Summary Generation for Temporal Extractions

Yafang Wang[1(✉)], Zhaochun Ren[2], Martin Theobald[3], Maximilian Dylla[4], and Gerard de Melo[5]

[1] Shandong University, Jinan, China
yafang.wang@sdu.edu.cn
[2] University of Amsterdam, Amsterdam, The Netherlands
z.ren@uva.nl
[3] University of Ulm, Ulm, Germany
martin.theobald@uni-ulm.de
[4] Max Planck Institute of Informatics, Saarbrücken, Germany
mdylla@mpi-inf.mpg.de
[5] Tsinghua University, Beijing, China
gdm@demelo.org

Abstract. Recent advances in knowledge harvesting have enabled us to collect large amounts of facts about entities from Web sources. A good portion of these facts have a temporal scope that, for example, allows us to concisely capture a person's biography. However, raw sets of facts are not well suited for presentation to human end users. This paper develops a novel abstraction-based method to summarize a set of facts into natural-language sentences. Our method distills temporal knowledge from Web documents and generates a concise summary according to a particular user's interest, such as, for example, a soccer player's career. Our experiments are conducted on biography-style Wikipedia pages, and the results demonstrate the good performance of our system in comparison to existing text-summarization methods.

Keywords: Temporal information extraction · Knowledge harvesting · Summarization

1 Introduction

In recent years, we have seen a number of major advances in large-scale text mining and information extraction (IE). Amongst others, such efforts have led to the emergence of large knowledge graphs, which are collected by companies like Google, Microsoft, and Facebook, as well as open efforts such as DBpedia [2] and YAGO [22]. Given a piece of input text, numerous open-domain tools, such as NELL [4], ReVerb [7] or PRAVDA [31], are readily available for extracting subject-predicate-object triples from the text. However, while the extracted triples concisely capture the essential information conveyed by the original text also with respect to their temporal scope, they are usually not directly suitable for presentation to human end users. In this paper, we thus present a novel method to automatically generate natural-language summaries of such extractions.

S. Hartmann and H. Ma (Eds.): DEXA 2016, Part I, LNCS 9827, pp. 370–386, 2016.
DOI: 10.1007/978-3-319-44403-1_23

There are a number of challenges to be addressed. Existing summarization methods for natural-language text mostly just return existing sentences from the text, instead of summarizing the content at a more "abstract" level. A sentence may be long and include both key facts but also large amounts of less essential information. Additionally, key facts may need to be ranked and aggregated. What, for instance, should a short summary of, say, 100 words for a soccer player's career include? For famous players, with long careers, it may well be quite impossible to list all their clubs and games. An ideal summary might thus focus on the most important clubs and honors. Also, when extracting facts, we often just obtain observations about a series of individual time points that needs to be aggregated to obtain a larger picture, e.g., that a person not only played for Arsenal in 2013 and 2015 but over a longer period of time. Thus, it is important to move towards systems that attempt to go beyond selecting pre-existing sentences and are able to produce more concise summaries.

Contributions. We propose a method that, unlike previous approaches, attempts to identify the key facts in a document, much like a human would, and then generates concise summaries from them. We propose a new method that (1) summarizes facts extracted from multiple documents, (2) deals with temporal reordering and aggregation of potentially noisy pieces of evidence, and (3) produces a coherent abstractive text summary. Our experiments on Wikipedia biographies demonstrate the strength of this method.

Overview. Figure 1 provides an overview of our approach. Our approach is designed to follow the way a human would summarize information, by first digesting it and then capturing, aggregating, and rearranging the essential pieces of knowledge. To harvest information from both semi-structured and textual data sources, we rely on a number of extraction rules to mine a set of seed facts for our relations of interest from the semi-structured parts of the input documents (e.g., tables and Wikipedia infoboxes). These seed facts are then used to identify

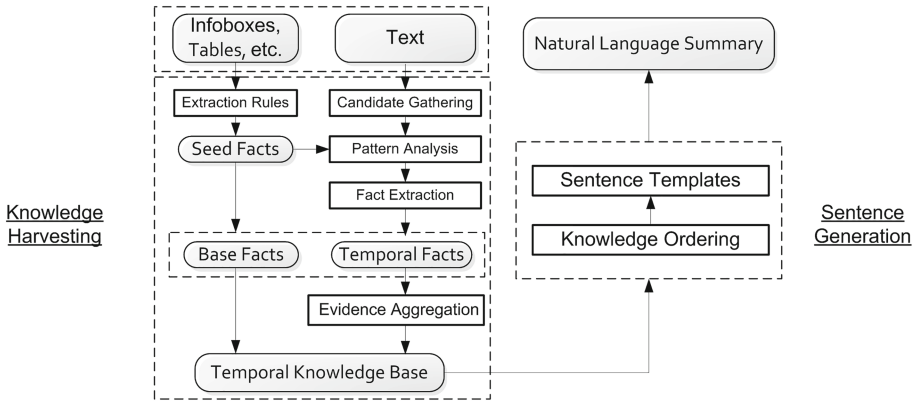

Fig. 1. Overall workflow

characteristic patterns to harvest more facts from the textual data sources. For this purpose, we rely on the general architecture of the PRAVDA system [31] to extract such facts (including temporal ones) from free-text sources. Multiple occurrences of temporal facts are reconciled via a form of evidence aggregation, which serves to condense the extracted knowledge, and in particular to extract high-confidence time intervals at which these new facts are found to be valid. Finally, for better readability and coherence of the final summary, these facts along with their time intervals are ordered chronologically and presented as natural-language sentences by mapping the temporally aligned facts onto a set of handcrafted sentence templates.

