Estimating Missing Temporal Meta-Information using Knowledge-Based-Trust

Yaser Oulabi and Christian Bizer

Data and Web Science Group
University of Mannheim, Germany
{yaser,chris}@informatik.uni-mannheim.de

Abstract. A large number of HTML Tables on the Web contain relational data which can be used to augment knowledge bases such as DBpedia, Yago, or Wikidata. A large part of this data is time-dependent, i.e., the correctness of a fact depends on a specific temporal scope. In order to use this data for knowledge base augmentation, we need temporal meta-information. Existing methods rely on timestamps within the table itself or its context as temporal meta-information. Yet, the relationship between these timestamps and data within a table is often unclear. Additionally, timestamps are rather sparse, and there are many web tables for which no timestamps exist.

Knowledge-Based-Trust (KBT) uses the overlap with ground-truth to estimate the trustworthiness of a dataset. This paper introduces Timed-KBT, which overcomes the dependence on sparse and possibly misinterpreted timestamps by propagating temporal meta-information from a knowledge base to web table data using KBT. It also derives a trust score that estimates the correctness of the data and the assigned temporal meta-information. We evaluate Timed-KBT on the use case of fusing data from a large corpus of web tables for filling missing facts in a knowledge base. Our evaluation shows that Timed-KBT yields an increase in $F_{0.25}$-Measure of 19.01 % when compared to KBT and 9.44 % when compared to a method that relies solely on timestamps extracted from the table and its context.

1 Introduction

Besides free text, information on the Web might be represented in the form of HTML tables that contain relational data, referred to as *web tables* [2]. This relational data is potentially very useful to extend or validate multi-domain knowledge bases, such as DBpedia, YAGO, or Wikidata, which are employed for an increasing number of applications, including natural language processing, web search, and question answering [13].

Many web tables contain *time-dependent data*, in which a fact is only valid given a certain temporal scope. There is large potential and growing interest in utilizing time-dependent web data [6], e.g. for knowledge base augmentation. *Slot filling* is an augmentation task, where missing facts in the knowledge base

are filled [16]. To perform slot filling using web table data, we need data fusion strategies, which determine the value that should be added to the knowledge base given the set of alternative values found in the web tables [5, 13, 18, 19, 9]. For time-dependent data we need strategies that are *time-aware* [6], i.e., they can understand the temporal scopes of data during the fusion process.

Time-aware fusion strategies require temporal meta-information. We define *temporal meta-information* as the overall presence of *temporal scopes*, which are time annotations of certain facts or values. Existing works estimate temporal meta-information by utilizing timestamps [21, 9]. *Timestamps* include all temporal expressions that exist in a table and its context. They can be extracted from multiple locations, e.g. from page titles, text around tables, headers of columns and cells of the table. Fusion strategies that solely make use of timestamps suffer from two problems. First, the relationship between timestamps and the data in the table is often unclear. More than one timestamp can usually be extracted per table and many of the extracted timestamps likely have no relevance to the data in the table at all. Secondly, web tables suffer from timestamp sparsity [9], so that for many tables we are unable to extract any timestamps.

Our task is therefore to estimate missing temporal meta-information using other sources than the timestamps within a webpage.

This paper introduces Timed-KBT, an approach that estimates missing temporal meta-information using Knowledge-Based-Trust (KBT) [5]. KBT estimates the correctness of data using its overlap with ground-truth, in our case the knowledge base. It is based on the idea that non-overlapping data shares similar quality with neighboring overlapping data. This shared quality possibly incorporates multiple dimensions, e.g. data, extraction and matching quality. Timed-KBT is based on the assumption that the temporal dimension, i.e the temporal scope, is one of the qualities shared by neighboring data. The idea is to use the knowledge base to detect this temporal scope for overlapping values, and propagate the scope to neighboring non-overlapping values. We further introduce and evaluate an extension to Timed-KBT, whereby the scopes that can be propagated are restricted to timestamps present in the table or its context.

