LAUREN - Knowledge Graph Summarization for Question Answering

Rricha Jalota
Data Science Group
Paderborn University, Germany
Email:rricha.jalota@uni-paderborn.de

Daniel Vollmers
Data Science Group
Paderborn University, Germany
Email:daniel.vollmers@uni-paderborn.de

Diego Moussallem
Data Science Group
Paderborn University, Germany
Email:diego.moussallem@uni-paderborn.de

Axel-Cyrille Ngonga Ngomo
Data Science Group
Paderborn University, Germany
Email:axel.ngonga@uni-paderborn.de

***Abstract*—Besides the challenge that a human can ask one question in many different ways, a key aspect in Question Answering approaches over Knowledge Graphs (KGQA) is to deal with the vast amount of information present in the knowledge graphs. Modern real-world knowledge graphs contain nearly millions of entities and relationships. Additionally, they are enriched with new facts every day. However, not all facts are relevant for answering particular questions, thus fostering several challenges to KGQA systems, which require interpretable and query-able data. One solution to filtering the extra data in knowledge graphs is to rely on graph summarization techniques. Graph-based summarization approaches aim to resize knowledge graphs to be more concise and precise by storing only relevant information. In this paper, we propose a framework named LAUREN that applies different summarization techniques on knowledge graphs to be used in KGQA systems. Our experiments show that LAUREN summarizes large knowledge graphs such as DBpedia by 2 million entities and its summarization still achieves the same performance on both question answering and linking tasks compared to the complete DBpedia.**

I. Introduction

Question Answering over Knowledge Graphs (KGQA) have recently achieved outstanding progress, catalysed by the release of different Neural Network (NN) architectures. Despite these notable advancements, a key aspect in Question Answering (QA) based on Knowledge Graphs (KGs) is to deal with the vast amount of information present in the KGs. Modern real-world knowledge graphs contain nearly millions of entities and are enriched with new facts every day. However, not all facts are relevant for answering particular questions, thus fostering several challenges to KGQA systems, which require interpretable and query-able data as well as time-efficient and accurate answers.

State of the art KGQA-systems rely on large KGs such as DBpedia, which contain millions of entities and relations. Considering that KGQA-systems process the data contained in these KGs to gather the correct answer for given questions, reducing the amount of unnecessary triples, i.e, out-of-domain data, may play a crucial role in improving the performance of these systems with respect to the processing speed and hardware requirements. Furthermore, the reduction of extra data may improve the performance of triple stores used in QA-Systems that are based on semantic parsing. To the best of our knowledge, no previous work investigated the direct impact of resizing KG data to be applied on QA task. Thus, we study graph summarization techniques to filtering extra data in KGs. Graph-based summarization approaches aim to resize knowledge graphs to be more concise and precise by storing only relevant information.

In this paper, we propose a framework named LAUREN that implements existing graph summarization approaches, HITS and SALSA, along with filtering techniques and study their impact on KGQA systems. With this aim, we apply LAUREN on the English DBpedia KG. We hence evaluate and compare two reduced variations of DBpedia, one based on HITS and another based on SALSA with the original English DBpedia. The evaluation is carried out on the tasks of QA and Entity Linking (EL) along with Relation Linking (RL) which are QA-sub-tasks.

The main contributions of this paper can be summarized as follows:

- We present a knowledge graph summarization framework, named LAUREN, focused on QA task.
- LAUREN summarizes large knowledge graphs such as DBpedia by 2 million entities, which corresponds to 50% size-reduction, and its summarization still achieves a similar performance on both question answering and linking tasks compared to the complete DBpedia Knowledge Graph.
- We studied the integration of summarized graphs in EL and QA Frameworks such as MAG, FALCON, and QANARY.
- Our results show, that the components based on the summarized graphs achieve the same results as the components based on the original graphs, by an improvement of the processing speed.

The version of LAUREN used in this paper and also all experimental data are publicly available[1].

[1]https://github.com/dice-group/LAUREN

II. Related Work

The use of large encyclopedic graphs for knowledge representation and reasoning comes with its own set of limitations. For one, it is a time and resource-intensive task to process billions of entities to answer a single query. Secondly, they require considerable storage space. As graphs are increasingly becoming the preferred choice for data representation and storage, graph summarization and compression techniques play a vital role in low-resource application scenarios. While graph summarization techniques aggregate nodes having similar structural characteristics to represent a graph with reduced RAM requirements, graph compression techniques leverage various encoding techniques to reduce their storage space on the disk. Here we briefly summarize the existing techniques for static graph summarization and how they have been applied to knowledge graphs so far.

Most of the graph summarization techniques found in the literature majorly rely on at least one of the following methods: *grouping or aggregation* [1], *bit compression* [2], [1], *simplification or sparsification* [3], [4]. In grouping-based approaches, nodes are recursively aggregated into *supernodes* based-on either structure or other application dependent properties. Some studies also utilize clustering techniques and map the resulting dense clusters to supernodes. Bit-compression based techniques, on the other hand, use summaries to minimize the number of bits needed to describe an input graph, while simplification-based approaches for graph summarization simply remove the less important nodes/edges from the input graph based on a pre-dominant criteria. A further in-depth account on each of these techniques can be found in the survey by Y. Liu et al [5].

