

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/304055051>

Accurate fact harvesting from natural language text in wikipedia with Lector

Conference Paper · June 2016

DOI: 10.1145/2932194.2932203

CITATIONS

4

READS

160

3 authors, including:



Denilson Barbosa

University of Alberta

110 PUBLICATIONS 1,095 CITATIONS

[SEE PROFILE](#)



Paolo Merialdo

Università Degli Studi Roma Tre

110 PUBLICATIONS 2,263 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



REX - RDF Data Extraction [View project](#)



XML data management [View project](#)

Leveraging Wikipedia Table Schemas for Knowledge Graph Augmentation

Matteo Cannavicchio*
Roma Tre University
Rome, Italy
cannavicchio@uniroma3.it

Denilson Barbosa
University of Alberta
Edmonton, Canada
denilson@ualberta.ca

Lorenzo Ariemma
Roma Tre University
Rome, Italy
lor.ariemma@uniroma3.it

Paolo Merialdo
Roma Tre University
Rome, Italy
paolo.merialdo@uniroma3.it

ABSTRACT

General solutions to augment Knowledge Graphs (KGs) with facts extracted from Web tables aim to associate pairs of columns from the table with a KG relation based on the matches between pairs of entities in the table and facts in the KG. These approaches suffer from intrinsic limitations due to the incompleteness of the KGs. In this paper we investigate an alternative solution, which leverages the patterns that occur on the schemas of a large corpus of Wikipedia tables. Our experimental evaluation, which used DBpedia as reference KG, demonstrates the advantages of our approach over state-of-the-art solutions and reveals that we can extract more than 1.7M of facts with an estimated accuracy of 0.81 even from tables that do not expose any fact on the KG.

ACM Reference Format:

Matteo Cannavicchio, Lorenzo Ariemma, Denilson Barbosa, and Paolo Merialdo. 2018. Leveraging Wikipedia Table Schemas for Knowledge Graph Augmentation. In *WebDB'18: 21st International Workshop on the Web and Databases*, June 10, 2018, Houston, TX, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3201463.3201468>

1 INTRODUCTION

Knowledge Graphs (KGs) have received much attention in the last decade due to their ability to provide background knowledge that enables emerging applications such as semantic search, question answering, recommendation systems [19], to mention just a few. Wikipedia has been leveraged as the primary source of information

*Research done with the support of Diffbot during a collaboration with the Diffbot Knowledge Graph team.


Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WebDB'18, June 10, 2018, Houston, TX, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

ACM ISBN 978-1-4503-5648-0/18/06...\$15.00

<https://doi.org/10.1145/3201463.3201468>



...	Title	Directed by	Written by	...
	"Homer the Whopper"	Lance Kramer	Seth Rogen	
	"Bart Gets a 'Z'"	Mark Kirkland	Matt Selman	
	"The Great Wife Hope"	Matthew Faughnan	Carolyn Omine	
	"Boy Meets Curl"	Chuck Sheetz	Rob LaZebnik	
	"The Color Yellow"	Raymond S. Persi	Billy Kimball	

Figure 1: Fragments of a Wikipedia table describing TV series episodes. DBpedia relations between the entities in the two columns are ① $\langle\text{dbo:director}\rangle$ and ② $\langle\text{dbo:writer}\rangle$.

for building and populating automatically large-scale KGs such as DBpedia [2] and YAGO [17]. By and large, such approaches to extract knowledge from Wikipedia leverage information stored in the info-boxes that are present in many articles [1, 2, 10]. Other solutions adopt distant supervision and NLP techniques to extract facts from the body of the articles by exploiting the regularities of textual patterns that frequently occur in Wikipedia texts [5, 10, 11].

Another source of factual knowledge that has not been explored as much are *tables* in Wikipedia articles. Indeed, the contents of many pairs of columns that are present in Wikipedia tables represent a natural source of facts for binary KG relations. As an example, consider the (portion of the) table in Fig. 1, which describes episodes of the popular TV series “The Simpsons”.¹ Two pairs of columns from this table suitably represent DBpedia relations, and the cells appearing on the same row correspond to subjects and objects of facts that can be used to populate such relations.