We evaluate Timed-KBT on the use case of data fusion using a large corpus of web tables. As a knowledge base to be augmented we use a subset of Wikidata that contains facts about countries, cities and athletes. We further extended the subset with various time-dependent datasets. We find that Timed-KBT is able to estimate missing temporal meta-information with enough quality to improve data fusion results. By using timestamps as a restriction for Timed-KBT we are furthermore able to derive a precision-orientated time-aware fusion method[1].

The next section provides a motivating example and describes the overall use case at hand. Section 3 frames this research within related work. In Section 4 we describe our fusion methodology and Timed-KBT itself, while our experimental setup, is described in Section 5. The results are discussed in Section 6. Section 7 is our conclusion.

---

[1] The methods presented in this paper are implemented as part of the publicly available T2K Framework: http://dws.informatik.uni-mannheim.de/en/research/T2K.
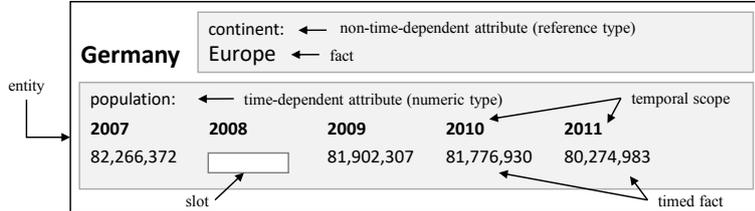
Fig. 1. An example of data for the entity 'Germany' in a temporal knowledge base.

2 Motivating Example

We aim to augment a temporal knowledge base using web table data [13, 7]. Figure 1 shows an example of data in a temporal knowledge base. Unlike *snapshot-based knowledge bases*, e.g. DBpedia, which try to reflect only the most recent facts, *temporal knowledge bases* store time-dependent data as series of timed facts. We define a *timed fact* as a fact that is annotated with a temporal scope. The knowledge base tries to reflect all current and historic facts given a certain triple. We define a *triple* as a combination of entity, attribute and fact or, given a time-dependent attribute, series of facts. An *entity* is the subject, while an *attribute* is a pre-defined property of a certain data type. In this work we deal with reference, where other entities are referenced, and numeric types.

A *slot* refers to a missing fact in the knowledge base. In this work we aim to use web table data for *targeted slot filling*, where we try to fill specific slots within a series of facts of an existing time-dependent triple. This means that the temporal scopes of the slots are previously known and provided as targets for fusion strategies. Fusion strategies are tasked to fill these *target slots* by using matched values. We define *matched values* as values extracted from web tables and matched to the knowledge base as described in the following example.

Figure 2 shows an example web table with three time-dependent columns: one leader and two population columns. The leader column corresponds to the year 2017, the first population column to 2015, a date not found on the webpage, i.e. lacking a timestamp, and the second to 1990. The figure shows that timestamps on the webpage are not explicitly associated with data, not all temporal scopes are described by timestamps and timestamps unrelated to the data exist. As such, 1990 is not assumed to be the temporal scope of the 5th column.

Rows in the table are matched to entities in the knowledge base, while columns are matched to attributes [12]. Cells are therefore matched to triples, which for time-dependent attributes corresponds to a series of timed facts. A second step is required to match cells to specific temporal scopes. More specific matching is not possible, due to the lack of explicit temporal scope annotations.

As a result, for a given target slot of the attribute population with the target year 1990, population numbers from the 4th and 5th columns are both taken as candidate values for fusion. The task of fusion methods is then to assign matched values to temporal scopes before using them to fill target slots.

Page Title: Country Data 2017

...

The following table provides information about those five countries, including capital, the national day, the current leader and current population. In comparison we provide population numbers from the year 1990

Country	Capital [15]	Current leader [98]	Population	Population 1990	National day
Germany	Berlin	Angela Merkel	81,41 M	79,43 M	3. October 1990
France	Paris	Emmanuel Macron	66,81 M	58,51 M	14th July 1790
United Kingdom	London	Theresa May	65,14 M	57,25 M	-
Japan	Tokyo	Shinzō Abe	127 M	123,5 M	11th February 660 BCE
United States	Washington, D.C.	Donald Trump	321,4 M	249,6 M	4th July 1776

....

