Entity Summarisation with Limited Edge Budget on Undirected and Directed Knowledge Graphs

Marcin Sydow[1,2], Mariusz Pikuła[1], Ralf Schenkel[3]

[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

Abstract. The paper concerns 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 in two variants: undirected and directed, together with a visualisation 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.

1 Introduction

Knowledge graphs are useful for representing semantic knowledge, often automatically extracted from open domains such as the web [7] 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 are usually pre-computed during the extraction phase. The weights can represent some notion of strength of the relation and can be used for improving the quality of data processing or searching in knowledge graphs (e.g. [6]). In this paper, the weights represent the inverse of "witness count" i.e. the frequency of encountering the considered triple in the corpus, so that an arc with high witness count (and thus, the low value of its inverse) could be regarded as more "important" or "stronger".

A standard example of this model is the RDF[4] data format with its SPARQL[5] query language, where a query can be viewed as a sub-graph pattern that is matched with the knowledge base to produce the results.[6]

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

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

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

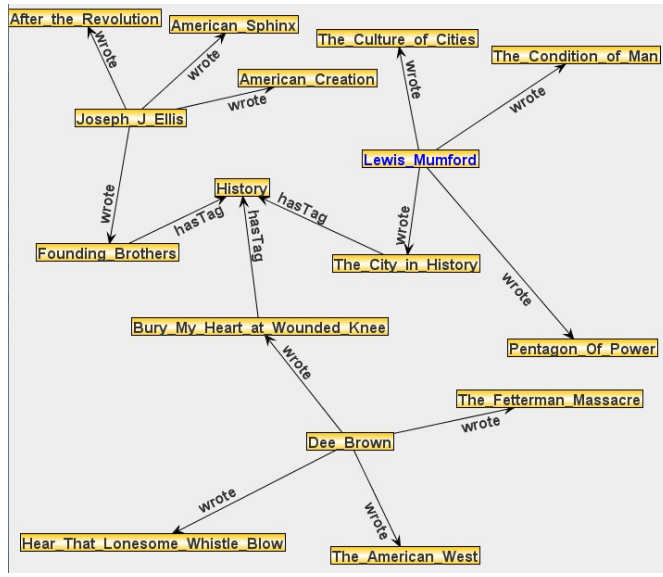

Fig. 1. An excerpt from a semantic knowledge graph extracted from the "Library Thing" database, concerning the books domain.

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.

1.1 Motivation and Contributions

To summarise, 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 summarising entities in semantic knowledge graphs, propose an efficient algorithm for solving it, demonstrate the implementation of the visualisation tool, and report experimental results concerning user evaluation of the algorithm on real data.

This article is an extended version of the conference paper [20]. The extension concerns, besides the extended "related work" section, newly implemented "undirected" variant of the summarisation algorithm together with its discussion (Section 5) and completely novel choice of figures produced by the latest version of the visualisation tool that illustrate the discussed examples.

1.2 Related Work

A related problem of "precis" queries was previously considered in the context of relational databases [11], from which we borrowed the name "precis" (meaning a short summary about a person, etc.) and in the context of query-dependent [10] and constrained [14] summarisation of XML documents. Summarization has been used in many fields for presenting complex data to human users.

Summarization of text documents has been intensively studied for many years, Mani [13] gives an overview of important results. More recent results include text summarization with latent semantic indexing [1], extraction of key phrases [3, 23], and summarization of information from multiple documents [22]. Summarisation of scientific papers was studied by Hassan et al. [9]. For all these approaches, their input is unstructured text, not semantic graphs. An important subclass of text summarization systems first extracts some semantic information from the text and then constructs a (textual) summary of the text from them [18, 5].

A second line of work considers summarization of semistructured documents such as XML. Amini et al. [2] aim at extracting the most important sentences of an XML document, using the structure of the text as additional features. Ramanath et al. [16, 17] propose methods for summarizing tree-structured XML documents within a constrained budget, focusing on data-centric XML such as IMDB or DBLP. The selection of elements and attributes to include in the summary is based on within-document and corpus-wide statistics such as corpus frequency of tags and a language model of the text in the XML document. Yu and Jagadish [25] consider the problem of summarizing huge XML schemas for presentation to a user, but do not include the content of documents in their summaries. DescribeX [4] allows to create summaries of huge collections of XML documents, focusing on both schema and content. Huang et al. [10] compute snippets of an XML document to return as result of a query. In contrast, our work is independent of a query and can summarize information of an entity from a graph, not just from an XML tree.