Identifying which KG relations hold between pairs of columns of a given table is a fundamental step in many solution that aim at extracting knowledge from Web tables. Current approaches are based on *local evidence* [9, 12–14]. Given one table, the overall idea is to link entities in the table with entities in the KG, and then to associate a pair of columns of the table with one relation in the KG if the two columns contain a sufficient number of pairs that correspond to facts of that relation. Unfortunately, these approaches fail whenever the overlap between entities in the table and those related in the KG is not sufficiently high. The example in Fig. 1 illustrates this problem: no pair of entities from columns

¹https://wikipedia.org/wiki/List_of_The_Simpsons_episodes#Episodes

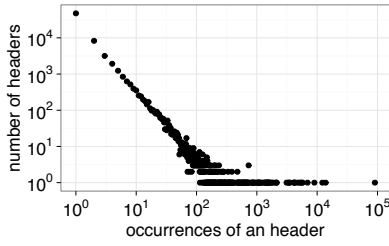


Figure 2: Distribution of headers in Wikipedia tables

Title and Directed by is present in a DBpedia relation, though they really represent facts of the $\langle \text{dbo:director} \rangle$ relation. This limitation is due to the notorious incompleteness of even the most well known KGs [4].

We describe MENTOR, a system that explores a novel solution to support KG augmentation with facts from Wikipedia tables that overcomes some of the challenges due to KG incompleteness. Our approach is inspired by the observation that Wikipedia editorial policies nudge editors to adopt very homogeneous headers on tables that describe similar concepts. For example, the header of the table in Fig. 1 is present in 1,680 distinct tables, each representing the same concepts. We analyzed the headers of a collection of 550K tables from a recent Wikipedia dump to verify this hypothesis, revealing that, excluding columns that are unlikely to describe KG entities (e.g., those containing long textual values), the fraction of distinct headers corresponds to only 10% of the total number of headers. Fig. 2 shows the distribution of the headers according to the frequency with which they are used. Observe that although there are 47,050 headers that occur just once (the point on the top left corner), there is a long tail of headers with more than 10 occurrences. The most frequent header ([party, candidate, votes]) has 90,511 occurrences (the point on the bottom right corner).

Based on this observation we consider the opportunity to identify the KG relations that hold between the columns of the tables by leveraging *global evidence* from all the tables in Wikipedia. We start by matching every column, of every table, with a KG entity type. Then, for each *pairs of columns* in the same table we exploit known KG facts to associate their schema, consisting of the header titles and KG types of the two columns, with a KG relation. The schemas can be seen as *patterns* that involve both lexical information (headers) and semantic information (types). Then, we use all the associations that we create locally between schemas and KG relations to derive a set of global associations that frequently occur over the whole corpus of tables. Finally, we leverage these global associations to annotate other (pairs of columns in other) tables, yielding new facts. Consider again our running example; we seek to leverage the many column pairs with the same types (TelevisionEpisode and Director) and headers (Title and Directed by) in other Wikipedia tables to annotate the table of Fig. 1.

It is worth observing that some local associations between table schemas and KG relations could arise by chance, introducing noise. For example, if the director and the writer of many TV episodes appearing in one table were married, and these relationships were the KG, we could end up associating the corresponding schema

School	Team	City	State	Primary Conference
Abilene Christian University	Wildcats	Abilene	Texas	Southland Conference
University of Akron	Zips	Akron	Ohio	Mid-American Conference
University of Denver	Pioneers	Denver	Colorado	The Summit League
Stanford University	Cardinal	Palo Alto	California	Pac-12 Conference
Pepperdine University	Waves	Malibu	California	West Coast Conference
University of San Francisco	Dons	San Francisco	California	West Coast Conference
Stony Brooke University	Seawolves	Stony Brook	New York	American East Conference

Figure 3: A Wikipedia table with many DBpedia relations between its pairs of columns: ① $\langle \text{dbo:team} \rangle$, ② and ⑤ $\langle \text{dbo:city} \rangle$, ③ and ⑧ $\langle \text{dbo:state} \rangle$, ④ $\langle \text{dbo:conference} \rangle$, ⑥ $\langle \text{dbo:location} \rangle$, ⑦ $\langle \text{dbo:athletics} \rangle$, ⑨ $\langle \text{dbo:headquarter} \rangle$.