2 Knowledge Harvesting

Model. We are given a set of input sources $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ (e.g., documents) and a set of binary target relations $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ of interest. Then, the knowledge harvesting step aims at extracting instances of these relations from the input sources. Each relation R has an associated type signature $(T_R^{\mathrm{s}}, T_R^{\mathrm{o}})$, providing valid entity types for the subjects and objects of this relation. We distinguish between base facts and temporal facts. A base fact (*b-fact*, for short) is of the form $R(e_1, e_2)$, where the entity e_1 is of type T_R^{s} and e_2 has type T_R^{o}.

Temporal Facts. A temporal fact includes an additional temporal marker, which we denote as $R(e_1, e_2)@t$. This indicates that the relationship holds (i.e., is valid) at time t, which may refer to either a time point or a time interval. In our system, we define the finest granularity to be days, and all coarser granularities are converted to time intervals (e.g., January 2012 to *[1-Jan-2012, 31-Jan-2012]*). For example, *playsForClub(David_Beckham, Real_Madrid)@2005* is consistent with *playsForClub*'s type signature (*Person*, *Club*) and indicates that *David_Beckham* played for *Real_Madrid* in 2005. This may reflect just one statement in a document, while the overall time interval would be 2003–2007. We thus further distinguish between *event* and *state* relations, which we describe in Sect. 3 in more detail.

Temporal Knowledge Extraction. For semi-structured input sources, such as tables or Wikipedia infoboxes, simple extraction rules such as regular expressions suffice to extract both base and temporal facts. These are then used as seeds to find more facts in textual sources. Although not being the main focus of this paper, we briefly summarize our extraction system as follows:

1. Candidate Gathering: This step generates fact candidates and their corresponding patterns from sentences containing at least two entities (and a time marker for temporal fact candidates). The entities must satisfy the type signature of any of the relations of interest. The textual pattern of the fact candidate in such a sentence is generated by considering n-grams of the surface string between the entity pair and accounting for POS tags (for nouns, verbal phrases, prepositions, etc.).

2. Pattern Analysis: We compute the initial weight of each pattern based on the seed facts and the output of the previous step. The weight depends on the number of co-occurrences between the seed facts and the textual patterns. Patterns with co-occurrence weights above a threshold are initialized with this weight for the algorithm we apply in the next step, while the initial value for other patterns is zero.
3. Fact Extraction: In our final step, a graph is built from the fact candidates and patterns. Edges between fact candidates and patterns are added if they co-occur within a sentence. Similar patterns are also connected this way. Then, a form of label propagation [25] is utilized to determine the most likely relation for each of the fact candidates. Once a fact candidate is labeled with a particular relation R, it is called a *valid observation* and added to the set of event facts that are returned as result of the knowledge harvesting phase.

3 Evidence Aggregation Model

A main challenge in extracting and mining temporal knowledge is the proper distinction between *event* and *state* relations. For an event relation, a t-fact is valid only at a single time point. For example, *visits*(*François_Hollande, Berlin*) is valid on 24-Aug-2015. Actually, President Hollande visits Germany frequently, so there could be multiple such facts, each with different time points. State relations hold for an extended time interval during which a fact is valid at any time point within a given interval. For example, *playsForClub*(*Diego_Maradona, FC_Barcelona*) is valid in the entire interval *[1-July-1982, 30-June-1984]*. Multiple non-contiguous time spans are represented by several such state facts. The extraction of time periods for state facts is challenging, because there are typically only few occurrences of facts in input sentences with explicit time intervals. Ideally, we would encounter sentences like "Maradona had a contract with FC Barcelona from July 1982 to June 1984". However, such explicit sentences are rare in both news and web sources. Instead, we can find cues that refer to the *begin*, *end*, or some time point *during* the desired interval. For example, news articles would often mention sentences such as "Maradona did not play well in the match against Arsenal London" with a publication date of 15-May-1983 (a time point presumably contained within the corresponding state fact's interval). Thus, having extracted specific time points, we need to aggregate these into intervals for state-oriented t-facts. To address this, our method (1) aggregates individual time points into time histograms, and (2) computes a high-confidence time interval from these histograms.

We aim to aggregate individual *begin*, *end*, and *during* observations of a fact into a concise time histogram. So even if we are aiming at state-oriented t-facts, we first collect and aggregate event-style cues. Ideally, these point-wise observations would then form a compact time interval that captures the validity of the fact. However, a general problem of such an approach is the inherent ambiguity when individual events are mapped to an initially unknown amount of time

intervals. This gets even more difficult due to frequent extraction errors, overlapping occurrences of *begin* and *end* events, or other inconsistencies. However, the observations are often noisy and require non-trivial reconciliation for each base fact. First, we construct histograms for each of the begin, end, and during events. After that, the histograms are combined into a single state-oriented histogram, which is distilled into a single high-confidence interval that represents the fact's temporal validity. Finally, an algorithm computes the confidence interval of the histogram.

