ISBN 978-83-60810-27-9
ISSN 1896-7094

Entity Summarisation with Limited Edge Budget on Knowledge Graphs

Marcin Sydow[1,2], Mariusz Pikuła[1], Ralf Schenkel[3], Adam Siemion[1]
[1]Polish-Japanese Institute of Information Technology, Warsaw, Poland
[2]Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
[3]Max-Planck Institut fuer Informatik, Saarbruecken, Germany
{msyd, mariusz.pikula}@poljap.edu.pl, schenkel@mpi-inf.mpg.de, adam.siemion@gmail.com

***Abstract*—We formulate a novel problem of summarising entities with limited presentation budget on entity-relationship knowledge graphs and propose an efficient algorithm for solving this problem. The algorithm has been implemented together with a visualising tool. Experimental user evaluation of the algorithm was conducted on real large semantic knowledge graphs extracted from the web. The reported results of experimental user evaluation are promising and encourage to continue the work on improving the algorithm.**

I. Introduction and Motivation

KNOWLEDGE graphs are useful for representing semantic knowledge, often automatically extracted from open domains such as the web [1] in the form of entity-relationship triples. In this data model the nodes represent entities (e.g. a director or actor, in the movie domain), and directed arcs represent binary relations between the entities (e.g. “directed”,”acted in”, etc. in the movie domain). Multiple arcs between nodes are allowed (as a person can be a director and a producer of the same movie, for example) resulting in a directed multi-graph (fig. 1). There can be weights attached to nodes or arcs in the knowledge graph that reflect some additional information, for example “witness count” – reflecting the frequency of encountering the triple in the corpus, so that an arc with high witness count could be regarded as more “important”.

A standard example of this model is the RDF[1] data format with its SPARQL[2] query language, where a query can be viewed as a sub-graph pattern that is matched with the knowledge base to produce the results.[3]

Structured query languages for knowledge graphs such as SPARQL allow for semantic search and are very expressive, however they are quite complex for unexperienced users and they also assume some prior knowledge about the domain (e.g. names of relations, etc.) which limits their applications.

Simply speaking, there are currently two extremes in search paradigms: popular and simple keyword-based search interfaces that do not support semantic search, and prototype semantic search systems that enable very precise querying but demand a lot of knowledge and experience from the user.

Consequently, there is a gap between these two extremes: it would be ideal to have a tool that enables search over semantic knowledge bases and, at the same time, is very simple and does not assume any prior experience or knowledge.

In this paper we aim at filling this gap. Namely, we formulate a novel problem of answering “precis” queries on knowledge graphs, propose an efficient algorithm for solving it, and report on prototype implementation together with preliminary experimental results obtained on real data. Similar problem of “precis” queries was previously considered in the context of relational databases [4], from which we borrowed the name “precis” (meaning a short summary about a person, etc.) and in the context of query-dependent [3] and constrained [5] summarisation of XML documents.

A. Related Work

Summarization has been intensively studied for text documents, for example by Wan et al.[9], specificly for scientific papers by Hassan et al., [2], and for multiple documents by Wan [8]. Zhang et al. [10] (as a recent example for a large set of similar papers) consider the problem of creating concise summaries for very large graphs such as social networks or citation graphs; in contrast to this, our work aims at summarizing information around a single node in a graph. Ramanath et al. [6] propose methods for summarizing tree-structured XML documents within a constrained budget.

Similar problem to the one discussed here, but with a special focus on diversification was recently discussed in [7].

II. Problem Formulation

Assume a user would like to know “something about” an entity (e.g. “Woody Allen”) but does not know anything except its name to start searching.

It would be desirable that a semantic search system accepts the query ”Woody Allen” and returns some fragment of the knowledge graph that “reasonably summarises” this entity.

Unfortunately, such kind of query is not supported by the SPARQL standard.