Fig. 2. An example of a web table with time-dependent data and various timestamps.

3 Related Work

Our work is related to three research areas: (1) utilization of ground-truth to estimate the quality of web data, (2) methods for consolidating time-dependent web data, and (3) time-aware fusion for web table data.

Utilizing a ground-truth for fusion is an approach employed by many fusion methods [20, 11], including for fusing web table data [13, 9]. We base Timed-KBT on KBT, which was introduced by Dong et al. [5]. In their research the authors differentiate between factual and extraction errors, which they can because they make use of multiple extractors. In our work, this is not possible as we have only one extraction pipeline, but we do consider the temporal dimension.

For time-aware consolidation methods for web data, there is a comprehensive exploration by Dong et al. [6], where the authors describe the task's requirements and challenges. They also differentiate between the identification of timestamps and their explicit mapping as temporal scopes, but they perceive both steps to be part of the extraction process, while we perceive the mapping, i.e. Timed-KBT, as part of the fusion process. For the identification of timestamps the authors suggest HeidelTime [14], which is the method we also use in this work. The authors introduce no approaches for assigning timestamps to actual values or for generating missing temporal meta-information, which we fulfill through Timed-KBT.

In regards to time-aware fusion, Dong et al. [6] assume that temporal meta-information is provided, whereas we combine the generation of temporal meta-information and fusion in the Timed-KBT method. As fusion methods the authors mention rule-based and learning-based data fusion. In both cases, methods are specifically geared towards the reconstruction of entities with time-dependent attributes, e.g. by user-specified constraints and preference rules [1], or by inference models that specifically consider features of time-dependent data [3, 10]. In comparison, with Timed-KBT, we are able to identify temporal scopes for individual values without comprehensively creating rules or modeling entities.

Finally we will introduce two works that provide time-aware fusion for web table data. The *InfoGather+* system [21] augments an input table with attributes from a large corpus of relational HTML tables using timestamps extracted from column headers. It handles matching and fusion using a probabilistic graphical model and introduces the idea of propagating timestamp information between web tables. In our own previous research [9] we introduce a method that uses timestamps and a ground-truth to learn weighted models of the relationships between timestamp locations and attributes. We implement and test this method in this paper as `TT-Weighting`. Both time-aware fusion approaches are limited to the presence of extractable timestamps, while Timed-KBT can generate temporal scopes for any web table data, even that without timestamps, if it overlaps with data in a temporal knowledge-base. Additionally, the approaches are not able to estimate the correctness of timestamps assigned to the data, so that all timestamps in web tables are assumed to be equally relevant.

4 Methodology

We evaluate the quality of estimated temporal meta-information for the use case of data fusion. For this, we implement and compare five fusion strategies: Two baselines strategies, `Voting` and `KBT`, a time-aware strategy from previous research [9], `TT-Weighting`, and two Timed-KBT-based strategies, `TKBT` and `TKBT-Restricted`. All strategies use one common fusion framework.

4.1 Fusion Framework

The underlying fusion methodology consists of four steps:

Scoring: All fusion strategies provide scores for each matched candidate value given the temporal scope of its target slot. The fusion strategies are essentially scoring functions that influence the fusion process solely by scoring the individual matched values. Scores are provided from a range 0.0 to 1.0.

Filtering: Based on their scores, values are filtered using a learned threshold. The filtering influences a possible precision/recall trade-off. Depending on the quality of the fusion strategy's score, filtering could lead to a favorable increase in precision or an unfavorable decrease in recall. We deal with this tradeoff by optimizing thresholds for a weighted F_β-Measure. We learn thresholds in 0.05 steps from 0.0 to 1.0 and per fusion strategy and class-attribute combination.