3.1 Aggregating Events into State Histograms

Among all observations of event facts with matching entities found in the input sentences, we first determine the time range $[t_b, t_e]$ of the largest possible validity interval of a corresponding state fact by selecting the earliest time point t_b and the latest time point t_e encountered, respectively. According to the relation an event fact has been labeled with, we classify the individual facts as *begin*, *end*, and *during* observations that mark either the possible begin or end time point, or a time point during which the corresponding state fact may be valid.

Next, all observations of *begin*, *end*, and *during* facts are aggregated into three initial histograms, each ranging over $[t_b, t_e]$. This yields one frequency value $f[t_i]$ per time point t_i. Initially, the i-th bin's frequency value $f[t_i]$ refers to the plain number of observations corresponding to this time point, for each of the three types of event facts. Subsequent time points with equal frequencies are coalesced into a single histogram bin. In each of the histograms, the bins' frequencies are then normalized to 1. For combining the three event-oriented histograms into a single histogram of the corresponding state fact, we apply the following assumptions:

1. A *during* observation at time point t_j should increase the confidence in the state fact being correct at t_j (for all time points captured by the interval of the *during* observation).
2. A *begin* observation at time point t_j should increase the confidence in the state fact for all time points ranging from t_j to t_e.
3. An *end* observation at time point t_j should decrease the confidence in the state fact for all time points t_j to t_e.

Our approach produces a multi-modal histogram if *end* facts interleave with *begin* and *during* events at different time points, which we can exploit to extract multiple validity intervals for the state fact (there are two time intervals in Fig. 2). In case none of the different event types interleave (i.e., all *begin* events occur before all *end* events, and all *during* indeed occur between all *begin* and *end* events), we obtain a uni-modal histogram from which we can extract just a single validity interval for the resulting state fact.

Algorithm 1 describes how we combine the *begin*, *end* and *during* histograms. We first merge the two *begin* and *end* histograms, before we merge the resulting *begin-end* histogram with the *during* histogram as follows (using De Morgan's law):

$$P = P_{\text{during}} \cup P_{\text{begin,end}} = \overline{\overline{P_{\text{during}}} \bigcap \overline{P_{\text{begin,end}}}} \quad (1)$$
$$= 1 - (1 - P_{\text{during}}) \cdot (1 - P_{\text{begin,end}})$$

Here, P denotes the final frequency obtained after all aggregation steps, P_{during} denotes the frequency of the *during* event, and $P_{\text{begin,end}}$ is the output (i.e., the $f[t_i]$ after the inner *for* loop in Algorithm 1) of aggregating the *begin* and *end* histograms. For all the non-empty bins in the *during* histogram, we use Eq. 1 to compute the new frequency value P. P_{during} refers to the probability of a time point indicating *during* given the observations from *during* events. The new frequency is thus the union of the probability of a time point by considering all types of events. Finally, all consecutive bins with the same frequency values are merged, and the bins are once more normalized to 1 (cf. Algorithm 1 and Fig. 2).

Algorithm 1. Aggregating events into state histograms.

Require: Event histograms with frequencies f_{begin}, f_{end}, f_{during} over the time range $[t_b, t_e]$
For all $t_i \in [t_b, t_e]$ **do**
  $f[t_i] \leftarrow 0$
  For all $t_j \in [t_i, t_e]$ **do** ▷ Aggregate *begin* and *end* histograms
    $f[t_j] \leftarrow f[t_j] + f_{\text{begin}}[t_i]$ ▷ aggregate *begin*
    $f[t_j] \leftarrow \max(0, f[t_j] - f_{\text{end}}[t_i])$ ▷ reduce *end*
  End
  $f[t_i] \leftarrow (1 - (1 - f_{\text{during}}[t_i]) \cdot (1 - f[t_i]))$
    ▷ Combine *begin,end* histogram with *during* one
End
Reorganize the bins and normalize their frequencies to 1
Return: State histogram with frequencies f

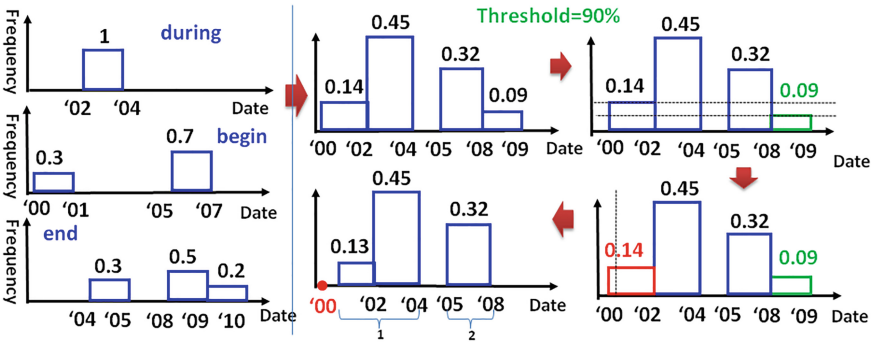

Fig. 2. Aggregating events into state histograms.

3.2 Extracting High-Confidence Intervals

The combined state histogram reflects the confidence distribution for a fact's validity over time. The value of a bin can be interpreted as the probability of the fact being valid during this bin's interval. For our temporal summarization, we next simplify the possibly very fine-grained histogram by discarding bins with a low confidence. Assuming, for example, we are interested in a final histogram that captures at least 90 % of the confidence mass of the original histogram, we discard all low-confidence bins whose cumulative frequencies sum up to at most 10 %.