with a wrong relation. Indeed, the same issue occurs also for approaches based on local evidence. This is alleviated in some methods, e.g. [12, 14], that consider only pairs of columns that involve the so-called *subject* column (i.e. the one corresponding to the main concept in the table). However, doing so is sub-optimal since large tables offer many pairs of columns that represent actual relations. Fig. 3 shows an example involving athletic departments of various universities² where our method can annotate five pairs of columns that do not involve the subject column (School) with meaningful DBpedia relations. As we rely on global evidence, we are able to address the drawbacks of relations that occur by chance without any limit on the composition of the pair of columns, thus exploiting the richness of large Wikipedia tables.

2 OVERVIEW OF THE MENTOR APPROACH

We now give an overview of MENTOR, describing DBpedia, which is our reference KG, and the collection of Wikipedia tables.

2.1 DBpedia

DBpedia is one of the largest freely available KGs. Its contents are essentially organized in entities, types, relations and facts.

Entities are identified by a unique (human readable) identifier and are associated with one fine-grained type. The set of types is defined in the DBpedia Ontology, a manually-curated ontology (based on OWL) that contains 685 different types. For example, the triple $\langle \text{Michelle_MacLaren} \rangle \langle \text{rdf:type} \rangle [\text{Person}]$ expresses that Michelle MacLaren is an entity of type person. As DBpedia is derived from Wikipedia info-boxes, every DBpedia entity is also nicely linked to the corresponding Wikipedia article.

In DBpedia, as in other KGs, relations are used to connect entities with other entities (or with text literals) and are defined with a specific schema expressed by a pair of types. For example, an entity of type [TelevisionEpisode] can be related to an entity of type [Person] via the relation $\langle \text{dbo:director} \rangle$. Finally, facts are used to define instances of relations with the relative pair of entities. For

²https://wikipedia.org/wiki/List_of_NCAA_Division_II_institutions

example, the triple $\langle \text{Michelle_MacLaren} \rangle \langle \text{dbo:director} \rangle \langle \text{Thirty-Eight_Snub} \rangle$ describes that Michelle MacLaren is the director of Thirty-Eight Snub (an episode of the TV series Breaking Bad).

2.2 Wikipedia Tables

With the 2016-10 extraction DBpedia released a corpus of HTML tables parsed from Wikipedia³. The corpus contains 2.2M tables (classified as *wikitable*⁴) extracted from the (English version) of Wikipedia articles of October 2016.

As we concentrate on *vertical relational tables*, i.e. tables with a horizontal header and composed by two or more columns, we operate a filtering process to discard the ones that do not comply this features. We also eliminate columns with numerical values and dates, and we clean them from vertical and horizontal spans.

To detect the entities we rely on the wiki-links that are present in the column cells (i.e., the HTML links used to highlight concepts and link them to Wikipedia articles). In this way we can easily associate the columns with a DBpedia type leveraging the links between DBpedia entities and Wikipedia articles.

Filtering and type-enriching processes are detailed in Section 3.

2.3 The MENTOR approach

We use the collection of vertical tables obtained from the previous filtering to associate DBpedia relations to the schemas and extract new facts. Fig. 4 illustrates an overview of our approach.

The tables are organized in pairs of columns from the same table, which hereafter we call *bi-columns*. Every bi-column exposes a specific *schema*, i.e., the ordered pair of header title and entity type of its columns. Based on local evidence, the bi-columns are associated with DBpedia relations, leveraging the presence of some pairs of entities that match facts of the same relation in the KG. Then, the global evidence is obtained by grouping all the local associations by relations and associating the schemas of the relative bi-columns with a ranked set of DBpedia relations. This ranking is used again to associate DBpedia relations to the bi-columns, extracting new facts even in the cases of bi-columns having low local evidences because of the KG incompleteness. With this approach we can extract 1.7M of facts that are new for DBpedia⁵.

