables t_1 and t_2 using WordNet, and $SubSetSim(V_{t1}, V_{t2})$ is the value similarity between tables t_1 and t_2 using SubSetSim metric. PureWordNetMatcher (Eq 3) compares just table headers and uses just WordNet. Because of this, it does not need any external string similarity metric. WordNetMatcher (Eq 4) compares table values in addition to headers, so it uses SubSetSim.

$$PureWordNetMatcher = WordNet(H_{t1}, H_{t2}) \tag{3}$$

$$WordNetMatcher = (WordNet(H_{t1}, H_{t2}) * 0.5) + (SubSetSim(V_{t1}, V_{t2}) * 0.5) \tag{4}$$

3.4 Similarity Metrics

SubSetSim is the base matching function used by WTMatcher. It is represented by Equation 5, where $\in_p$ and $\in_d$ are value sets; n and m are the size of the sets $\in_p$ and $\in_d$, respectively. The function calculates the similarity between values, comparing all of them each other, and sums the max similarities. The sim function is a string similarity metric and can be changed properly. Because of this, we investigated string similarity metrics contained in *SimMetrics* library [Chapman 2006]. They were evaluated to determine the most effectives to compare Web tables. Observing their strengths and weaknesses, as well as the way they compare strings, we selected 7 metrics: *Cosine, Monge Elkan, Chapman Length, Euclidean Distance, Jaro Winkler, Jaccard,* and *Soundex*.

To evaluate which metrics are the best for our purpose, we have conducted some experiments using each one of them. These experiments are described in Section 4.

$$SubSetSim(\epsilon_p, \epsilon_d) = \frac{\sum \epsilon_p^i.\eta = \epsilon_d^j.\eta^{(max(sim(\epsilon_p^i, [\epsilon_d^1, \ldots, \epsilon_d^m])))}}{max(m, n)} \tag{5}$$

4. EXPERIMENTS

This Section presents preliminary experiments using the proposed algorithms, as well as a comparison between the chosen string similarity metrics. The experiments were executed over two sets of Web tables from general domains. The first set contains 450 Web tables extracted from the Web site of our research group laboratory. The other set contains 1000 Web tables extracted from Wikipedia.

For the experiments, we first analyzed the table sets manually to identify which tables are, in fact, similar to each other. Then, the automatic tests compare all tables to each other using the proposed algorithms, in combination to string similarity metrics, in order to obtain the similarity scores. We consider the similarity score threshold as 0.7. It was the better setting based on initial tests.

First of all, we ran SimpleMatcherPlus algorithm over the first table set. We tested this algorithm with several string similarity metrics. Table 4 shows the results in terms of Precision, Recall and F-Measure, being the last one a balance between Precision and Recall. All metrics had obtained good results in terms of F-Measure, with all values over 0.72. The Cosine metric got the best result in terms of F-Measure. Jaccard and Euclidean Distance also obtained high F-Measure values. It means that they found a large amount of truly

similar tables and did not return many false similar tables. Monge Elkan and Soundex had generated the worse results, mainly in terms of Precision. It means that they matched false similar tables.

Table 4. Experiment with string similarity metrics\

Metric	*Precision*	*Recall*	*F-Measure*
Monge Elkan	0.58	0.82	0.72
Cosine	0.95	0.76	0.91
Chapman Length	0.91	0.68	0.84
Euclidean Distance	0.97	0.69	0.87
Jaro Winkler	0.67	0.83	0.81
Jaccard	0.95	0.71	0.88
Soundex	0.58	0.81	0.75

We also performed some experiments to compare the algorithms. They were executed over the second table set, and we used the three metrics with best results in the previous experiments: Cosine, Euclidean and Jaccard. We also consider the metric with the worse result, in this case, Monge Elkan. The intention is to analyze if the results from the previous experiments are maintained. As we had access to its source code, we also considered the RTED algorithm [Pawlik and Ausgen 2011] as the baseline in these experiments.

Table 5 shows the experiments' results in terms of Precision, Recall and F-Measure. The Monge Elkan and Euclidean Distance metrics did not provide good results in terms of Precision. The SimpleMatcher and RTED algorithms did not generate good results in terms of Recall, what means that they did not match all expected tables. On the other hand, WordNetMatcher and SimpleMatcherPlus got good results in terms of F-Measure. WordNetMatcher had obtained the best general scores, while SimpleMatcherPlus got a high Precision value and low Recall. The low Recall for RTED was expected, as it just compares tables' structures. Thus, it just matches tables with the same representation, ignoring tables in the same domain but with distinct structures.