Since the original histogram's bins form a discrete confidence distribution, we pursue an iterative strategy. Starting from the lowest-frequency bin, we first sort all bins by their frequency values and then check for the remaining confidence mass when cutting off these bins horizontally. Let τ be the expected threshold of the confidence interval (e.g., 90 %). Our algorithm stops as soon as we have cut off more than a threshold of $1-\tau$ (e.g., 10 %) of the overall confidence mass. We then pick the previous solution, which must still be above τ. This procedure is further refined by a final vertical trimming step of the remaining bins. To this end, we assume a uniform confidence distribution within each bin, and we adjust the frequency $f[i]$ of the trimmed bin proportionally to its cut-off width (cf. Fig. 2) until we reach τ.

4 Sentence Generation and Reordering

When summarizing multiple sources, possibly containing randomly ordered facts, it is usually not a-priori clear in which order to present these facts to the user. For short texts, we conjecture that a chronological order is appropriate in many cases.

4.1 Knowledge Ordering

Before sorting the individual facts about an entity of interest, we first roughly sort the more abstract relations associated with t-facts. Some relations can be naturally ordered. Considering a person's life, for example, the time point of a t-fact for the *isBornIn* relation must occur before the start point of a *isMarriedTo* t-fact for the same person, which in turn must occur before the time point of a *diedIn* t-fact of that person.

This order of relations can be learned statistically. Given a set of relations $\mathcal{R}$ and their temporal instances (t-facts), we build a time-ordered directed graph $G = (V, A)$, where each vertex refers to a relation and each arc represents a chronological dependency. We start by creating an initial graph $G' = (\mathcal{R}, E)$ by adding an arc (R_i, R_j) (indicating that R_i tends to precede R_j) if the support s_{ij} of R_i occurring before R_j is much greater than the inverse s_{ji}. s_{ij} is calculated by counting the instances of R_i and R_j having the same subject, i.e. $(a, b)@t_1 \in R_i$ and $(a, c)@t_2 \in R_j$, satisfying that t_1 precedes t_2. The final graph

G is then obtained from G' by adding two extra vertices representing the *start* and *end* states to G' and by removing all transitive dependencies from G'. For example, *isBornIn* may have an edge with many relations, such as *graduatedFromHighSchool*, *graduatedFromUniversity* and *diedIn*. These edges are removed according to the transitive dependencies among these relations, and only a path from *isBornIn* through *graduatedFromHighSchool*, *graduatedFromUniversity* to *diedIn* is kept. If the graph G contains a cycle, we remove the cycle by dropping the edge with the lowest support within the cycle. Figure 3 illustrates an example for transforming a set of relations into G, while Algorithm 2 shows details about how to determine the chronological order of both t-facts and b-facts according to G. For a state fact, which is valid during an entire time interval, only the start time point is taken into consideration. For example, suppose we captured that *David_Beckham* played for *Manchester_United* from 1991 to 2003, *Real_Madrid* from 2003 to 2007 and got married on 4-July-1999. The three temporal facts are ordered as {*playsForClub*(*David_Beckham*, *Manchester_United*), *getsMarriedTo*(*David_Beckham*, *Victoria_Beckham*, *playsForClub*(*David_Beckham*, *Real_Madrid*)}, according to the time points {1-January-1991, 4-July-1999, 1-January-2003}. Base facts (b-facts), which generally cannot be ordered explicitly by time, are inserted into G after the temporal facts (t-facts) in the same relation according to the topological order (Line 14).

4.2 Natural Language Generation

Similar to many other abstractive summarization methods [18], we rely on templates for natural language generation. For each relation, we manually define a number of sentence templates to construct the summary sentences. After the knowledge ordering, t-facts of the same relation are ordered next to each other due to the topological order in the relation graph. For each relation, we randomly

Algorithm 2. Knowledge Ordering

Require: Graph G; the base and temporal facts F_b and F_t.
1: $S = \emptyset$ ▷ Empty list that will contain the sorted facts.
2: $L \Longleftarrow$ Set of all vertices with no incoming edges.
3: **while** L is non-empty **do**
4: remove a vertex n from L
5: **if** n has not been visited yet **then**
6: insert all t-facts (in temporal order) of relation n into S
7: insert all b-facts of relation n into S
8: **for** each node m with an edge e from n to m **do**
9: remove edge e from the graph G
10: **if** m has no other incoming edges **then**
11: insert m into L and mark m as visited
12: sort all the t-facts of relation m by time
13: insert sorted t-facts into S
14: insert b-facts of relation m into S
15: **return** S ▷ Facts sorted by topological order of relations in G.