A third line of work considers creating concise summaries of limited size from very large graphs such as social networks or citation graphs. Here, important examples include k-snap [21] and CANAL [26]. Navlakha et al. [15] present a graph summarization that groups nodes into aggregate nodes and keeps a list of corrections, i.e., edges that occur between aggregate nodes, but not between nodes in the original graph. Wang et al. [24] use cohesive subgraphs to visualize large graphs. Similar techniques have been used in the context of summarizing knowledge graphs as a whole [12]. Harth et al. [8] make use of concise summaries of distributed knowledge bases for query routing. In contrast, our work aims at summarizing information around a single node in a graph, not at summarizing the complete graph.

The only work that attempts to solve a similar problem to the one discussed here is [19], which considers the problem of summarizing information in knowledge graphs with a special focus on diversification; that work is orthogonal to that presented here and can be combined with it.

2 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. 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

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, where the arcs can be traversed only in accordance with their orientation (i.e. from "source" to "target" of the arc).

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

3 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

```
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.

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).

4 Experimental Results

The algorithm has been implemented, integrated with a graph-visualising tool (figure 4), and applied to the two real datasets concerning the domains of movies and books, respectively (Tab. 1). Example of result is visualised on fig. 3.

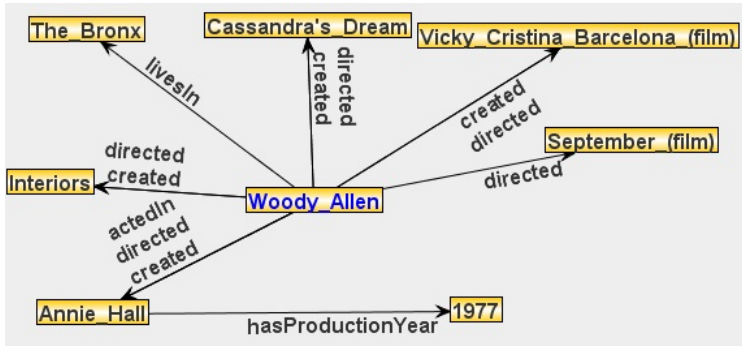

Fig. 3. An example of running the algorithm for query "Woody Allen" and $k = 12$ on IMDB-1 dataset.

Table 1. Datasets used for preliminary experiments. The IMDB dataset was further used for user evaluation experiment