The most equivalent SPARQL query seems to be *“Woody Allen” ?r ?x* that requires returning the whole subgraph of the knowledge base that is in the one-hop distance from the “Woody Allen” node. However, such a solution might be not the most desired one for at least two reasons:

[1] http://www.w3.org/RDF

[2] http://www.w3.org/TR/rdf-sparql-query

[3] Formally, data in RDF consists of triples: subject-predicate-object, but this is mathematically equivalent to the multi-graph model described above

978-83-60810-27-9/09/$25.00 © 2010 IEEE

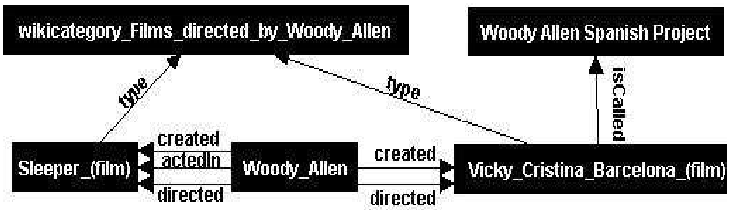
wikicategory_Films_directed_by_Woody_Allen
Woody Allen Spanish Project
type
type
isCalled
Sleeper_(film)
created
actedIn
directed
Woody_Allen
created
directed
Vicky_Cristina_Barcelona_(film)

Fig. 1: An excerpt from a semantic knowledge graph extracted from the IMDB database, concerning the movie domain.

1) the result may be too large to be comprehended by a user (e.g. in the knowledge graph extracted from the imdb database that we use, concerning movies, the “Woody Allen” node is adjacent to over 170 different arcs).
2) there may be “interesting” pieces of information concerning “Woody Allen” that are “closer” (in terms of arc weights) to the entity but are a few hops away from it in the knowledge graph (fig. 4,5)

To address these issues we introduce two elements to our model of “precis” query. First, we introduce a parameter $k \in N$ that models limited “budget” of user comprehension or display device capacity, etc. that specifies the upper bound on the number of the arcs in the presented result. Second, we introduce a novel but natural notion of distance between an arc a and node x in the knowledge graph. We assume it is the minimum sum of arc weights on a path connecting x and a, including the weight of a.

We propose the following specification of the “precis” query problem on knowledge graphs:

INPUT: B (knowledge base) – a multi-digraph with positive, real weights on arcs (considered as distance measure – currently, we use $1/witnessCount$ as the weight value); x (entity under interest) – a node of B; $k \in N$ (limit on arcs)

OUTPUT: subgraph D of B, containing at most k arcs of B, together with their end nodes, that are “closest” to x with respect to the arc-node distance

III. THE ALGORITHM

The problem is similar to some very well known ones such as the single source shortest paths or incremental search, however the specific conjunction of constraints such as multiple and weighted arcs, the notion of arc-node distance and limited presentation budget, taken together, make it a unique, novel graph problem, up to the author’s knowledge. We propose the following algorithm to solve the problem (fig. 2).

It can be obiously viewed as a modification of the Dijkstra’s single-source shortest paths algorithm adapted to the formulation of our problem. In each iteration an arc is added to RESULT thus the algorithm always stops after k iterations, at most. If we assume that there are n edges in the radius k from x in B and that comparison of *distance* value is the dominating operation, time complexity is $O(nlog(n))$ if we use a hashset implementation for RESULT and even if we use ordinary Heap for implementing PQ (the algorithm could be faster, though, if Fibonacci Heap is used instead).

```
visitTop-kClosestArcsInMultiGraph(B,x,k)

forEach a in radius k from x: a.dist := "infinity"
forEach a adjacent to x: {a.dist := a.weight; PQ.insert(a)}
while( (RESULT.size < k)
and ((currentArc = PQ.delMin()) != null) )
  forEach a in currentArc.adjacentArcs:
    if (not RESULT.contains(a)) then
     a.dist := min(a.dist, (a.weight + currentArc.dist))
      if (not PQ.contains(a)) then PQ.insert(a)
      else PQ.decreaseKey(a,a.dist)
  RESULT.add(currentArc)
return RESULT
```

Fig. 2: Algorithm for computing “precis” queries. We assume that each arc a has two real attributes: *weight* and *distance* as well as *adjacentArcs* attribute that keeps the set of arcs sharing a node with a (except a). *PQ* is a min-type priority queue for keeping the arcs being processed, with the value of weight serving as the priority and *RESULT* is a set. *PQ* and *RESULT* are initially empty. We also assume that “infinity” is a special numeric value being greater than any real number.

