Martin, Leon; Boockmann, Jan H.; Henrich, Andreas

Fast Pathfinding in Knowledge Graphs Using Word Embeddings

Date of secondary publication: 17.02.2025

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-1063964

Primary publication

Martin, Leon; Boockmann, Jan H.; Henrich, Andreas (2020): Fast Pathfinding in Knowledge Graphs Using Word Embeddings, in: Ute Schmid, Franziska Klügl, und Diedrich Wolter (Ed.), KI 2020: Advances in Artificial Intelligence : 43rd German Conference on AI, Bamberg, Germany, September 21–25, 2020, Proceedings, Cham: Springer, pp. 305–312, doi: 10.1007/978-3-030-58285-2_27.

Fast Pathfinding in Knowledge Graphs Using Word Embeddings

Leon Martin(✉), Jan H. Boockmann, and Andreas Henrich

University of Bamberg, An der Weberei 5, 96047 Bamberg, Germany
{leon.martin,jan.boockmann,andreas.henrich}@uni-bamberg.de

Abstract. Knowledge graphs, which model relationships between entities, provide a rich and structured source of information. Currently, search engines aim to enrich their search results by structured summaries, e.g., obtained from knowledge graphs, that provide further information on the entity of interest. While single entity summaries are available already, summaries on the relations between multiple entities have not been studied in detail so far. Such queries can be understood as a pathfinding problem. However, the large size of public knowledge graphs, such as Wikidata, as well as the large indegree of its major entities, and the problem of concept drift impose major challenges for standard search algorithms in this context.

In this paper, we propose a bidirectional pathfinding approach for directed knowledge graphs that uses the semantic distance between entity labels, which is approximated using word vectors, as a search heuristics in a parameterized A*-like evaluation function in order to find *meaningful* paths between two entities *fast*. We evaluate our approach using different parameters against a set of selected within- and cross-domain queries. The results indicate that our approach generally needs to explore fewer entities compared to its uninformed counterpart and qualitatively yields more meaningful paths.

Keywords: Knowledge graphs · Word embeddings · Pathfinding

1 Introduction

Modern web search engines aim to extend their core functionality by providing key information relevant to a queried entity in a structured way alongside with the ranked web pages. To do so, search engine providers leverage additional data sources that characterize the information of entities and the relationships among them. Google introduced the concept of a Knowledge Graph (KG) in their blog [11] as a tool for enhancing their search engine via three main features: the disambiguation of entities, the generation of summaries for entities, and the provision of links to related entities. The latter two are displayed as a knowledge panel [12] next to the vertical list of ranked web pages. Indeed, KGs provide the necessary information to construct knowledge panels, however, these panels are only constructed for queries that focus on a *single entity*. While such panels can meet the information need related to directly adjacent entities, e.g., 'inhabitants of Elva', a city in Estonia, they do not cover relationships between non-adjacent entities, e.g., 'Elva' and 'Europe'. However, the latter, so far unaddressed use case of *multiple entities* is a typical information need expressed by users.

Answering such dual-entity queries can be understood as pathfinding in KGs, e.g., the relationship between Elva and Europe is represented as the path(s) that exist between the KG entities 'Elva' and 'Europe'. Although pathfinding in general graphs is well studied [3], finding paths in KGs remains a non-trivial problem, given that paths should be found *fast* and yield a *meaningful* explanation of the entities' relationship, i.e., shorter paths are not necessarily more meaningful. The semantic focus of the query is left – a problem known as concept drift [4] – when a search algorithm decides to explore non-meaningful paths. In addition, general KGs, such as DBpedia [1] or Wikidata [13], comprise many entities and relationships yielding a large search space, i.e., the work in [9] observed that popular KGs comprise up to 18 million entities and that the approximate in-/outdegree of an entity ranges from 10/40 to 10/100.

The key challenge is the design of an appropriate search heuristics and the choice of an accompanying search algorithm suited for this particular use case where *meaningful* paths must be found *fast* in a large graph. In this paper, we thus investigate the following research question: "Does a bidirectional search algorithm using the semantic distance as a search heuristics lead to meaningful paths in KGs fast?"