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

Figure 5 is a good illustration of the ability of the algorithm to reach interesting pieces of information that are few hops away from the summarised node and that make the summary richer and more useful.

4.1 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 ($k = 12$)

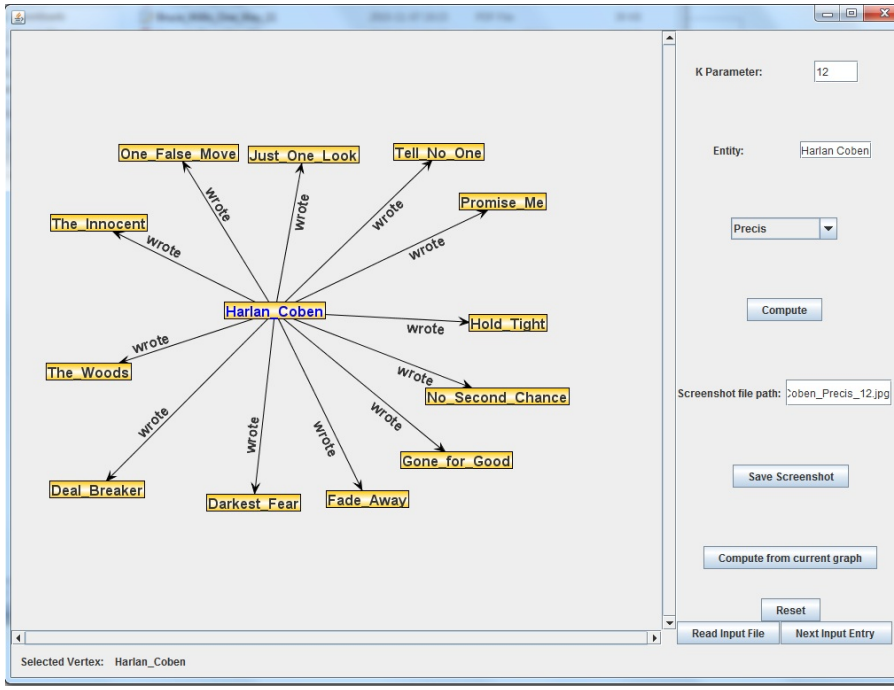

Fig. 4. Interface of the visualisation tool.

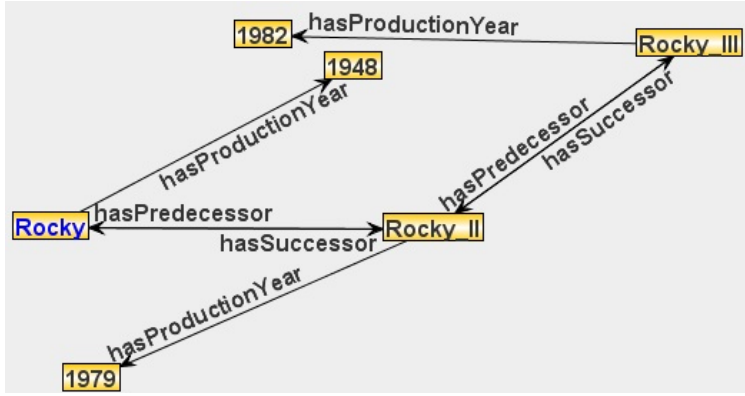

Fig. 5. An example of summary (query: "Rocky", k=7) where the algorithm can reach interesting pieces of information that are many hops away from the summarised entity which results in a more complete summary.

(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.

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.

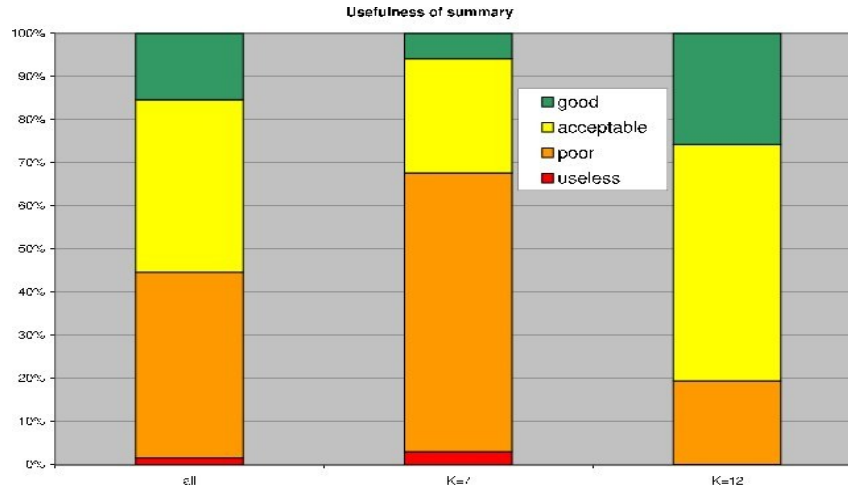

Fig. 6. Usefullness of the results

5 Undirected Variant of Entity Summarisation

Knowledge graphs considered in this paper are directed since relations (represented by edges) between the entities (represented by nodes) are not symmetric in most cases.

Due to this, and to the definition of arc-node distance (Section 2) that is used to define the summarisation algorithm, the result of entity summary can contain only those nodes that are reachable from the entity being summarised by paths that comply with the orientation of the edges.

In this section, we consider a variant of the algorithm where the edges are treated as undirected. In this variant, the summary can contain edges (and nodes) that are traversed in the opposite direction to their actual orientation. In other words, the arc-node distance is computed as the graph was undirected (however, the visualisation of the result presents the information concerning the actual direction of the edges, to not cause confusion).

The motivation for studying such a variant is two-fold.

First of all, during the experimentation with the basic variant of the algorithm we found examples for which the directed nature of the graph traversal could not find enough edges reachable from the summarised node to meet the edge number constraint (see figure 10 for example).