IV. EXPERIMENTAL RESULTS

The algorithm has been implemented, integrated with a graph-visualising tool, and applied to the two real datasets concerning the domains of movies and books, respectively (figure 3). The selected results are visualised on fig. 4 and 5

A. User Evaluation Experiment

We also conducted a user evaluation experiment aiming at assessing the quality of results of the presented algorithm and collecting feedback in order to improve the algorithm.

We selected 20 active actors from the IMDB dataset, generated the summarisations for them, for two different levels of the egde budget k: low ($k = 7$) and high ($k = 12$) (20 results for each level). Next, we asked about 10 anonymous evaluators, who did not know details about the algorithm, for assessing the summaries. Technically, the summaries computed for 20 actors and for 2 different budget levels were presented to the evaluators by a web interface. The evaluators assessed them by answering the following questions:

- How useful do you find the result as a small entity summarisation with a very limited number of facts (edges) to be presented? (“good”,”acceptable”,”poor”,”useless”).
- How many interesting/irrelevant/missing facts are in the presented summary? (three separate questions; possible answers: “almost all”,”some”, “hardly any”)

The evaluators could also give optional textual explanations of their answers. We collected about 70 assessments.

dataset	out-nodes	in-nodes	edges	relations	weights on arcs	source
IMDB-1	59013	106682	536455	73	1/witness count	www.imdb.com
LT-1	17254	45535	644055	12	1/witness count	librarything.com

Fig. 3: Datasets used for preliminary experiments. The IMDB dataset was further used for user evaluation experiment

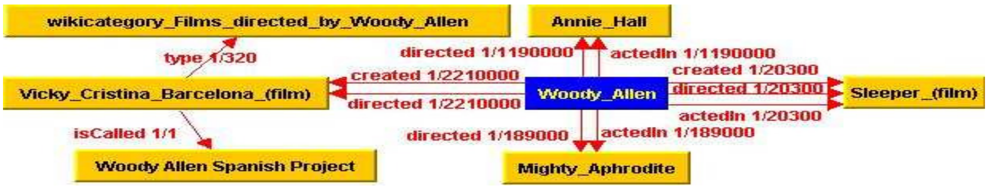

Fig. 4: An example of running the algorithm for precis query "Woody Allen" and $k = 11$ on IMDB-1 dataset. Weights represent $1/witnessCount$

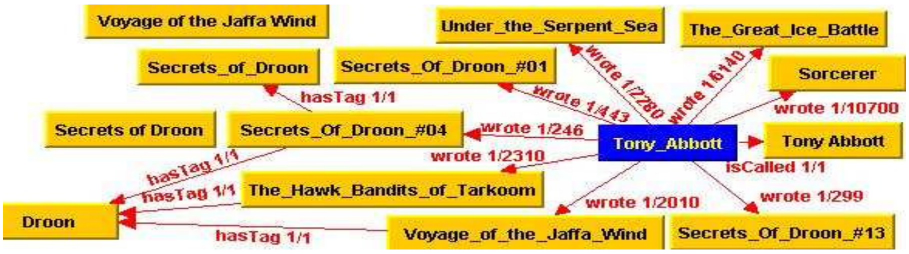

Fig. 5: An example of running the algorithm for precis query "Tony Albott" and $k = 13$ on LT-1 dataset. Weights represent $1/witnessCount$

In over 80% of the cases the summary was assessed as good or acceptable, for $k = 12$ (26% as good and only 19% as poor). In majority of cases the summary was assessed as good or acceptable (figure 6). It is noticeable that the results of the algorithm have better quality for higher value of k, while for the low value of $k = 7$, the majority of cases (67% of cases) was assessed as poor or useless (figure 6).

Concerning the assessment of the facts (edges) selected by the algorithm (figure 7), the algorithm selected "almost all" or "some" interesting facts in 83% of cases, according to evaluators. Missing facts were noticed only in 9% of cases. In 79% of cases, users did not complain about many "irrelevant" selected facts. Again, the assessments are better for the higher value of $k = 12$. See figures 8 and 9 for a detailed comparison.