2 Foundations

The central problem tackled in this paper is finding a path between the two entities v_{source} and v_{target} in a KG $= (V, E)$, as usual. As there does not exist a formal definition of a KG [5], we utilize the characterization of [8], according to which a KG encodes entities of the real world (*nodes*) and their relations (*edges*) taken from various domains in a graph representation. A schema, which links entities to classes and properties, may provide further type information.

A best-first search guided by an appropriate search heuristics is key to perform pathfinding in large KGs in a reasonable amount of time. We propose to use the semantic distance between the labels of entities as a search heuristics. Word embedding toolkits like word2vec [7] and fastText (FT) [2] produce vector representations of words in a metric space by training a model using a large textual corpus. In the remainder of this paper, we approximate the semantic distance of two KG entities v_1 and v_2 using the cosine distance $d(v_1, v_2) = 1 - cos(\theta)$, where θ is the angle between the word vectors of the corresponding entity labels. The cosine distance is derived from the cosine similarity $cos(\theta) = \frac{v_1 \cdot v_2}{||v_1|| \cdot ||v_2||}$, a standard measure used in information retrieval and natural language processing [6].

Algorithm 1. Our bidirectional pathfinding algorithm.

Require: $v_{source}, v_{target} \in \mathit{Knowledge\ Graph}$
  procedure FINDPATH(v_{source}, v_{target})
    $\mathit{priorityqueue} \leftarrow \langle v_{source}, v_{target} \rangle$
    $\mathit{reachable}_{source} \leftarrow \{v_{source}\}$
    $\mathit{reachable}_{target} \leftarrow \{v_{target}\}$
    while $\mathit{priorityqueue} \neq \varnothing$ **do**
      $\mathit{entity} \leftarrow \mathit{dequeue}(\mathit{priorityqueue})$
      if $\mathit{entity} \in (\mathit{reachable}_{source} \cap \mathit{reachable}_{target})$ **then**
        return $\mathit{reconstructPath}(v_{source}, \mathit{entity}, v_{target})$
      end if
      $\mathit{nextEntities} \leftarrow \mathit{getAdjacentEntities}(\mathit{entity})$
      $\mathit{priorityqueue} \leftarrow \mathit{enqueueAll}(\mathit{priorityqueue}, \mathit{nextEntities})$
      if $\mathit{entity} \in \mathit{reachable}_{source}$ **then**
        $\mathit{reachable}_{source} \leftarrow \mathit{reachable}_{source} \cup \mathit{nextEntities}$
      else if $\mathit{entity} \in \mathit{reachable}_{target}$ **then**
        $\mathit{reachable}_{target} \leftarrow \mathit{reachable}_{target} \cup \mathit{nextEntities}$
      end if
    end while
  end procedure

3 Concept

Our bidirectional A*-like search algorithm (see Algorithm 1) is capable of finding direct paths, i.e., from v_{source} to v_{target} and vice versa, and paths where one entity is reachable from both v_{source} and v_{target}. A shared priority queue ensures that the least costly path is explored next.

Word embeddings, used as our search heuristics, allow to estimate the semantic distance using the cosine distance of vector representations of words, i.e., entity labels. Named entities like people and geographic points of interest account for a significant part of all encoded entities and pose a problem for traditional word embedding models that can only produce word vectors for words present in their training data. The ability to reliably compute the semantic distance between all entities is essential for our use case, because missing semantic distances lead to a biased path selection. Modern word embedding models provide additional functionalities, such as producing word vectors for unknown words, e.g., the herein employed FT model achieves this by computing word vectors based on vectors of substrings. In particular, we utilize the pre-trained English word vectors `cc.en.300.bin` as provided by FT.