This kind of problem can be alleviated by allowing for traversing the edges in both directions. Such a variant of the summarisation algorithm has been implemented[7]. The result for the problematic query mentioned above, but obtained with the "undirected" variant, is presented on figure 11.

[7] Actually, the figure 1 is obtained by running the undirected variant of the algorithm

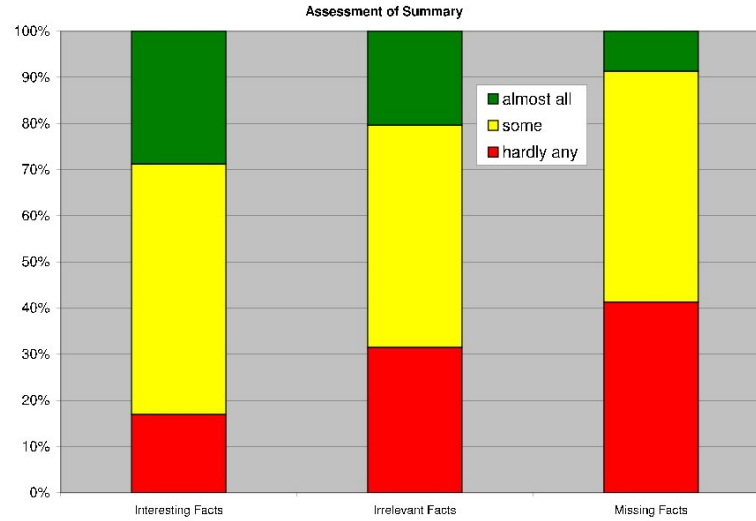

Fig. 7. Assessment of selected facts over all evaluated examples: interesting facts (left), irrelevant facts (middle), missing facts (right)

Second, it is quite natural to expect to find interesting and important pieces of information concerning the summarised entity that are close to it, but are not reachable when the arcs can be traversed only according to their direction.

Having stated this, one should be aware of some new problems that arise when the undirected variant of the algorithm is applied. As could be observed on figure 11, allowing for traversing the edges in both directions can easily bring to the summary some pieces of information that are not closely related to the summarised entity (e.g. "Danny Aiello acted in Lon" on figure 11). This phenomenon is known as *topic drift*.

We have observed that topic drift happens for the undirected variant of summarisation algorithm in many cases, even for those, for which the directed variant behaved satisfactorily. Figure 12 illustrates such a case.

A special kind of topic drift can be observed when summarisation algorithm reaches a node that represents an object for many unrelated triples that results in an "explosion" of unrelated facts in the summary. In the case of IMDB dataset this is especially true for node of type "year" (see figures 12 and 13).

It does not seem to be a trivial problem of how to balance the undirected and directed traversal in order to get high quality summaries in general. A simple idea to be studied in

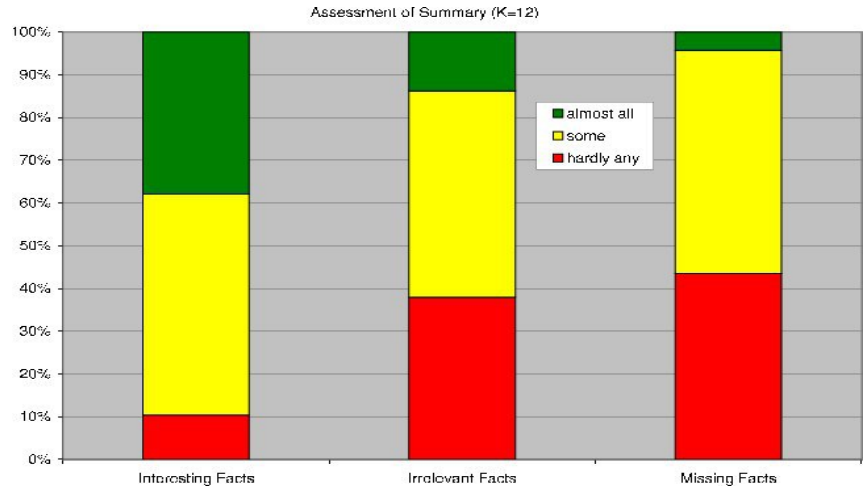

Fig. 8. Assessment of selected facts for limit on edges set to 12: interesting facts (left), irrelevant facts (middle), missing facts (right)

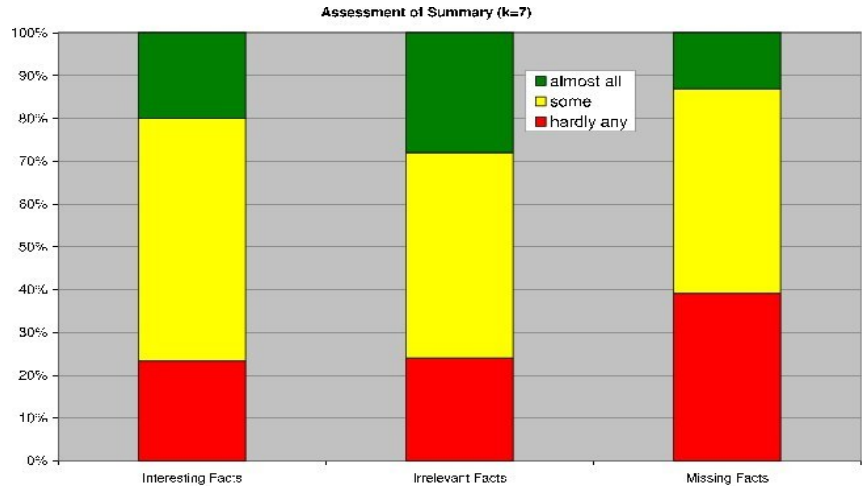