Section 4 describes technical details of the approach and Section 5 reports results of our experimental evaluation.

3 VERTICAL RELATIONAL TABLES

We process all the *wikitable* tables to obtain the vertical relational tables that are used by MENTOR. These expose multiple columns containing DBpedia entities and each of them is labeled with a specific DBpedia type. Here we describe how we obtain them from the original dump.

Filtering tables. Since Wikipedia is intended for human readers, tables in its pages are used for many purposes. The goal of

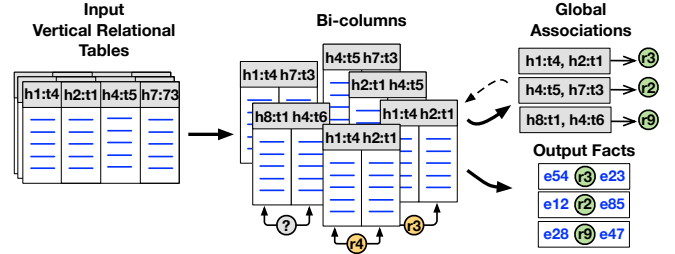


Figure 4: Overview of MENTOR approach. (h1, h2,...) denote column headers; (t1, t2,...) denote DBpedia types; (r1, r2,...) denote DBpedia relations and (e1, e2,...) are DBpedia entities.

the filtering step is to detect and discard tabular structures that are not used to organize data in vertical relational tables. In particular, we discard tables that represent matrices and nested tables. The former usually model ternary relationships among the content of every cell with the row and column headers. Therefore, to detect these structures we consider the presence of header cells both on the first row and on the first column of the table. The latter in Wikipedia are usually adopted for creating sophisticated layouts with aesthetic objectives. They can be simply detected by considering tables whose cells contain other tables.

Normalization and cleaning. The selected tables can contain cells that span horizontally or vertically. Therefore we normalize them, by suitably duplicating the contents of the related columns and rows, respectively. As an exception, we eliminate columns and rows that span on the whole table, since they usually represent captions or descriptions of horizontal portions of the table.

After normalizing the tables we clean their internal contents. As we are interested only on cells containing named entities (which can be linked to DBpedia), we consider only the first wiki-link that is present in each cell, thus filtering out text, dates or numerical values, embedded contents (e.g., images and icons), external references (e.g., footnotes, links to external resources). Since, sometimes editors avoid to link all the instances of an entity that appear in many distinct cells of the same table, we recover the missing wiki-links of these entities by looking and replacing possible occurrences of anchor texts already used for other entities in the same column.

DBpedia type assignment. As a last step, we filter out columns that contain heterogeneous information, as they are unlikely to participate to a KG relation. To this end, we rely on the *type coverage* of a column, as follows. Let T denote the set of entity types in DBpedia. The coverage of a type $t \in T$ on column C is defined as: $cov(t, C) = \frac{count(t, C)}{|C|}$, where $count(t, C)$ is the number of entities in C that are related with the type t . We use the coverage of the types to eliminate noisy columns, whose entity types are not homogeneous. In particular, we discard a column C if there is no type t such as $cov(t, C)$ is greater than a threshold (in our experiments we use 0.8⁶). Finally, we associate every column the DBpedia type with maximum coverage.

⁶To avoid a penalization for large tables, we keep columns with a minimum amount of entities (10) that have the same type.

³<http://downloads.dbpedia.org/2016-10/core-i18n/en/>

⁴The corpus offers also other kinds of tables, such as, those extracted from info-boxes or tables obtained from paragraph outlines, which are out of the scope of our system.

⁵All the data used, including the vertical tables and the bi-columns, are available for download at: <http://dx.doi.org/10.7939/DVN/F36TGC>

4 LEVERAGING GLOBAL EVIDENCE

From the original corpus of Wikipedia tables, the previous phase returns 106K vertical relational tables. The columns of these tables are then organized in bi-columns, each exposing a schema as defined in Sec. 2.3. In total, we obtain 224K bi-columns that expose around 40K different schemas. Here we describe the method that we use to associate the schemas to the DBpedia relations that they represent, which is based on evaluating the global evidence from all the bi-columns.