Orthogonal to the above approaches, GLIMPSE - a personalized knowledge graph summarization framework [6] employs graph sampling to preserve graph-specific properties in KG samples while using past user queries as seeds to infer other entities and relations of potential interest to the user. Other related studies revolve either around entity-summarization or fact-contextualization, wherein an entity or a fact from the KG is given as an input and the output is a selection of facts describing the entity or the fact from the underlying knowledge base. In contrast, we follow a simplification-based approach to summarize the entire knowledge graph - independent of any given query or entity. In the subsequent sections, we elaborate our approach. To the best of our knowledge, no previous work investigated the impact of graph summarization on a large KG for KGQA.

III. The Framework

The intuition behind LAUREN is that important entities are linked by a lot of nodes while irrelevant entities have just a small set of incoming links. Thus, we utilize the concept of Hubs and Authorities to identify entities that are essential in a KG. Additionally, not all links in a graph carry the same weightage, since nodes that have links to several authorities, are more important in a graph, than the nodes that have just a few outgoing links. These nodes are called Hubs. Figure 1 shows the basic steps of LAUREN. For a KG, LAUREN first computes the Hub (h) and Authority (a) scores. With the computed scores, we calculate the mean authority score t, which is used as a threshold for filtering entities. In the last step all nodes in the graph, which are below the threshold are filtered. The threshold is the mean value of Authoritative scores. The remaining nodes define the summarized KG. A

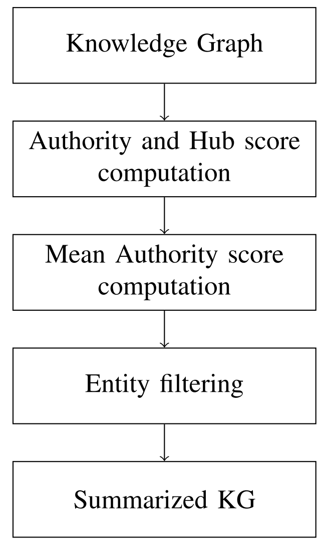

Fig. 1: Graph Summarization methodology

scalable and precise algorithm is essential to calculate the Hub and Authority scores. Thus, LAUREN relies on two popular algorithms from the literature, which are described in the following sections.

A. HITS

The Hyperlink-Induced Topic Search (HITS) algorithm [7] works by assigning Hub- and Authority-scores to each node.

The Authority score a_p for the node p is calculated with Formula 1.

$$a_p \leftarrow \sum_{q:(q,p)\in E} h_q \tag{1}$$

Where h_q is the Hub-score for node q, where there is an directed edge from node q to the node p in the Graph E. Analogously the Hub weight h_p for a node p is calculated with Formula 2.

$$h_p \leftarrow \sum_{q:(q,p)\in E} a_q \tag{2}$$

Where a_q is the Authority-weight for node q, which is linked by node p [7]. For calculating the weights for the whole graph, an iterative algorithm is used, which updates the Hub- and Authority-weights simultaneously until convergence [7].

B. SALSA

The SALSA algorithm is a stochastic approach for calculating Hub- and Authority-scores based on random walks. Based on a site collection C, a bipartite undirected graph $G = (V_h, V_a, E)$ is built:

- $V_h = s_h | s \in C$ and $out - degree(s) > 0$ (the hub side of G)
- $V_a = s_a | s \in C$ and $in - degree(s) > 0$ (the authority side of G)
- $E = (s_h, r_a) | s \to r$ in C.

Each node $s \in C$ is represented by two nodes s_h and s_a and each link $s \to r$ is represented by an undirected edge connecting s_h and s_a.[8] Based on the bipartite graph, two distinct random walks are performed, where each step consists of traversing two edges of the Graph. Thus, the random walks will only visit one side of G. Each random walk starts off from a different side of the Graph. The outcome of the random walks are two different Markov chains. The Markov chain that visits the authority side of G defines the authority matrix A:

$$a_{i,j} = \sum_{k|(k_h,i_a),(k_h,j_a)\in G} \frac{1}{deg(i_a)} \times \frac{1}{deg(k_h)}$$

and the Markov chain that visits the Hub side of the Graph defines the Hub matrix H:

$$h_{i,j} = \sum_{k|(i_h,k_a),(j_h,k_a)\in G} \frac{1}{deg(i_h)} \times \frac{1}{deg(k_a)}$$

A positive transition probability $a_{i,j} > 0$ implies that a certain page points to both pages i and j,and hence page j is reachable from page i by two steps retracting along the link $h \to i$ and then following the link $h \to j$ [8]. Based on the authority Matrix a unique real positive eigenvalue $\mu(A)$ can be calculated. The corresponding unit eigenvector which corresponds to $\mu(A)$ whose first non-zero coordinate is positive is called the principal eigenvector ($v_{\mu(A)}$) of the Authority Matrix. For more details, see R. Lempel and S. Moran [8]. The values of $v_{\mu(A)}$ are the Authority-scores for the nodes of the Graph G. The size of the resulting summarized graphs have been reported in Table I.