Fig. 9. Assessment of selected facts for limit on edges set to 7: interesting facts (left), irrelevant facts (middle), missing facts (right)

further work is to experiment with allowing for undirected variant only in cases when there are too little pieces of information available in the directed mode and to block undirected traversal for nodes that have high in-degree (this would work for the "year" example presented above).

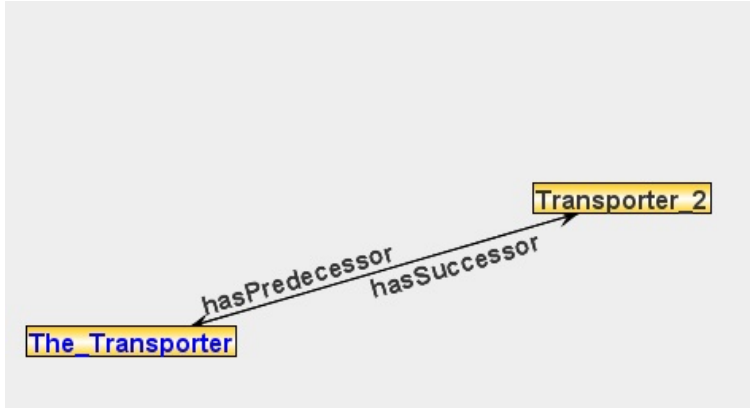

Fig. 10. Example of query "The Transporter" (movie) with k=7, where directed variant of the algorithm cannot return enough edges in the summary. This is due to the fact that in the dataset under study there are no edges outcoming any of the two identified nodes.

However, deeper study and experimental evaluation of the undirected variant of the summarisation algorithm is left as a further work.

6 Conclusions and Further Work

We have formulated a novel problem of summarising entities in knowledge graphs. As the user evaluation experiments demonstrate, the results are quite positive, despite the

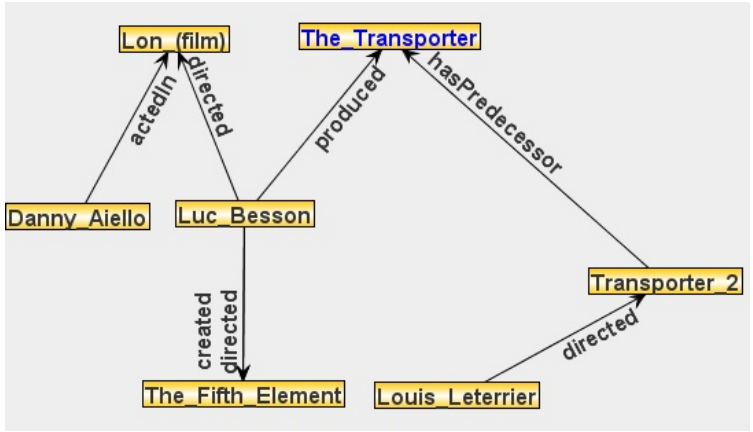

Fig. 11. Undirected variant of summary for the query "The Transporter" with k=7, does not have problems with collecting enough edges in the summary

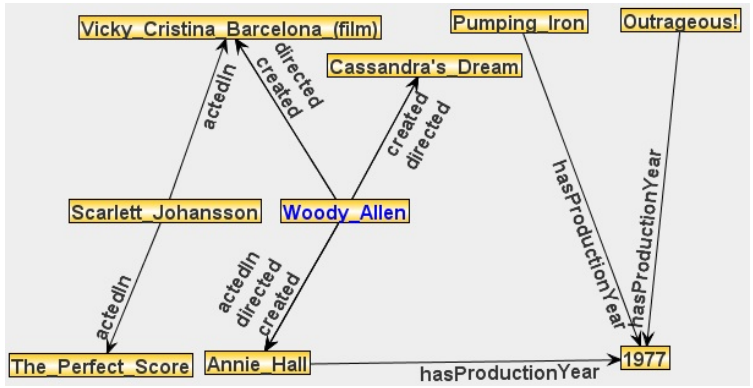

Fig. 12. Undirected variant of the summary from picture 3 ("Woody Allen", k=12). Noticeable topic drift.

relative simplicity of the algorithm, especially for the higher value of the limit on the number of presented edges.

In addition, we have implemented and discussed a novel, "undirected" variant of the algorithm that could produce more valuable summaries for some special cases that are identified and shortly discussed. Further analysis and user evaluation of the novel variant is left for future work.

It seems that reported low assessments of the basic variant of the agorithm observed 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

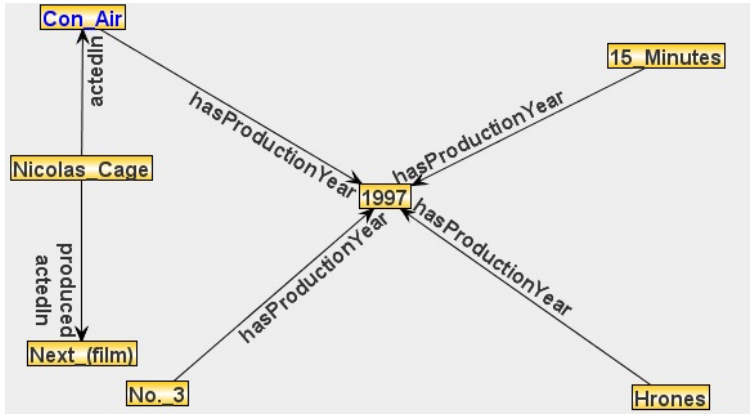

Fig. 13. Example of the summary when undirectedness introduces a novel problem: "explosion" of facts around node "year" (query: "Con Air", undirected)