Grouping: For a given target slot, we cluster all unfiltered *equal* values into groups. Numeric-type values are considered equal if they have a normalized similarity of at least 0.98, while reference-type values must refer to the same entity. If multiple values from the same source, i.e. the same pay-level domain, are present in one group, only the value with the highest score is kept. We sum the scores of all values in a group to calculate a group score.

Selection: For every target slot we select the group with the highest summed score and *extract* from the group a value that is chosen as the resulting fused fact of that target slot. For numeric values, where not all values in a group are exactly equal, we use the median as the extracted value.

4.2 Baseline Strategies

`Voting` is a common baseline strategy [4], where all matched values are scored 1.0. Given a target slot, the group with the most sources is therefore chosen. E.g. values from the table in Figure 2, which has only correct data, will share the same score as values of a table with mostly incorrect data. As all scores are equal, no thresholding is possible. Additionally `Voting` is not time-aware, so that the fused facts for all target slots within the same triple will be the same.

`KBT` is a fusion strategy based on Knowledge-Based-Trust [5]. It uses the correctness of data that overlaps with the knowledge base to estimate a trust score for the remaining data. It is based on the assumption that neighboring values share similar correctness. As data within a single web table column has equal extraction, normalization, matching and potentially factual quality, we compute `KBT` scores per web table columns, as shown in the equation below. As `KBT` is not time-aware, the fused facts of all target slots within one triple will be the same. With `KBT`, the values in the table in Figure 2 will e.g. have a higher score than a table with mostly incorrect overlapping data, while both population column will still be used as fusion candidates for any population target slot.

$$\text{KBT}(\text{column}) = \frac{\text{\# values in column with correct overlap}}{\text{\# values in column with overlap}} \quad (1)$$

4.3 Timed-KBT

Timed-KBT assigns explicit temporal scopes to web table data by exploiting its overlap with a temporal knowledge base. It is based on the assumption that neighboring values, e.g. within one column, share a common temporal scope. The idea is to use the knowledge base to detect this scope for overlapping values, and propagate the scope to the neighboring non-overlapping values.

To generate missing temporal scopes we first find temporal scope t that maximizes the KBT_t score of a column. The KBT_t score is computed by only using values from the knowledge base that are annotated with the given temporal scope t. We assign t to the to the web table column, while the KBT_t score itself is then used as the fusion score of the matched values. In the table in Figure 2, Timed-KBT will e.g. be able to assign different scopes to the population columns as the temporal scope t that maximizes KBT_t will likely differ per column.

$$\text{KBT}_t(\text{column}) = \frac{\text{\# values in column with correct overlap given scope t}}{\text{\# values in column with overlap given scope t}} \quad (2)$$

$$t_{column} = \underset{t \in T}{\text{argmax}}\, \text{KBT}_t(\text{column}) \quad (3)$$

$$\text{Timed-KBT}(\text{column}) = \text{KBT}_{t_{column}}(\text{column}) \quad (4)$$

We implement two Timed-KBT-based fusion strategies. In the first, `TKBT`, T is a set of temporal scopes derived from the knowledge base. In the second, `TKBT-Restricted`, we restrict T to temporal scopes that exist as timestamps

extracted from the table and its context. For the second approach, this would mean in the specific case of the table in Figure 2, that the first population column cannot be assigned a scope of 2015.

Neighboring Scope Estimation: As we assign only one temporal scope to a table column, its values are only used for the fusion of slots with that assigned scope. Assuming that temporal scopes are years and that facts of certain attributes do not completely change yearly, it would make sense to allow values that were assigned one scope, to be used to fuse facts for neighboring scopes. Given for example Figure 2 and that the first population column was assigned the scope 2015, its values can be used as candidates for slots with scope 2014, with an adapted score computed as

$$\text{estimatedScore} = \begin{cases} \text{neighboringScore} - \text{diff} \times \alpha & \text{diff} <= \text{maxDiff} \\ 0 & \text{diff} > \text{maxDiff} \end{cases} \quad (5)$$

, where *neighboringScore* equals the assigned score of the column, *diff* equals the absolute difference between the assigned year and the target year, while *maxDiff* equals the maximum difference allowed. This maximum difference is learned per class-attribute combination from 0 to 10. We therefore define α to be $0.1 = 1/10$.