4.1 Associating Bi-columns to Relations

We use the facts already in DBpedia to associate a bi-column with a relation, in case there is enough evidence from its entity pairs. Given a bi-column B composed of a list of DBpedia entity pairs, let $\text{count}(r_j, B)$ denote the number of entity pairs from B that are related with a relation r in DBpedia. We score the *local evidence* of the relation r on B , denoted $\text{local}(r, B)$, as the ratio of pairs that are related with the relation r :

$$\text{local}(r, B) = \frac{\text{count}(r, B)}{\sum_{r_t \in R^+} \text{count}(r_t, B)} \quad (1)$$

Since entity pairs can be involved in multiple facts, so with different relations, we normalize using all the relations in DBpedia: the denominator represents the sum of all the instances of relations including a further relation used to consider pairs that are not linked in the graph (R^+), in which we do not know if a relation holds.

The local evidence represents a rough estimate of the probability that the bi-column effectively describes the relation. Thus, for each bi-column we rank all the relations according to their local evidence, associating it with the one that is ranked as first, i.e. the DBpedia relation with the highest local evidence. We use a cutoff to limit the possibility to associate relations by chance ($\text{local} < 0.25$), excluding the bi-columns having low evidence in this step.

4.2 Associating Schemas to Relations

Having associated bi-columns with relations according to their local evidence, we can now estimate the relations represented by each schema considering its global evidence on the corpus.

Intuitively, we can be confident of a schema describing a certain relation if it frequently occurs on bi-columns that likely represent such relation. Thus, we estimate the confidence of a schema with a relation by combining multiple local evidences from all the bi-columns that expose that schema. We calculate the global score of schema s and a relation r_j as:

$$\text{global}(r_j, s) = 1 - \prod_{b \in \text{Bi}[s]} (1 - \text{local}(r_j, bi)) \quad (2)$$

where $\text{Bi}[s]$ represents all the bi-columns that expose the schema s . Essentially, a relation r_j scores high with s if it has high local evidence with any of the bi-columns in $\text{Bi}[s]$. The same effect can be obtained in case many bi-columns in $\text{Bi}[s]$ are locally associated with r_j , although with low values of evidence. With this approach we are able to associate 9.6K distinct schemas to the DBpedia relations that they represent.

4.3 Associating Relations to Bi-columns

Finally, we associate KG relations, again, to all the bi-columns of the collection and extract new facts from them. Differently from the local assignments, described in Sec. 4.1, here we do not consider their local evidence but rely only on the schemas and their global associations with DBpedia relations.

In order to assign a relation for a given bi-column, we check if its schema matches exactly one of the schema associated with the relations. Two schemas match if the two pairs of headers and the two pairs of types agree (the order of the headers is irrelevant). However, more sophisticated techniques are possible.

Associating bi-columns to relations allows to augment the KG with new facts, by using the relation and all the entity pairs of the bi-column. It is worth noting that using global evidences we can associate a DBpedia relation even to the bi-columns containing pairs of entities that exist in the KG but are not yet related, which could not be used by any approach that rely on local evidence only.

5 EXPERIMENTAL EVALUATION

We evaluate MENTOR considering the global associations that we can derive with the highest global scores. From the bi-columns extracted from our table corpus, we derive 9.6K unique schemas that we can be associated with KG relations using a certain global score (Eq. 2). The score reflects the strength of the association and ranges between 0 and 1: determining what is a good threshold to produce reliable associations is challenging. To address this issue, we adopt a practical data-driven solution, as follows.

We consider the set of bi-columns whose entity pairs do not match any facts in DBpedia (i.e., no local evidence at all). Those bi-columns correspond to almost the half of the collection (47%) and represent an interesting test-set: while some of them are derived from accidental associations of the columns, others express actual relations that could not be captured relying on the *local evidence* because of the incompleteness of the KG.