Table 5. Experiment with the proposed algorithms and comparison with a baseline

Algorithm	*Metric*	*Precision*	*Recall*	*F-Measure*
PureWordNetMatcher	-	0.572	0.494	0.700
WordNetMatcher	Cosine	0.493	0.560	0.672
WordNetMatcher	Euclidean	0.111	0.660	0.299
WordNetMatcher	Jaccard	0.502	0.548	0.674
WordNetMatcher	Monge Elkan	0.099	0.674	0.267
SimpleMatcher	Cosine	0.888	0.086	0.536
SimpleMatcher	Euclidean	0.624	0.270	0.470
SimpleMatcher	Jaccard	0.887	0.059	0.517
SimpleMatcher	Monge Elkan	0.503	0.361	0.526
SimpleMatcherPlus	Cosine	0.760	0.396	0.661
SimpleMatcherPlus	Euclidean	0.138	0.604	0.340
SimpleMatcherPlus	Jaccard	0.770	0.367	0.662
SimpleMatcherPlus	Monge Elkan	0.100	0.697	0.249
RTED	-	0.685	0.080	0.481

5. CONCLUSION

This paper presents and evaluates an approach for matching Web tables called *WTMacther*. Different from related work, it considers, in a joint way, tables with heterogeneous representations and synonyms treatment.

Preliminary experiments highlighted some good results, especially for WordNetMatcher using Cosine and Jaccard metrics. However, the considered amount of data was not so large. In the future, we intend to evaluate the approach over a larger data set to verify if the good results remain. We also intend to create better heuristics to detect Web tables' schema, possibly based on Sarma's work [Das Sarma et al. 2012] that uses a KB for this purpose. By using a KB, we also intend to define another matching algorithm that combines WordNet for headers comparison, and other KB for values comparison. This algorithm will compare headers as we have already made in WordNetMatcher, but instead of comparing table values directly, we may recover their concepts and compare it, like in Fan et al. (2013) work. We expect that this combination will help us to deal with the problem of matching tables with same domains, but not necessarily with same instances.

Regarding heterogeneous structures, we intend to create a mapping between distinct Web tables' structures, making it possible to compare, for example, a cell with a nested table, or a value with a multivalued cell.

It is also important to observe that the string similarity metric results are not absolute for all domains. Depending on the domain, results may change, as happened to Euclidean Distance from one experiment to another. The use of a KB could also help dealing with that problem, given that we could consider more KB concepts than tables' values.

REFERENCES

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, New York, NY, USA, pp. 1247–1250. ACM.

Cafarella, M. J., A. Halevy, D. Z.Wang, E.Wu, and Y. Zhang (2008, August). Webtables: Exploring the power of tables on the web. Proc. VLDB Endow. 1(1), 538–549.

Chapman, S. (2006). Simmetrics. URL *http://sourceforge. net/projects/simmetrics/. UK Sheffield University*

funded by (AKT) an IRC sponsored by EPSRC, grant number GR N 15764.

Crestan, E. and P. Pantel (2011).Web-scale table census and classification. *In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, New York, NY, USA, pp. 545–554. ACM.

Das Sarma, A., L. Fang, N. Gupta, A. Halevy, H. Lee, F. Wu, R. Xin, and C. Yu (2012). Finding related tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, New York, NY, USA, pp. 817–828. ACM.

Dorneles, C. F., C. A. Heuser, A. E. N. Lima, A. S. da Silva, and E. S. de Moura (2004). Measuring similarity between collection of values. In *Proceedings of the 6th Annual ACM International Workshop on Web Information and Data Management*, WIDM '04, New York, NY, USA, pp. 56–63. ACM.

Embley, D.W., M. Krishnamoorthy, G. Nagy, and S. Seth (2011). Factoring web tables. In *Proceedings of the 24th international conference on Industrial engineering and other applications of applied intelligent systems*, IEA/AIE'11, Berlin, Heidelberg, pp. 253–263. Springer-Verlag.

Fan, J., M. Lu, B. C. Ooi, W.-C. Tan, and M. Zhang (2013). A hybrid machinecrowdsourcing system for matching web tables. Technical report, Technical Report.

Hedley, J. (2009). jsoup: Java html parser.

Jaccard, P. (1908). *Nouvelles recherches sur la distribution florale*.

Lautert, L. R., M. M. Scheidt, and C. F. Dorneles (2013, October). Web table taxonomy and formalization. SIGMOD Rec. 42(3), 28–33.

Miller, G. A. (1995, November). Wordnet: A lexical database for english. *Commun*. ACM 38(11), 39–41.

Pawlik, M. and N. Augsten (2011, December). Rted: A robust algorithm for the tree edit distance. *Proc*. VLDB Endow. 5(4), 334–345.

Sardi Mergen, S. L., J. Freire, and C. A. Heuser (2010). Indexing relations on the web. In *Proceedings of the 13th International Conference on Extending Database Technology*, EDBT '10, New York, NY, USA, pp. 430–440. ACM.

Scheidt, M. M. and C. F. Dorneles (2013). Ferramenta para extração de webtables e criação de scripts sql. Technical report.

Venetis, P., A. Halevy, J. Madhavan, M. Pas¸ca, W. Shen, F. Wu, G. Miao, and C. Wu (2011, June). Recovering semantics of tables on the web. *Proc*. VLDB Endow. 4(9),528–538.