5 Experimental Setup

In this section we describe the knowledge base, the web table data and how we measure fusion performance.

5.1 Knowledge Base

As the knowledge base to be augmented in our evaluation we use a subset of the temporal knowledge base Wikidata [17]. The attributes in the subset were chosen based on a profiling of the web table corpus to ensure a high overlap with web table data. For some of the chosen time-dependent attributes, Wikidata did not contain enough facts for a proper evaluation. We therefore extended the subset with various datasets that cover time-dependent data[2].

Table 1 provides an overview of the classes, entities, attributes and facts in the knowledge base. *Classes* are categories of entities and their corresponding attributes. The table also shows by class from which sources datasets were used to complement Wikidata. We acquired data from the sources either by manually written crawlers and extractors, or through data dumps.

5.2 Web Table Corpus

For our experiments we use the Web Data Commons Web Table Corpus from 2015[3], which was extracted from the July 2015 Common Crawl. The original

---

[2] The resulting knowledge base is publicly available as the Time-Dependent-Ground-Truth dataset: http://webdatacommons.org/timeddata/

[3] http://webdatacommons.org/webtables/#toc2

Table 1. Overview of classes in the knowledge base and their sources.

Class	Entities	Time-Dependent Attributes	Facts	Sources
Basketball Athlete	3,781	1	10,625	basketball-reference.com
City	11,372	2	40,666	wikidata.org
Country	197	7	44,013	worldbank.org, wikidata.org
NFL Athlete	12,756	2	96,711	footballdb.com
Political office holder	22,062	1	36,816	wikidata.org
Soccer Athlete	134,617	1	778,602	wikidata.org
Traded Company	1,646	5	73,806	stockrow.com

Table 2. Overview of attributes in the web table corpus.

Class	Attribute	Series	Avg. sources	Overlap
Basketball Athlete	Team (R)	2296	2.64	1064
City	Population (N)	5884	4.25	3120
Country	Nominal GDP (N)	191	15.14	2020
Country	Nominal GDP per capita (N)	191	68.27	7734
Country	Population (N)	193	43.93	8032
Country	Population Density (N)	193	100.37	8805
NFL Athlete	Number (N)	11295	4.35	9344
NFL Athlete	Team (R)	8962	3.56	1075
Soccer Athlete	Team (R)	14970	2.12	4688

Table 3. Proportion of sources containing timestamp in certain locations.

Class	Attribute	before	after	on page	page title	table caption	column header	in row	at least one
BA	Team	52.10%	38.75%	21.86%	26.27%	1.14%	1.45%	30.70%	72.16%
CI	Population	58.04%	55.88%	21.74%	20.48%	2.68%	16.74%	10.19%	83.00%
CO	Nom. GDP	45.38%	46.07%	24.49%	12.38%	0.80%	3.01%	26.25%	68.21%
CO	Nom. GDP-PC	53.71%	55.90%	23.71%	21.29%	3.70%	12.99%	19.65%	76.01%
CO	Population	52.47%	56.03%	27.72%	27.07%	0.96%	14.58%	21.70%	78.65%
CO	Pop. Density	56.28%	55.42%	26.62%	27.40%	1.98%	9.17%	19.00%	78.62%
NA	Number	62.20%	39.46%	30.37%	45.31%	1.23%	0.46%	28.87%	83.22%
NA	Team	69.24%	44.14%	23.78%	51.18%	1.17%	2.50%	30.79%	84.56%
SA	Team	69.38%	47.41%	38.64%	63.72%	2.91%	0.04%	8.31%	89.11%
	Average	**57.64%**	**48.79%**	**26.55%**	**32.79%**	**1.84%**	**6.77%**	**21.72%**	**79.28%**

corpus contains 1.78 billion HTML pages, whereas the web table corpus consists of 90 million relational HTML tables [7]. We use the matching component of the T2K Framework [12] to match the corpus to the knowledge base. Columns in the web tables are matched to attributes, while the rows are matched to entities.