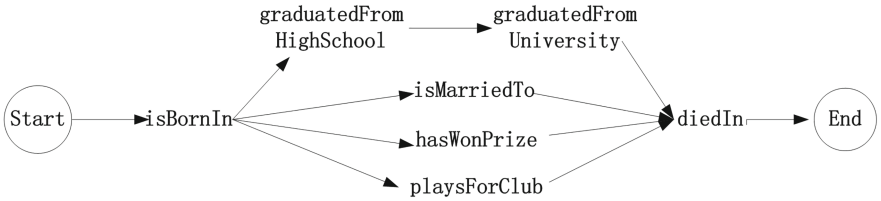

Fig. 3. Relation graph.

choose among the templates for a given subject to improve the diversity of the output. For a given subject, sentences representing the same relation are likely to contain a lot of redundancy. Thus we enable merging of arguments. For example, "*David Beckham played for Manchester United from 1993 to 2003*" and "*David Beckham played for Real Madrid from 2003 to 2007*" are merged into "*David Beckham played for Manchester United (1993–2003) and Real Madrid (2003–2007)*". For his honors, we similarly obtain the merged sentence "*David Beckham won the Premier League (1996), the FA Cup (1999), the UEFA Champions League (1999), the Intercontinental Cup (1999), and the La Liga (2007), etc.*" There are many honors, so we resort to only show the first ones.

In case there are too many facts holding the same relation, the method chooses among omitting unimportant facts, reporting the total number, or choosing only some examples for the summary sentences. Similarly, repeated occurrences of the main subject name (e.g., "David Beckham") are replaced by the corresponding pronoun ("he"), as heuristically determined by the most frequent pronoun in the source text, if available. Hence, the final summary is compressed into "*David Beckham has played for about eight clubs. He joined Manchester United in 1993. During his career in Manchester United, he won about fifteen honors including the Premier League (1996), the FA Cup (1999), etc.*". The initially redundant sentences were thus condensed into just three sentences with the key facts about David Beckham.

5 Experiments

5.1 Experimental Setup

We evaluate our method on Wikipedia articles from two domains: soccer players and movie stars. The corpora include Wikipedia articles for soccer players from the "FIFA 100 list"[1], and movie stars from the "Top 100 movies stars"[2]. For extraction, we preprocessed the corpora by replacing the most frequent pronoun by the title of the Wikipedia article, and all the entity mentions were disambiguated against the YAGO [22] knowledge base using the AIDA [10] framework for named entity disambiguation.

[1] http://en.wikipedia.org/wiki/FIFA_100/.

[2] http://articles.cnn.com/2003-05-06/entertainment/movie.poll.100_1_star-movies-godfather?_s=PM:SHOWBIZ/.

Table 1. Example sentence templates for relations.

Relation	Templates
isBornIn	ARG1 was born in ARG2
worksForClub	ARG1 served for ARG2; ARG1 worked for ARG2
actedIn	ARG1 acted in ARG2; ARG1 appeared in ARG2
hasWonHonor	ARG1 has won ARG2; ARG1 received ARG2

Relations. We list some example templates in Table 1. These are used for base facts. For temporal facts, an additional time point or time interval placeholder is added. For example, the template for temporal facts of *isBornIn* is "ARG1 was born in ARG2 on TIME". The template for temporal facts of *worksForClub* with a single time interval is "ARG1 served for ARG2 from begin_TIME to end_TIME", and for multiple time intervals we use "ARG1 served for ARG2 (begin_TIME*1*-end_TIME*1*, begin_TIME*2*-end_TIME*2*,...,begin_TIME*n*-end_TIME*n*)". For both domains, we query the system for summaries about facts associated with the birth and death dates of the respective persons, their family life (including marriage and children), honors they won, and their career (including the relations *worksForClub* for soccer players or *actedIn* for movie stars, as well as playing positions for soccer players).

Baseline Systems. We compare our system to four alternative approaches. For existing extraction-based multi-document summarization methods, we choose **NIST-Wiki** as the baseline in our experiments. NIST-Wiki extracts the first n sentences from a Wikipedia article. Since the top paragraphs in a Wikipedia article usually contain a short biography of the subject of the article, this is a very strong baseline. **LDA** here refers to an latent dirichlet allocation-based summarization method [1], which uses probabilistic topic distributions to calculate the salience for each input sentence. Additionally, as a representative model for recent abstractive summarization methods, we use Opinosis as another baseline. **Opinosis** [9] is a graph-based abstractive summarization framework. It constructs a graph from a set of input sentences set by considering redundancy and generates an optimal path from the graph. Finally, we also add a **Random** baseline to our comparison. The Random baseline just randomly selects n sentences from the data source. Since these baseline systems only support textual input data, the semi-structured sources (such as infoboxes) are translated to natural-language sentences via the sentence templates, yielding, e.g. "David Beckham has won FIFA 100".

Evaluation Procedures. We conduct two experiments. (1) We generate the summary with all the facts about a person, and (2) we generate a summary with only the most important facts and aggregated statistics. Since the first summary is longer than the second one, the corresponding baselines generate more sentences for the first experiment. We call the results from each experiment a *long summary* and a *short summary*, respectively. For the short summaries, we

limit the number of words to at most 100, for long summaries 200. Since Opinosis is limited based on the number of sentences, a short summary is limited to 10 sentences, while a long summary contains 20 sentences. We evaluate the summary for *informativeness*, *diversity*, *coherence*, and *precision* [13,16] by performing a user study. We randomly sample thirty summaries for each domain. For each of the above metrics, two human judges rate the summary on a Likert scale from one to five, where *one* means {"least informative", "least diverse", "very incoherent", "very imprecise"}, depending on the measure; while a rating of *five* means {"very informative", "very diverse", "very coherent", "very precise"}, respectively. The final score of each metric then is the average of all thirty summaries. The score of each metric in Table 4 is the average of all sixty summaries on both domains. The overall score then is the average over all metrics for each system.