Our approach is based on the assumptions that the semantic distance varies between adjacent entities and that entities in proximity show a lower semantic distance on average. Figure 1 shows the distribution of the semantic distance for a selection of sample entities. Note that the semantic distance is broadly distributed indicating that some neighboring entities are indeed more worthwhile to pursue than others. For example, the adjacent entities of 'river' include 'watercourse' with a semantic distance of 0.45, 'Template:Infobox river' with a

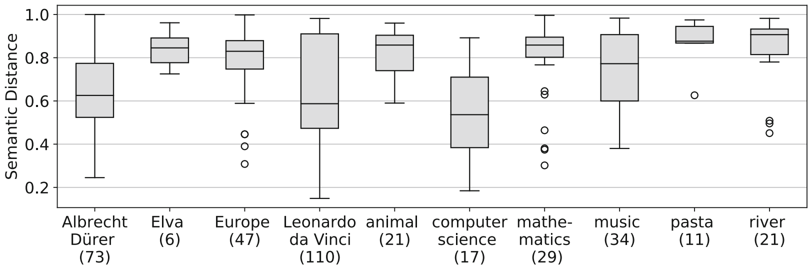

Fig. 1. Distribution of the semantic distance among the Wikidata entities reachable via a single non-reflexive outgoing edge; outdegree of an entity provided in parenthesis.

semantic distance of 0.91, and 'Explanatory Dictionary of the Living Great Russian Language' with a semantic distance of 0.98. Note that FT yields a higher semantic distance for less related entities, supporting our assumption for adjacent entities.

In the following, we consider paths meaningful that minimize the average semantic distance to the target, i.e., preventing concept drift. Hence, inspired by a general A* evaluation function, the function $f(p) = g(p) + h(p)$ denotes the costs of a path p with length n as follows:

$$
\begin{aligned}
g(p) &:= \alpha \cdot \overline{d}(p_{[1..n-1]}, v_{target}) + \beta \cdot n \\
h(p) &:= \gamma \cdot d(v_n, v_{target}) \\
&\text{where } p = \langle v_1, v_2, \ldots, v_n \rangle \\
&\text{and } p_{[i..j]} \text{ is the sub-path } \langle v_i, \ldots, v_j \rangle
\end{aligned}
$$

Formula $\overline{d}(p_{[1..n-1]}, v_{target})$ calculates the average of the semantic distances between each entity on the path – excluding the last one – and the target entity. Formula $d(v_n, v_{target})$ estimates the semantic distance between the last entity and the target entity. Note that this approach is a heuristic adaptation of the A* idea. The algorithm does not minimize the sum of the distances but focuses on the distance of all nodes on the path to the target node. The evaluation function is parameterized over variables α, β, and γ to enable component-based weight adjustments. Due to missing benchmarks for pathfinding in KGs, no parameter fitting could be performed. In the following, we explore the four sample evaluation function configurations:

1. *Uninformed* ($\alpha = \gamma = 0$, $\beta = 1$) neglects the semantic distances and only considers the prior path length, thus representing a baseline.
2. *Semantics-Only* ($\alpha = \gamma = 1$, $\beta = 0$) ignores the path length and only considers the semantic distances.
3. *Greedy* ($\alpha = \beta = 0$, $\gamma = 1$) does not consider prior path length and average semantic path costs, and estimates the remaining path costs using the semantic distance of the last entity to v_{target}.

4. *Balanced* ($\alpha = \gamma = 1$, $\beta = 0.5$) takes semantic distances and path length into account; $\beta < 1$ reduces the impact of path length on the overall costs.

4 Evaluation

Since there does not exist a standard benchmark for evaluating the performance of pathfinding in the Wikidata KG, we conduct a preliminary evaluation of the configurations above using a small handmade selection of diverse queries. This benchmark contains within- and cross-domain queries to observe how our approach performs in different search scenarios. Since the employed Wikidata SPARQL interface yields a timeout for many entities in our test set when requesting their incoming edges, this evaluation considers only outgoing edges.