Table 2 describes the time-dependent attributes of the corpus that we utilize. Attributes of datatype reference and numeric are denoted by (R) and (N) respectively. The column 'Series' lists the number of triples of an attribute for which values from the web tables were matched. We use the term series, because a match for a time-dependent triple is seen as a candidate for the whole series of timed facts of that triple. The following column shows how many sources exist per series on average. The column 'Overlap' measures for how many timed facts in the knowledge base, there were candidate matched values which were equal to the fact, i.e. correct matches. We have filtered from the corpus all sources

that were used to create the knowledge base (see Table 1). We additionally excluded all triples with only one matched source from our experiments, because we consider the fusion of values from one source not to be a proper fusion task.

We extracted timestamps using *HeidelTime*[4] [15]. Table 3 shows the proportion of sources that have timestamps in certain locations. Columns '*before*' and '*after*' refer to timestamps found in the context before and after the table respectively. Column '*on page*' refers to timestamps found anywhere on the page, while column '*page title*' refers to those found in the page title. The following three columns refer to timestamps extracted from table captions, column headers and cells of the same row of a value. The final column gives the proportion of sources for which a timestamp can be extracted from at least one location.

Most timestamps are found in the context of the table, which could mean that they have no explicit relation to the data in the table. Timestamps extracted from cells of the same row could similarly describe an unrelated date attribute, e.g. the 'National day' column in Figure 2. Timestamps in table captions and column headers, which are likely to be the most relevant, are sadly also the least present. Presence also differs by class: For Country and City we find many in the column header, while for NFL Athlete we find more in cells of the same row.

5.3 Evaluation

To test our fusion methods we make use of the Local-Closed-World-Assumption (LCWA), where we assume that facts present in the knowledge base are correct and can be used to determine whether fused facts are correct. The LCWA has been used and empirically examined by research with a similar task [13].

We use the F_β-Measure [8] as our performance metric. The F_1-Measure has also been used for a similar task [9, 13]. It has equal weights for both precision and recall. For the task of slot filling we must ensure the correctness of filled facts, so that we care primarily about precision. We therefore compute results for F_β-Measure at β of 1.0 and 0.25, where the latter weights precision four times as high as recall. The choice of β also affects the learned filtering thresholds described in Section 4.1. We measure performance per class-attribute combination.

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{\beta^2 \times Precision + Recall} \qquad (6)$$

As the knowledge base is used for both, learning and testing, we split the data four times, each time placing approximately 25 % of the data in the testing set, and the remainder in the learning set. To replicate the use-case of targeted slot filling, where some missing slots within a series are to be filled, we split by series of timed facts, so that some timed facts of a triple are in the testing set, while the remaining are used for learning. To ensure that the temporal scopes of removed facts are well distributed, we randomize how each series is split.

Within this paper we define temporal scopes as years. Nonetheless, it is possible that in web tables we will also find attributes which are more frequently updated and would require more fine-grained temporal scopes.

---

[4] https://github.com/HeidelTime/heideltime

Table 4. Average performance by fusion strategy.

Method	Optimized for F_1				Optimized for $F_{0.25}$			
	Prec	Rec	F_1	σ_{F_1}	Prec	Rec	$F_{0.25}$	$\sigma_{F_{0.25}}$
Voting	.141	.318	.189	.179	.141	.320	.145	.140
KBT	.314	.461	.368	.294	.319	.432	.323	.267
TT-Weighting	.330	.452	.376	.293	.442	.254	.413	.297
TKBT	.495	.467	.471	.227	.541	.340	.513	.220
TKBT-Restricted	.517	.380	.430	.203	.658	.231	.571	.227

6 Results and Findings

In this section we will present and discuss the overall results of the implemented fusion strategies and discuss the effect of the neighborhood scope estimation.