5.2 Experimental Results

Table 2 provides examples of generated summaries, while the experimental results are given in Tables 3 and 4. In terms of the average over all metrics, our system outperforms all baseline systems (see Table 5). More specifically, it outperforms all the others on diversity and informativeness. On precision and coherence, NIST-Wiki is slightly better, because it just picks the first n sentences from each Wikipedia article, which are essentially human-written summaries. The precision of NIST-Wiki in the soccer domain is not 100 % correct, because in some cases it misinterpreted URL links as sentences. Since there is no 100 % perfect extraction methodology, incorrect extractions obviously affect the precision of our method. Furthermore, extraction recall can affect short summaries, since we report an aggregate number, as in "*David Beckham won about one honor*", which is incorrect. To reduce such errors, we might consider including also vague statements like "at least". Opinosis compresses the text by considering the sentence redundancy, so the newly generated sentences may change the semantics of the original sentences. This holds for semi-structured contents, which is presented as natural language, e.g. "Rui Costa has won Toulon Tournament in 1992. Rui Costa has won FIFA U-20 World Cup in 1991." Opinosis is able to compress them into one meaningful sentence "Rui Costa has won Toulon Tournament in 1992 and FIFA U-20 World Cup in 1991." While for other sentences in the Wikipedia article, most generated sentences are meaningless and often incorrect, as evident in Table 2. Opinosis generates the sentence "*Beckham's marriage in 2007--/:.*", but Beckham actually married Victoria on July 4, 1999. NIST-Wiki produces perfect coherence, as it just returns contiguous n input sentences. Other extraction-based methods, such as LDA and Random, introduce incoherence. They also introduce imprecision when the extracted sentence contains indicative pronouns, such as "after this", and temporal phrases, such as "one year later", when the prior sentences were not chosen. As for the abstractive method, some sentences generated by Opinosis are meaningless, increasing the difficulty of reading the summary. On the contrary, our system exploits simple templates that are easy to understand. Only when too many facts hold for the same relation, the generated sentence feels non-fluent. For example, "*David Beckham won*

Table 2. Example summaries.

Our System (long): *David Beckham was born in London in 1975/05/02. He played as Midfielder. He served in Manchester United F.C. (1991-2003), Real Madrid C.F. (2003-2007), Los Angeles Galaxy (2007-). He has won FA Youth Cup (1992), FA Community Shield (1993,1994,1996,1997), Premier League (1996,1997,1999,2000,2001,2003), FA Cup (1996,1999), UEFA Club Football Awards (1999), MLS Cup (2011), FIFA 100, etc.*
Our System (short): *David Beckham was born in London in 1975/05/02. He played as Midfielder. He has played for about 3 clubs and won about 45 honors. In 1991 he joined Manchester United F.C. and served for 13 years. During this time period, he was awarded FA Youth Cup (1992), FA Cup (1996), FA Cup (1999), Intercontinental Cup (1999), UEFA Club Football Awards (1999), etc.*
NIST-Wiki: *David Robert Joseph Beckham, Order of the British Empire (born 1975-05-02) is an England association footballer who plays for Los Angeles Galaxy. David Beckham has played for Manchester United F.C., Preston North End F.C., Real Madrid C.F., A.C. Milan, and the England national football team for which David Beckham holds the appearance record for a Outfield#Association football. David Beckham's career began when David Beckham signed a professional contract with Manchester United, making his first-team debut in 1992 aged 17.*
LDA: *Beckham scored the equaliser and United went on to win the match and the league. Beckham scored 9 goals that season, all in the Premier League. The income from his new contract, and his many endorsement deals, made Beckham the highest-paid player in the world at the time. In the first nine matches David Beckham started, Real lost 7. David Beckham returned to play in the final home match of the season. Beckham is Officers of the Order of the British Empire. Beckham is England expatriates in the United States.*
Opinosis: *David Beckham enjoyed tremendous following. Beckham's right midfield position. Beckham's contract became public knowledge. Beckham's maternal grandfather was Jewish. Beckham's best season as united player and united. Beckham is England under-21 international footballers. Beckham England people of Jewish descent. Beckham's marriage in 2007- -/:. Beckham crumpled hard to the ground. Beckham of the most recognisable athletes throughout the world, not concentrating on the tournament and England 's next match.*
Manual: *David Beckham, born in 2 May, 1975, is a midfielder. Beckham began his career with Manchester United in 1991. During his 13 years career there, he won several honors. He received Premier League 10 Seasons Awards for his contribution from the 1992-93 to 2001-02 seasons. He also played for Real Madrid, LA Galaxy, etc. To honor his contribution, he was named FIFA 100. On 4 July 1999, David married Victoria. They have four children: sons Brooklyn Joseph, Romeo James, and Cruz David; and daughter Harper Seven.*

the Primier League (1996,1997,1999,2000,2001,2003), FA Cup (1996), La Liga (2007), MLS Cup (2011) ...". Notice also that the informativeness and diversity are affected by recall. Our system managed to find the key information. The sentences from the semi-structured input contents facilitate LDA and Opinosis to find this key information. Specifically for LDA, those sentences get higher

Table 3. Long summaries.