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 [19].

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.

Acknowledgments

This work was supported by Polish Ministry of Science and Higher Education grant number N N516 481940.

References

1. Ai, D., Zheng, Y., Zhang, D.: Automatic text summarization based on latent semantic indexing. Artificial Life and Robotics 15, 25–29 (2010), http://dx.doi.org/10.1007/s10015-010-0759-x, 10.1007/s10015-010-0759-x
2. Amini, M.R., Tombros, A., Usunier, N., Lalmas, M.: Learning-based summarisation of xml documents. Inf. Retr. 10(3), 233–255 (2007)
3. Bedathur, S.J., Berberich, K., Dittrich, J., Mamoulis, N., Weikum, G.: Interesting-phrase mining for ad-hoc text analytics. PVLDB 3(1), 1348–1357 (2010)
4. Consens, M.P., Miller, R.J., Rizzolo, F., Vaisman, A.A.: Exploring xml web collections with describex. TWEB 4(3) (2010)
5. Demartini, G., Missen, M.M.S., Blanco, R., Zaragoza, H.: Entity summarization of news articles. In: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 795–796. SIGIR '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1835449.1835620
6. Elbassuoni, S., Ramanath, M., Schenkel, R., Sydow, M., Weikum, G.: Language-model-based ranking for queries on rdf-graphs. In: CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management. pp. 977–986. ACM, New York, NY, USA (2009)
7. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Commun. ACM 51(12), 68–74 (2008)
8. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: Data summaries for on-demand queries over linked data. In: WWW. pp. 411–420 (2010)
9. Hassan, A., Fader, A., Crespin, M.H., Quinn, K.M., Monroe, B.L., Colaresi, M., Radev, D.R.: Tracking the dynamic evolution of participants salience in a discussion. In: COLING. pp. 313–320 (2008)
10. Huang, Y., Liu, Z., Chen, Y.: Query biased snippet generation in xml search. In: SIGMOD Conference. pp. 315–326 (2008)
11. Koutrika, G., Simitsis, A., Ioannidis, Y.: Précis: The essence of a query answer. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering. p. 69. IEEE Computer Society, Washington, DC, USA (2006)