We use all the schemas associated to DBpedia relations to label all of them with a DBpedia relation and we evaluate the correctness of the relations assigned for different intervals of score. Namely, we consider three score intervals (0.25 – 0.50, 0.50 – 0.75, 0.75 – 1) and manually evaluate a random sample of 100 bi-columns from each interval. In the manual evaluation, we consider correct the DBpedia relation associated with a bi-column only if the relation is really valid (not by chance) for all the entity pairs of the bi-column. The accuracy is estimated for each interval using the Wilson score for $\alpha = 5\%$ ensuring to reach a significant confidence interval (below $\pm 0.1\%$). Table 1 reports the number of associations that we obtained in three intervals. Note that, the interval with the higher score contains almost half (4,6K out of 9,6K) of the schemas. As a result, we associate 10,1K of such bi-columns with a DBpedia relation and produce 154K facts that are new for DBpedia with an estimated accuracy of 0.71.

Most of the errors in this interval originate from a small number of associations between schemas and KG relations that occur by chance but are very frequent in the corpus. For example, the schema $\langle \text{Opponent}:[\text{SoccerClub}], \text{Venue}:[\text{Stadium}] \rangle$, found in bi-columns in tables describing soccer matches in a certain season, is associated with the relation $\langle \text{dbo:tenant} \rangle$. The score here has a

Score Interval	Schemas	Bi-columns	Accuracy
0.75-1.00	4,696	10.1 K	0.71 ± 0.09
0.50-0.75	2,590	1.2 K	0.43 ± 0.10
0.25-0.50	2,696	3.5 K	0.23 ± 0.08

Table 1: Estimated accuracy for different score intervals.

high value because the schema occurs many times with local evidence scores that are, on average, around 0.5. This is explained by the fact that in most leagues, teams play half of their matches in the stadium of their opponents, for whom the relation holds.

On the other hand, the associations obtained by low values of the score (0.25-0.50) are attributable to bi-columns that are associated completely random. For example, we observed that 30% of the errors in the lower score interval is due to the association of the schema $\langle \text{Cast:}[\text{Director}], \text{Person:}[\text{Person}] \rangle$ with the relation $\langle \text{dbo:spouse} \rangle$, which was derived from a single bi-column. Using these associations to add new facts on the KG would be very risky because some of these schemas appear very frequently.

5.1 Extraction of new facts

Given the above evaluation, we computed a complete extraction of new facts considering only the 4.6K associations from the higher interval (0.75-1.00). We predicted the relations for all the other 118K bi-columns which contain, even a small, local evidence with some DBpedia relation. Although some of them have been used to derive the associations, we still keep them to predict possible relations since we are now relying on the global evidence. Overall, we can predict a relation for 68.5K of such bi-columns, extracting 1.59M facts that are not present in DBpedia. We evaluate a sample of them in the next experiment.

5.2 Comparison with other approaches

We compare MENTOR with two state-of-the-art systems that are based on the local inference approach; namely, Wikitables [9] and T2K [12, 13]. They both use DBpedia as reference KG so we can easily compare the results on the task of predicting a relation for the pairs of columns of a given table. Since an independent ground-truth is missing, we execute MENTOR, T2K and Wikitables over a set of random bi-columns from the whole collection.

Wikitables. The primary goal of Wikitables is to extract a set of facts from a given Wikipedia table. The tool relies on *local evidence*, by looking for matches between pairs of cells and facts in the KG, but it also exploits a large set of features extracted from the table and its context. Similarly to our approach, it considers all the possible pairs of columns of the table and assumes that DBpedia entities are already detected in the cells. As a result, it provides the list of facts that can be extracted from the table. In order to facilitate the evaluation on a given bi-column we consider, as output of the tool, the relations that have been predicted for every entity pairs of the input bi-column. While there can be multiple relations predicted, we evaluate the answer as correct if at least one of the relations is correct for the bi-column.

All Bi-columns			
	Precision	Recall	F-measure
Mentor	0.81	0.63	0.71
Wikitables	0.74	0.65	0.69
Bi-columns with subject-column			
	Precision	Recall	F-measure
Mentor	0.82	0.68	0.74
T2K	0.78	0.51	0.61

Table 2: Evaluation against state-of-the-art tools.