System	Diversity	Informativeness	Coherence	Precision	
Ours	3.93	4.73	4.33	4.57	*Soccer*
NIST-Wiki	3.13	3.73	4.97	4.97	
LDA	3.10	4.10	3.47	4.73	
Opinosis	1.97	3.87	1.87	3.10	
Random	1.63	2.27	1.63	4.53	
Ours	3.40	4.83	4.10	4.70	*Movie Star*
NIST-Wiki	2.23	3.63	4.47	5.00	
LDA	1.87	3.63	1.97	4.77	
Opinosis	1.20	3.20	1.77	3.37	
Random	1.60	2.47	1.87	4.83	

Table 4. Short summaries.

System	Diversity	Informativeness	Coherence	Precision	
Ours	3.73	4.23	4.40	4.17	*Soccer*
NIST-Wiki	2.73	2.93	4.93	4.97	
LDA	2.40	3.63	3.23	4.73	
Opinosis	1.80	3.07	1.77	3.07	
Random	1.27	1.50	1.63	4.80	
Ours	3.37	4.53	4.47	4.03	*Movie Star*
NIST-Wiki	1.90	3.27	4.53	5.00	
LDA	1.33	3.10	2.03	4.83	
Opinosis	1.10	2.80	1.63	3.33	
Random	1.13	1.70	2.17	4.80	

topic saliency than other sentences from the free text contents in the article for each topic. Thus, LDA could extract more information from those structured sentences into the final summary. Because of this, the score of LDA and Opinosis for informativeness is better than or close to NIST-Wiki (the natural biography), according to Table 5. The diversity is not very good for all systems. No system managed to extract all information of interest. Looking at the last parts of the examples in Table 2, no system extracted summaries about Beckham's marriage and children. Considering the honors, even if our system extracted all the honors for Beckham, it is difficult to decide which ones are the most important ones to be shown in the summary, since it takes expert knowledge to judge which are the most significant. Since the LDA-based summarization strategy calculates the saliency in multiple topics, it could get different sentences focusing on different sub-topics for each article. Therefore, as shown in the results, for diversity, the LDA-based method could obtain scores close to those of NIST-Wiki.

Table 5. Overall score.

System	Diversity	Informativeness	Coherence	Precision	Overall
Ours	**3.61**	**4.58**	4.33	4.37	**4.22**
NIST-Wiki	2.50	3.39	**4.72**	**4.98**	3.90
LDA	2.17	3.61	2.68	4.77	3.31
Opinosis	1.52	3.24	1.76	3.22	2.44
Random	1.41	1.98	1.83	4.74	2.49

6 Related Work

Summarization strategies for text can broadly be categorized as either extractive or abstractive. Extractive frameworks produce a summary by selecting existing sentences from the input text and concatenating them. For example, MEAD [19] relies on a centroid clustering-based strategy to score the saliency of input sentences, while others use random walks [28] and coverage maximization with bigram concepts [20]. For the supervised methods, HMMs [6], CRFs [21] and system combinations [11] have proven effective for extractive document summarization. However, all of these approaches merely pick sentences from input documents, without attempting to identify the key facts expressed in them.

Abstractive document summarization methods seek to produce novel sentences summarizing the contents at a more abstract level. Some methods apply sentence compression techniques to remove less important parts of existing sentences [8,12]. Opinosis [9,15] generate a summary from redundant data sources by building a graph-based representation. [3] constructs new sentences by selecting and merging informative phrases. Still, all of these works aim at summarizing text, which is different from our goal of summarizing key facts extracted from both semi-structured and unstructured sources while aggregating temporal evidence.

There are also some works that aim to summarize factual information from a knowledge base. [32] introduced the notion of RDF sentences and to summarize an ontology by ranking in the ontology graph. [27] retrieves the salient type properties for a certain entity. [23] presented a diversity-aware algorithm for graphical entity summarization. [5] generates a ranked list of textual summaries for the two-length entity chains. These works only consider existing knowledge bases as input, and the summary is merely given in the form of a subgraph or list of properties, while our work automatically harvests knowledge from heterogeneous data sources, aggregates temporal and other evidence (which is much noisier and incomplete in automatic extractions than in knowledge graphs), and produces a textual summary.

There has been some previous work on temporal extraction. For instance, [29] use a combination of statistical aggregation, label propagation, and integer linear programming to extract fact. [14,26] connect time events in documents by using unimodal time histograms, whereas our aggregation approach also supports

multimodal histograms. [17,24] study the properties of relations, e.g. whether a relation is time-dependent and unique. However, all of these works aim at temporal information extraction-related tasks and do not address the issue of summarization.

An approach that handles queries over uncertain temporal facts has been presented by [30]. However it was mainly about probabilistic reasoning with rules and lineage and histograms played only an auxiliary role. CoTS [26] applies a classifier to publication dates to determine the *begin* and *end* dates of temporal facts, but does not make use of temporal expressions in text. Most importantly, both methods are limited to coping with unimodal distributions. So they cannot express that a football player was with the same club during two non-contiguous time-spans. In contrast, aggregation in this work can handle multimodal distributions.

7 Conclusion

Given the wealth of new knowledge graphs and knowledge harvesting efforts, we have proposed the novel task of summarizing temporal extractions. Our system achieves this by aggregating information in a temporally aware manner, supporting both semi-structured and textual sources. This leads to abstractive multi-document summaries beyond the capabilities of current summarization tools for text, opening up important new avenues of research on how to exploit extraction techniques in information retrieval and information management.