12. Li, N., Motta, E.: Evaluations of user-driven ontology summarization. In: The 17th International Conference on Knowledge Engineering and Knowledge Management by the Masses (2010)
13. Mani, I.: Automatic Summarization. MIT Press (2001)
14. M.Ramanath, K.S.Kumar: A rank-rewrite framework for summarizing xml documents. In: ICDE Workshops. pp. 540–547 (2008)
15. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph summarization with bounded error. In: SIGMOD Conference. pp. 419–432 (2008)
16. Ramanath, M., Kumar, K.S.: A rank-rewrite framework for summarizing xml documents. In: ICDE Workshops. pp. 540–547 (2008)
17. Ramanath, M., Kumar, K.S., Ifrim, G.: Generating concise and readable summaries of xml documents. CoRR abs/0910.2405 (2009)
18. Rusu, D., Fortuna, B., Mladenić, D., Grobelnik, M., Sipoš, R.: Visual analysis of documents with semantic graphs. In: Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration. pp. 66–73. VAKD '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1562849.1562857
19. Sydow, M., Pikuła, M., Schenkel, R.: DIVERSUM: Towards diversified summarisation of entities in knowledge graphs. In: Proceedings of Data Engineering Workshops (ICDEW) at IEEE 26th ICDE Conference. pp. 221–226. IEEE (2010)
20. Sydow, M., Pikuła, M., Schenkel, R., Siemion, A.: Entity summarisation with limited edge budget on knowledge graphs. In: Proceedings of the International Multiconference on Computer Science and Information Technology. pp. 513–516. IEEE (2010)
21. Tian, Y., Hankins, R.A., Patel, J.M.: Efficient aggregation for graph summarization. In: SIGMOD Conference. pp. 567–580 (2008)
22. Wan, X.: Topic analysis for topic-focused multi-document summarization. In: CIKM. pp. 1609–1612 (2009)
23. Wan, X., Xiao, J.: Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. ACM Trans. Inf. Syst. 28(2) (2010)
24. Wang, N., Parthasarathy, S., Tan, K.L., Tung, A.K.H.: Csv: visualizing and mining cohesive subgraphs. In: SIGMOD Conference. pp. 445–458 (2008)
25. Yu, C., Jagadish, H.V.: Schema summarization. In: VLDB. pp. 319–330 (2006)
26. Zhang, N., Tian, Y., Patel, J.M.: Discovery-driven graph summarization. In: ICDE. pp. 880–891 (2010)