T2K. The tool performs all the tasks necessary to understand Web tables, from linking the cells with the KG entities and associate a KG relation to the columns. However, T2K considers only bi-columns that include the subject-column of the table. Therefore, for a fair comparison with this tool, we use a random sample of bi-columns containing the subject-column (that we detect following the same heuristics of used by T2K) of the original table.

Evaluation. We pick a set of 1,000 random bi-columns from the whole collection and process them with MENTOR and with the other tools. Also in this case we have manually evaluated the quality of the relations predicted (thus excluding from the evaluation the bi-columns that were not predicted by any tool).

To compare the performances of the tools we consider precision, recall and F-measure. Precision is calculated as the ratio between the number of bi-columns predicted correctly and the total number of bi-columns predicted; recall is estimated using the same numerator but divided by the total number of bi-columns that can be predicted from the tools involved in the comparison. F-measure is the harmonic mean of precision and recall. Even in this case we provide the judges with the whole tables where the bi-columns are extracted, considering correct only the bi-columns that really represent the relation produced by the system (not by chance).

Results. We compare MENTOR versus Wikitables using 469 bi-columns where at least one of the two tools predicted a relation. Table 2 shows the results obtained by the tools. MENTOR performed better than Wikitables in terms of precision and F-measure, due to its ability to combine global evidence.

In the comparison against T2K, we evaluated 453 bi-columns as the ones involving a key-column and both the tool provided a result. The precision of T2K is close to MENTOR but its recall is lower. This is due to the limits in the entity linking step, since T2K does not rely on already disambiguated entities, in many case it fails to disambiguate correctly the columns.

6 RELATED WORK

A large body of research investigate on using tables for different information extraction tasks, such as retrieval/search [15, 16, 18] or for set expansion/tables completion [3, 20, 21]. Our focus is Knowledge Graph augmentation, which consists in completing an existing KG with new facts. The task can be considered a direct application of the Web tables understanding problem [6, 7, 9, 13, 22, 23] which aims to annotate a table with entities, types and relations

from a given KG. One of the earlier approach [7] consists in describing the dependencies among the components of a table and the KG using a probabilistic graphical model. At inference time they label cells with entities, columns with types and pairs of columns with KG relations relying on a joint inference (they used YAGO [17]). While the approach is based on local evidence it also relies on the schema: a similarity between headers and relation is considered one of the features of their graphical model.

As described in Sec. 5, one of the current state-of-the-approaches based on local evidence is proposed in [12, 13]. They model each table as a set of similar entities from DBpedia, each one described with multiple attributes. The approach consists in an iterative algorithm which assigns a class to the table, disambiguate the entities on each row and assign a relation for each column. While the approach captures also relations with literal values, they miss possible facts between intermediate pairs of columns. In [14], they experiment a *dictionary* of column headers and KG relations derived from matching a large collection of Web tables to DBpedia, which improves both precision and recall of the tool.

Our goal is similar to [9] where a large number of DBpedia facts have been extracted from Wikipedia tables relying on a combination of local evidence and a posterior classifier based on a large set of table’s features and used to eliminate noisy facts. We evaluate it and compare with MENTOR in the experiment section.

Headers of the columns are often involved as features for the entity disambiguation step in many other Web tables understanding approaches. For example, in [8] the candidate entities used for disambiguation are obtained considering the similarity between header and their types other than the text in the cell and in the whole row. In [22] the headers are considered “in-table” features, with other “out-table” features such as captions, web-page title, text in surrounding paragraphs, all involved to perform entity disambiguation against Freebase. Finally, the concept of global evidence is exploited in [20] to detect possible synonyms of tables headers from a large collection.

7 CONCLUSION

We described MENTOR, a system that investigates a novel solution to support KG augmentation from Wikipedia tables that overcomes the limitations due to the incompleteness of the KG. Our evaluation reveals that MENTOR produces over 1.7M facts that are new to DBpedia with an accuracy estimated around 0.81. Using a global evidence from the schemas, our method can improve current state-of-the-art approaches which rely on local evidence. It is worth observing that 10% of the facts have been extracted from tables with no evidence of any relation from the KG.