6.1 Fusion Results

Table 4 shows the average performance by fusion strategy. We can first of all see that KBT outperforms Voting by a large margin for both F_1 and $F_{0.25}$. Additionally KBT has the highest recall for $F_{0.25}$ and among the highest for F_1, which means that any strategy that outperforms KBT, does so by increasing precision.

For F_1 the difference between KBT and TT-Weighting is minimal. For $F_{0.25}$, there is an increase in the F-Measure and a larger increase in precision from KBT to TT-Weighting. This shows that the scores computed by TT-Weighting are relevant to the fusion precision, but also that they are only effective when a drop in recall is acceptable. The results could indicate that some timestamps are relevant to the data in the table and that timestamp locations have certain relationships with attributes, which is the main assumption behind TT-Weighting [9].

Both Timed-KBT-based approaches show an increase in performance when compared to other methods for both F_1 and $F_{0.25}$. TKBT for F_1 even has the highest recall. Through this we can infer that a knowledge base can successfully be used to generate temporal meta-information for web table data.

TKBT-Restricted outperforms TKBT for $F_{0.25}$. While its increase in precision comes at the cost of recall, the decline happens at a favorable rate. TKBT is unable to yield a higher precision for $F_{0.25}$, e.g. by increasing the threshold, without a performance drop, whereas the precision increase for TKBT-Restricted is large enough to compensate for the drop in recall. This shows that timestamps from the tables and their context can be relevant to the data and that TKBT-Restricted is able to use them effectively.

Strategies that use timestamps, i.e. TKBT-Restricted and TT-Weighting, generally speaking come at a large cost to recall. This could show that timestamps in web tables are too sparse for a high recall fusion strategy.

Table 5. Effect of neighborhood scope estimation on fusion performance.

Method	No estimation			With estimation			Change		
	Prec	Rec	F_β	Prec	Rec	F_β	$\Delta Prec$	ΔRec	ΔF_β
TKBT ($\beta = 1.0$)	.290	.299	.233	.495	.467	.471	+.204	+.168	+.237
TKBT-Restricted ($\beta = 1.0$)	.355	.299	.239	.517	.380	.430	+.162	+.081	+.190
TKBT ($\beta = 0.25$)	.427	.131	.367	.541	.340	.513	+.114	+.209	+.146
TKBT-Restricted ($\beta = 0.25$)	.567	.121	.444	.658	.231	.571	+.092	+.110	+.127

6.2 Impact of Neighboring Scope Estimation

From Table 5 we can see that incorporating neighborhood estimation into the Timed-KBT-based strategies had a large positive effect on fusion performance. The relative increase was more than 40 % for F_1 and 20 % for $F_{0.25}$ for both strategies. A rather unexpected result is that neighborhood estimation increases precision in addition to recall. The reason for the increase in precision is likely that matched values of neighboring temporal scopes with a high score can outweigh low-scoring, and probably incorrect, values assigned to the target scope.

7 Conclusion

In this work we introduced Timed-KBT, an approach that exploits the temporal meta-information in a knowledge base to generate missing temporal meta-information for web table data. We test Timed-KBT using a large web table corpus and an extended subset of Wikidata as a knowledge base for slot filling, a knowledge base augmentation task that makes use of fusion methods.

We find that Timed-KBT is able to assign useful explicit temporal scopes to web table data. We also find that using scores estimated by Timed-KBT for fusing time-dependent web table data yields a performance increases when compared to other fusion methods. We then utilized timestamps extracted from the web tables and their contexts as a restriction for candidate temporal scopes used by Timed-KBT. This approach yields a higher performance in regards to precision, and therefore a possibly more favorable performance for knowledge base augmentation. We conclude that timestamps in the table and its context are useful for a precision-oriented time-aware fusion strategy. Finally we show that data with assigned explicit temporal scopes is highly useful for estimating facts with neighboring temporal scopes.