V. Conclusions and Further Work

To summarise, despite the relative simplicity of the algorithm, the results are quite positive, especially for the higher value of the limit on number of presented edges.

It seems that low assessments for the low value of k is caused by the redundancy of selected facts: "actedIn" in case of actors, or "wrote" in case of writers, for example. This is also confirmed by some textual explanations of the evaluators.

Due to this observation, to improve the summarisation algorithm in future continuation of this work we plan to pay special attention to the *diversification* of the results as preliminarily studied in [7].

Though experimentation on real datasets is still in progress, the preliminary experimental results are promising since the algorithm can reach interesting pieces of information that are a few arcs away from the entity under interest. We plan to continue experimentation, also with different settings, weights and distance computation methods and modifications of the algorithm, e.g. regarding the diversity of the returned summary.

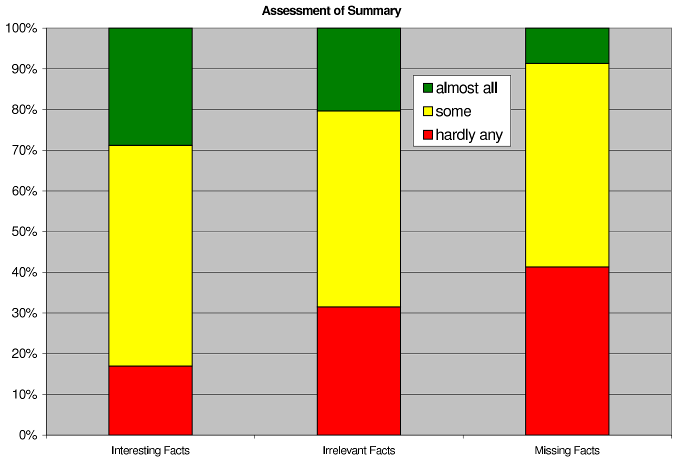

Fig. 7: Assessment of selected facts over all evaluated examples: interesting facts (left), irrelevant facts (middle), missing facts (right)

References

[1] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, 2008.