Table 1 summarizes our results. First note that every query can be answered by at least one configuration that considers the semantic distance. With only a single unanswered query, the configurations *Uninformed* and *Balanced* yield the best coverage of the query set. However, the *Balanced* configuration needs to explore less entities to find a path and is also faster on average when excluding the outlier query from 'France' to 'air pollution', for which 270 entities had to be explored to find a path. Note that we forcefully stop a search that exceeds the limit of 500 explored entities. Despite the additional time necessary to properly enqueue elements into the priority queue, the runtime performance of any informed configuration is better than the uninformed configuration on average when ignoring forcefully stopped searches. In general, note that every configuration requires more time and has to visit more nodes in order to find a path for a cross-domain query compared to within-domain queries on average. Interestingly, despite having the most unanswered queries, only the configuration *Semantics-Only* found a path for the cross-domain query 'Bamberg, computer science'. Observe that this path spans 19 entities in total and could performance-wise not have been found by an uninformed search approach. Hence, the semantic distance appears to be a viable search heuristics to find paths in a large KG *fast*.

We conduct a qualitative analysis to assess the *meaningfulness* of the paths found by our configurations. The benchmark results show that our informed configurations can find useful paths, e.g., 'Elva' → 'Estonia' → 'Europe' and 'pasta' ← 'flour' ← 'pizza'. However, especially configurations that consider the path length find paths where the shared entity is an encyclopedia; the *meaningfulness* of such entities is debatable. Nevertheless, *meaningfulness* of paths strongly depends on an underlying definition, which is a subjective quality and may vary between different use cases. Hence, a ground truth, e.g., obtained from a user study, is necessary to properly assess the quality of found paths. From our point of view, the meaningfulness of the paths found using an informed configuration is mixed but promising considering that no parameter fitting has been done yet.

Note that the number of samples is way too small to draw a meaningful conclusion but suffices to motivate further investigation.

Table 1. Benchmark results for within- and cross-domain queries from v_{source} to v_{target} (entity label and Wikidata entity identifier) using the four evaluation function configurations. We provide information on the length of the found path (l) measured in the number of entities including source and target, and use symbols '♦', '♣', and '♠' in superscript to denote difference/equivalence among paths in the same row; we report on the number of explored entities (c) and the total runtime in seconds (t).

	Queries	Uninformed			Semantics-Only			Greedy			Balanced		
	v_{source}, v_{target}	l	c	t	l	c	t	l	c	t	l	c	t
Within-Domain	Elva (Q213071), Europe (Q46)	$3^{\blacklozenge}$	131	123	$3^{\blacklozenge}$	3	8	$3^{\blacklozenge}$	3	6	$3^{\blacklozenge}$	3	8
	Albrecht Dürer (Q5580), Leonardo da Vinci (Q762)	$3^{\blacklozenge}$	74	71	$c \geq 500$			$c \geq 500$			$3^{\blacklozenge}$	19	30
	computer science (Q21198), mathematics (Q395)	$3^{\blacklozenge}$	21	19	$4^{\clubsuit}$	4	7	$c \geq 500$			$3^{\spadesuit}$	19	19
	pasta (Q178), pizza (Q177)	$3^{\blacklozenge}$	31	34	$3^{\clubsuit}$	7	9	$3^{\clubsuit}$	9	10	$3^{\clubsuit}$	8	10
	river (Q4022), ocean (Q9430)	$3^{\blacklozenge}$	27	24	$5^{\clubsuit}$	9	13	$5^{\clubsuit}$	10	12	$3^{\blacklozenge}$	21	31
	Ludwig v. Beethoven (Q255), Wolfgang A. Mozart (Q254)	$2^{\blacklozenge}$	34	33	$2^{\blacklozenge}$	2	7	$2^{\blacklozenge}$	2	3	$2^{\blacklozenge}$	2	6
Cross-Domain	Leonardo da Vinci (Q762), Italy (Q38)	$3^{\blacklozenge}$	117	110	$3^{\clubsuit}$	3	15	$3^{\clubsuit}$	3	6	$3^{\clubsuit}$	3	13
	Bamberg (Q3936), computer science (Q21198)	$c \geq 500$			$19^{\blacklozenge}$	29	43	$c \geq 500$			$c \geq 500$		
	air pollution (Q131123), France (Q142)	$4^{\blacklozenge}$	260	838	$c \geq 500$			$4^{\clubsuit}$	4	5	$4^{\clubsuit}$	270	2312
	Loki (Q133147), Bible (Q1845)	$3^{\blacklozenge}$	108	70	$c \geq 500$			$c \geq 500$			$3^{\clubsuit}$	8	10
	animal (Q729), building (Q41176)	$4^{\blacklozenge}$	212	161	$c \geq 500$			$16^{\clubsuit}$	54	134	$3^{\spadesuit}$	6	5
	wood (Q287), music (Q638)	$3^{\blacklozenge}$	36	26	$c \geq 500$			$11^{\clubsuit}$	29	33	$3^{\blacklozenge}$	32	30