Overall we demonstrate that a temporal knowledge base can be used to estimate missing temporal meta-information for web table data. We also show that with Timed-KBT, we are able to perform knowledge base augmentation from web table data for current and historic facts, instead of just for facts limited to one point in time. Our findings enable the utilization of time-dependent web data even when that data lacks temporal meta-information.

References

1. Alexe, B., Roth, M., Tan, W.C.: Preference-aware integration of temporal data. Proc. VLDB Endow. 8(4), 365–376 (Dec 2014)
2. Cafarella, M.J., Halevy, A.Y., Zhang, Y., Wang, D.Z., Wu, E.: Uncovering the relational web. In: WebDB. Citeseer (2008)
3. Dong, X.L., Berti-Equille, L., Srivastava, D.: Truth discovery and copying detection in a dynamic world. Proc. VLDB Endow. 2(1), 562–573 (Aug 2009)
4. Dong, X.L., Gabrilovich, E., Heitz, G., Horn, W., Murphy, K., Sun, S., Zhang, W.: From data fusion to knowledge fusion. Proc. VLDB (2014)
5. Dong, X.L., Gabrilovich, E., Murphy, K., Dang, V., Horn, W., Lugaresi, C., Sun, S., Zhang, W.: Knowledge-based trust: Estimating the trustworthiness of web sources. Proc. VLDB (2015)
6. Dong, X.L., Kementsietsidis, A., Tan, W.C.: A time machine for information: Looking back to look forward. SIGMOD Rec. 45(2), 23–32 (Sep 2016)
7. Lehmberg, O., Ritze, D., Meusel, R., Bizer, C.: A large public corpus of web tables containing time and context metadata. In: Proceedings of the 25th International Conference Companion on World Wide Web. pp. 75–76 (2016)
8. Manning, C.D., Schütze, H., Raghavan, P.: Introduction to information retrieval (2008)
9. Oulabi, Y., Meusel, R., Bizer, C.: Fusing time-dependent web table data. In: Proceedings of the 19th International Workshop on Web and Databases. pp. 3:1–3:7. WebDB '16, ACM, New York, NY, USA (2016)
10. Pal, A., Rastogi, V., Machanavajjhala, A., Bohannon, P.: Information integration over time in unreliable and uncertain environments. In: Proceedings of the 21st International Conference on World Wide Web. pp. 789–798 (2012)
11. Pasternack, J., Roth, D.: Knowing what to believe (when you already know something). In: Proceedings of the 23rd International Conference on Computational Linguistics. pp. 877–885. Association for Computational Linguistics (2010)
12. Ritze, D., Lehmberg, O., Bizer, C.: Matching html tables to dbpedia. In: WIMS '15. p. 10 (2015)
13. Ritze, D., Lehmberg, O., Oulabi, Y., Bizer, C.: Profiling the potential of web tables for augmenting cross-domain knowledge bases. In: Proceedings of the 25th International Conference on World Wide Web. pp. 251–261. WWW '16 (2016)
14. Strötgen, J., Gertz, M.: Heideltime: High quality rule-based extraction and normalization of temporal expressions. In: SemEval '10 (2010)
15. Strötgen, J., Gertz, M.: A baseline temporal tagger for all languages. In: EMNLP 2015 (2015)
16. Surdeanu, M., Ji, H.: Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In: TAC2014 (2014)
17. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM 57(10), 78–85 (Sep 2014)
18. Yakout, M., Ganjam, K., Chakrabarti, K., Chaudhuri, S.: Infogather: Entity augmentation and attribute discovery by holistic matching with web tables. In: ACM SIGMOD Conference. SIGMOD '12 (2012)
19. Yin, X., Han, J., Yu, P.S.: Truth discovery with multiple conflicting information providers on the web. IEEE TKDE'08 (2008)
20. Yin, X., Tan, W.: Semi-supervised truth discovery. In: Proc. WWW'11 (2011)
21. Zhang, M., Chakrabarti, K.: Infogather+: Semantic matching and annotation of numeric and time-varying attributes in web tables. In: Proc. of the 2013 ACM SIGMOD International Conference on Management of Data. pp. 145–156 (2013)