Acknowledgments. We thank the anonymous reviewers for their valuable comments. This project was sponsored by National Natural Science Foundation of China (No. 61503217), Shandong Provincial Natural Science Foundation of China (No. ZR2014FP002), and The Fundamental Research Funds of Shandong University (Nos. 2014TB005, 2014JC001).

References

1. Arora, R., Ravindran, B.: Latent Dirichlet allocation based multi-document summarization. In: Second Workshop on Analytics for Noisy Unstructured Text Data (AND), pp. 91–97. ACM (2008)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
3. Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., Passonneau, R.J.: Abstractive multi-document summarization via phrase selection and merging. In: ACL, pp. 1587–1597 (2015)
4. Carlson, A., Betteridge, J., Wang, R.C., Hruschka Jr., E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: WSDM (2010)
5. Chhabra, S., Bedathur, S.: Towards generating text summaries for entity chains. In: de Rijke, M., Kenter, T., de Vries, A.P., Zhai, C.X., de Jong, F., Radinsky, K., Hofmann, K. (eds.) ECIR 2014. LNCS, vol. 8416, pp. 136–147. Springer, Heidelberg (2014)

6. Conroy, J., O'leary, D.: Text summarization via hidden Markov models. In: SIGIR, pp. 406–407. ACM (2001)
7. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: EMNLP, Edinburgh, Scotland, UK, pp. 1535–1545, 27–31 July 2011
8. Filippova, K.: Multi-sentence compression: finding shortest paths in word graphs. In: ACL, pp. 322–330 (2010)
9. Ganesan, K., Zhai, C., Han, J.: Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In: ACL, pp. 340–348 (2010)
10. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: EMNLP, pp. 782–792 (2011)
11. Hong, K., Marcus, M., Nenkova, A.: System combination for multi-document summarization. In: EMNLP, pp. 107–117 (2015)
12. Knight, K., Marcu, D.: Summarization beyond sentence extraction: a probabilistic approach to sentence compression. Artif. Intell. **139**(1), 91–107 (2002)
13. Li, L., Zhou, K., Xue, G., Zha, H., Yu, Y.: Enhancing diversity, coverage and balance for summarization through structure learning. In: WWW, pp. 71–80. ACM (2009)
14. Ling, X., Weld, D.S.: Temporal information extraction. In: AAAI, pp. 1385–1390, 11–15 July 2010
15. Liu, F., Flanigan, J., Thomson, S., Sadeh, N.M., Smith, N.A.: Toward abstractive summarization using semantic representations. In: NAACL, pp. 1077–1086 (2015)
16. Mani, I.: Summarization evaluation: an overview (2001)
17. McClosky, D., Manning, C.D.: Learning constraints for consistent timeline extraction. In: EMNLP-CoNLL, pp. 873–882 (2012)
18. McDonald, D., Pustejovsky, J.: Natural language generation. In: IJCAI. Citeseer (1986)
19. Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Celebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., et al.: MEAD-a platform for multidocument multilingual text summarization. In: LREC, vol. 2004 (2004)
20. Schluter, N., Søgaard, A.: Unsupervised extractive summarization via coverage maximization with syntactic and semantic concepts. In: ACL, pp. 840–844 (2015)
21. Shen, D., Sun, J., Li, H., Yang, Q., Chen, Z.: Document summarization using conditional random fields. IJCAI **7**, 2862–2867 (2007)
22. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW, pp. 697–706. ACM, New York (2007)
23. Sydow, M., Pikula, M., Schenkel, R.: The notion of diversity in graphical entity summarisation on semantic knowledge graphs. J. Intell. Inf. Syst. **41**(2), 109–149 (2013)
24. Takaku, Y., Kaji, N., Yoshinaga, N., Toyoda, M.: Identifying constant and unique relations by using time-series text. In: EMNLP-CoNLL, pp. 883–892 (2012)
25. Talukdar, P.P., Crammer, K.: New regularized algorithms for transductive learning. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part II. LNCS, vol. 5782, pp. 442–457. Springer, Heidelberg (2009)
26. Talukdar, P.P., Wijaya, D., Mitchell, T.: Coupled temporal scoping of relational facts. In: WSDM. Association for Computing Machinery, Seattle, February 2012
27. Tylenda, T., Sozio, M., Weikum, G.: Einstein: physicist or vegetarian? Summarizing semantic type graphs for knowledge discovery. In: WWW (Companion Volume), pp. 273–276 (2011)
28. Wan, X., Yang, J.: Multi-document summarization using cluster-based link analysis. In: SIGIR, pp. 299–306. ACM (2008)

29. Wang, Y., Dylla, M., Spaniol, M., Weikum, G.: Coupling label propagation and constraints for temporal fact extraction. In: ACL, vol. 2, pp. 233–237 (2012)
30. Wang, Y., Yahya, M., Theobald, M.: Time-aware reasoning in uncertain knowledge bases. In: MUD, pp. 51–65 (2010)
31. Wang, Y., Yang, B., Qu, L., Spaniol, M., Weikum, G.: Harvesting facts from textual web sources by constrained label propagation. In: CIKM, pp. 837–846 (2011)
32. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on RDF sentence graph. In: WWW, pp. 707–716 (2007)