5 Conclusions

In this paper, we proposed the use of the semantic distance approximated using word embeddings as a heuristics to guide search in a KG. The introduced bidirectional A*-like search algorithm employs a parameterized evaluation function that considers path length *and* semantic distances.

For further optimization, we plan to explore which evaluation function configuration yields the best results with respect to a larger benchmark. Here, we assume that the semantic distance between v_{source} and v_{target} entity can be used as a decision criterion to allow for a query dependent parameter selection. Dropping unpromising paths from the path queue in a beam search fashion may enable to handle entities with many incoming/outgoing edges. A performance increase may also be achieved by using a manually trained FT model tailored to a particular domain, in contrast to the currently used pre-trained model. Furthermore, we want to extend our approach to also consider edge labels in the evaluation function and lift the support from dual-entity queries to multi-entity queries. Enriching our purely semantic heuristics with information on the graph's local structure, e.g., using RDF2Vec [10], is also worth to explore.

References

1. Auer, S., et al.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguistics **5**, 135–146 (2017). https://transacl.org/ojs/index.php/tacl/article/view/999
3. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications. Macmillan Education, UK (1976). https://doi.org/10.1007/978-1-349-03521-2
4. Dietz, L., Kotov, A., Meij, E.: Utilizing knowledge graphs for text-centric information retrieval. In: Collins-Thompson, K., Mei, Q., Davison, B.D., Liu, Y., Yilmaz, E. (eds.) 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR 2018), pp. 1387–1390. ACM (2018). https://doi.org/10.1145/3209978.3210187
5. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. In: Martin, M., Cuquet, M., Folmer, E. (eds.) 12th International Conference on Semantic Systems (SEMANTiCS 2016). CEUR Workshop Proceedings, vol. 1695. CEUR-WS.org (2016). http://ceur-ws.org/Vol-1695/paper4.pdf
6. Gomaa, W.H., Fahmy, A.A., et al.: A survey of text similarity approaches. Int. J. Comput. Appl. **68**(13), 13–18 (2013)
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations (ICLR 2013) (2013). http://arxiv.org/abs/1301.3781
8. Paulheim, H.: Knowledge graph refinement: a survey of approaches and evaluation methods. Semantic Web **8**(3), 489–508 (2017). https://doi.org/10.3233/SW-160218

9. Ringler, D., Paulheim, H.: One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In: Kern-Isberner, G., Fürnkranz, J., Thimm, M. (eds.) 40th Annual German Conference on AI (KI 2017). LNCS, vol. 10505, pp. 366–372. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67190-1_33
10. Ristoski, P., Paulheim, H.: Rdf2vec: RDF graph embeddings for data mining. In: Groth, P.T., et al. (eds.) The Semantic Web - ISWC 2016–15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9981, pp. 498–514 (2016). https://doi.org/10.1007/978-3-319-46523-4_30
11. Singhal, A.: Introducing the Knowledge Graph: Things, not Strings (2012). https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html
12. Stekkelpak, Z., Simonyi, G.: Providing customized content in knowledge panels, uS Patent App. 13/750, 354, 18 June 2015
13. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014). https://doi.org/10.1145/2629489