For future work, an inspection of preliminary results shows that filtering the headers involved in complex relationships lead to substantial improvements in accuracy. For this, evaluating functional dependencies between columns might help to avoid noisy associations. Another promising direction is to consider *semantic* completion of the schemas in order to allow some generalization.

ACKNOWLEDGEMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Diffbot.

REFERENCES

- [1] Alessio Palmero Aprosio, Claudio Giuliano, Alberto Lavelli. Automatic Expansion of DBpedia Exploiting Wikipedia Cross-Language Information. In ESWC, 397–411, 2013.
- [2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, 2009.
- [3] Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu, Yang Zhang. WebTables: exploring the power of tables on the web. *PVLDB*, 1(1):538–549, 2008.
- [4] Matteo Cannavicchio, Denilson Barbosa, Paolo Merialdo. Towards Annotating Relational Data on the Web with Language Models. In WWW, 2018.
- [5] Matteo Cannavicchio, Denilson Barbosa, Paolo Merialdo. Accurate fact harvesting from natural language text in wikipedia with Lector. In WebDB, 2016.
- [6] Yusra Ibrahim, Mirek Riedewald, Gerhard Weikum. Making Sense of Entities and Quantities in Web Tables. In CIKM, 1703–1712, 2016.
- [7] Girija Limaye, Sunita Sarawagi, Soumen Chakrabarti. Annotating and Searching Web Tables Using Entities, Types and Relationships. *PVLDB* 3, 1, 1338–1347. 2010.
- [8] Varish Mulwad, Tim Finin, Anupam Joshi. Semantic Message Passing for Generating Linked Data from Tables. In ISWC, 363–378. 2013.
- [9] Emir Muñoz, Aidan Hogan, Alessandra Mileo. Using linked data to mine RDF from wikipedia’s tables. In WSDM, 533–542. 2014.
- [10] Roberto Navigli, Simone Paolo Ponzetto. BabelNet: Building a Very Large Multilingual Semantic Network. In ACL, 216–225. 2010.
- [11] Thomas Palomares, Youssef Ahres, Juhana Kangaspunta, Christopher Ré. Wikipedia Knowledge Graph with DeepDive. In ICWSM, 2016.
- [12] Dominique Ritze, Christian Bizer. Matching Web Tables To DBpedia - A Feature Utility Study. In EDBT, 210–221, 2017.
- [13] Dominique Ritze, Oliver Lehmberg, Christian Bizer. Matching HTML Tables to DBpedia. In WIMS, 10:1–10:6, 2015.
- [14] Dominique Ritze, Oliver Lehmberg, Yaser Oulabi, Christian Bizer. Profiling the Potential of Web Tables for Augmenting Cross-domain Knowledge Bases. In WWW, 251–261. 2016.
- [15] Sunita Sarawagi, Soumen Chakrabarti. Open-domain quantity queries on web tables: annotation, response, and consensus models In KDD, 711–720 2014.
- [16] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y. Halevy, Hongrae Lee, Fei Wu, Reynold Xin and Cong Yu. Finding related tables. In SIGMOD, 817–828, 2012.
- [17] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. *J. Web Sem.* 6, 3, 203–217, 2008
- [18] Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, Chung Wu. Recovering Semantics of Tables on the Web. *PVLDB* 4, 9, 528–538, 2011.
- [19] Gerhard Weikum. What Computers Should Know, Shouldn’t Know, and Shouldn’t Believe. In WebDB, 2016
- [20] Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, Surajit Chaudhuri. InfoGather: entity augmentation and attribute discovery by holistic matching with web tables. In SIGMOD, 97–108, 2012.
- [21] Shuo Zhang, Krisztian Balog. EntiTables: Smart Assistance for Entity-Focused Tables. In SIGIR, 255–264, 2017.
- [22] Ziqi Zhang. Towards Efficient and Effective Semantic Table Interpretation. In ISWC, 487–502, 2014.
- [23] Xu Chu, John Morcos, Ihab F. Ilyas, Mourad Ouazzani, Paolo Papotti, Nan Tang, Yin Ye. KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing. In SIGMOD, 1247–1261, 2015.