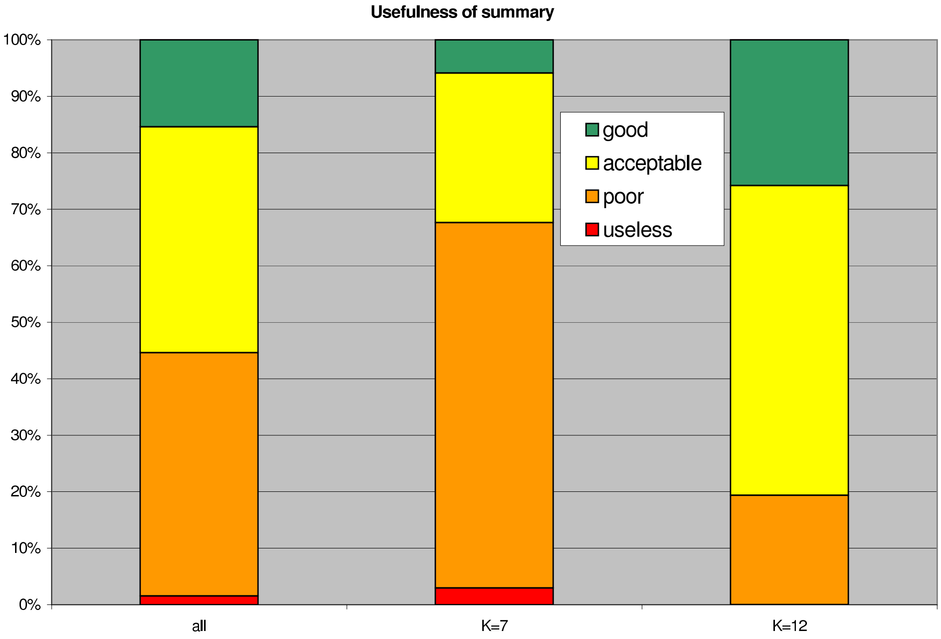

Fig. 6: Usefullness of the results

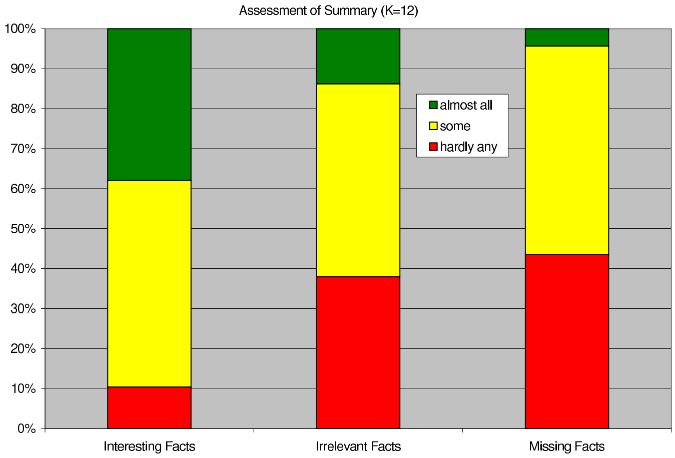

Fig. 8: Assessment of selected facts for limit on edges set to 12: interesting facts (left), irrelevant facts (middle), missing facts (right)

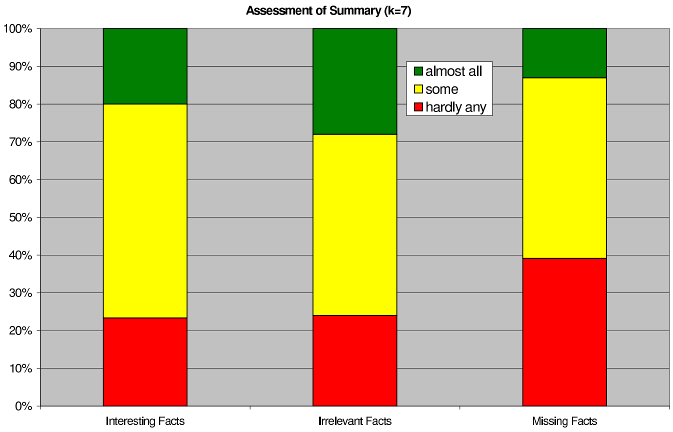

Fig. 9: Assessment of selected facts for limit on edges set to 7: interesting facts (left), irrelevant facts (middle), missing facts (right)

[2] Ahmed Hassan, Anthony Fader, Michael H. Crespin, Kevin M. Quinn, Burt L. Monroe, Michael Colaresi, and Dragomir R. Radev. Tracking the dynamic evolution of participants salience in a discussion. In *COLING*, pages 313–320, 2008.

[3] Yu Huang, Ziyang Liu, and Yi Chen. Query biased snippet generation in xml search. In Jason Tsong-Li Wang, editor, *SIGMOD Conference*, pages 315–326. ACM, 2008.

[4] Georgia Koutrika, Alkis Simitsis, and Yannis Ioannidis. Précis: The essence of a query answer. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*, page 69, Washington, DC, USA, 2006. IEEE Computer Society.

[5] M. Ramanath and K. S. Kumar. A rank-rewrite framework for summarizing xml documents. In *ICDE Workshops*, pages 540–547. IEEE Computer Society, 2008.

[6] Maya Ramanath, Kondreddi Sarath Kumar, and Georgiana Ifrim. Generating concise and readable summaries of xml documents. *CoRR*, abs/0910.2405, 2009.

[7] Marcin Sydow, Mariusz Pikuła, and Ralf Schenkel. DIVERSUM: Towards diversified summarisation of entities in knowledge graphs. In *Proceedings of Data Engineering Workshops (ICDEW) at IEEE 26th ICDE Conference*, pages 221–226. IEEE, 2010.

[8] Xiaojun Wan. Topic analysis for topic-focused multi-document summarization. In *CIKM*, pages 1609–1612, 2009.

[9] Xiaojun Wan and Jianguo Xiao. Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. *ACM Trans. Inf. Syst.*, 28(2), 2010.

[10] Ning Zhang, Yuanyuan Tian, and Jignesh M. Patel. Discovery-driven graph summarization. In *ICDE*, pages 880–891, 2010.