TABLE I: KG-size after applying Summarization Techniques

Knowledge Graph	Summarized Nodes	Total Nodes
ORIGINAL-DBpedia	–	3,116,745
HITS-DBpedia	1,235,850	**1,880,895**
SALSA-DBpedia	1,027,786	2,088,959

IV. Evaluation on QA-subtasks

In this section, we investigate the performance of LAUREN on EL and RL, which are both important QA-Subtasks for many QA-Systems. Our evaluation utilizes the fair evaluation Framework GERBIL [9] to measure the performance on all tasks. For all experiments we used DBpedia [10] as reference KG[2].

[2]https://wiki.dbpedia.org/downloads-2016-10

A. Evaluation on EL

One important subtask for Question Answering is EL, also known as Named Entity Disambiguation (NED). The goal of an EL approach is: Given a piece of text, a reference knowledge graph K and a set of entity mentions in that text, map each entity mention to the corresponding resource in K. The entity mentions are generated in a previous entity recognition step (NER). In contrast to NER, EL depends highly on underlying KBs. Thus, in our first evaluation step, we integrated the summarized graphs in the EL framework MAG [11]. MAG consists of two basic steps: (1) candidate generation and (2) disambiguation. For both steps, the basic data structure is an inverted index named Triple-index, which contains all the triples of a KG. For the candidate generation, the Triple-index is queried with the labels of the named entities of the input text. The output of this step is a set of candidate URIs for each of the given entities. These candidates represent the nodes of a local graph. For the disambiguation step, edges and more related nodes are extracted from the Triple-index and added in the local graph. Afterwards, the HITS algorithm is applied to identify the best node for each of the candidate entities. To integrate the summarized graphs into the framework, we generated two new Triple-indexes, one for each of the summarized graphs. The rest of the framework needs no further adjustments. For each of the Triple-indexes, we set up one MAG-instance. To compare the performance of LAUREN, on EL, we evaluated the performance of the two instances on the fair evaluation framework GERBIL. Furthermore, we compared the results with the performance of the original system with uncompressed graph. We tested each system on the six different EL benchmarks, presented in Table II.

TABLE II: EL datasets statistics

Dataset	Topic	Documents	Entities
ACE2004 [12]	news	57	253
AQUAINT [13]	news	50	747
DBpediaSpotlight [14]	news	58	330
MSNBC [15]	news	20	747
N3-RSS-500 [16]	mixed	500	1000
N3-Reuters-128 [16]	news	128	880

Results The results on the EL benchmarks presented in Table III show, that the systems that apply the summarized Graphs achieve comparable performance on each of the six Benchmark-datasets. The difference between SALSA and HITS is relatively small w.r.t F-measure, Precision and Recall. Moreover, the average answer time was better for the summarized Graphs compared to the original one. The reason for that is, the index sizes for the summarized Triple-indexes are smaller than the size of the original Triple-index.

B. Evaluation on RL

Another very important subtask for Question Answering is RL. The goal of an RL approach is as follows: Given a piece of text, a reference knowledge graph K and a set of mentions in that text, map mentions which are identified as predicates

TABLE III: Results of MAG for multiple EL Data sets on the original and the summarized graphs.

Graph	Dataset	Precision	Recall	F-Measure	Avg. Time (milliseconds)
ORIGINAL-DBpedia	ACE2004	0.74	0.74	**0.74**	878.81
ORIGINAL-DBpedia	AQUAINT	0.67	0.67	**0.67**	1,064.24
ORIGINAL-DBpedia	DBpediaSpotlight	0.67	0.68	**0.67**	889.18
ORIGINAL-DBpedia	MSNBC	0.72	0.72	**0.72**	2,133.79
ORIGINAL-DBpedia	N3-RSS-500	0.69	0.69	0.69	141.39
ORIGINAL-DBpedia	N3-Reuters-128	0.71	0.71	**0.71**	252.01
HITS-DBpedia	ACE2004	0.73	0.73	0.73	770.00
HITS-DBpedia	AQUAINT	0.66	0.66	0.66	**906.84**
HITS-DBpedia	DBpediaSpotlight	0.67	0.68	**0.67**	783.37
HITS-DBpedia	MSNBC	0.71	0.71	0.70	**1,758.26**
HITS-DBpedia	N3-RSS-500	0.69	0.69	0.69	137.67
HITS-DBpedia	N3-Reuters-128	0.70	0.70	0.70	230.33
SALSA-DBpedia	ACE2004	0.73	0.73	0.73	**740.38**
SALSA-DBpedia	AQUAINT	0.66	0.66	0.66	966.24
SALSA-DBpedia	DBpediaSpotlight	0.68	0.68	0.68	**739.09**
SALSA-DBpedia	MSNBC	0.72	0.72	0.71	1,867.00
SALSA-DBpedia	N3-RSS-500	0.70	0.70	**0.70**	**127.86**
SALSA-DBpedia	N3-Reuters-128	0.70	0.70	0.70	**228.82**

among other entity mentions to the corresponding relation resource in K. We evaluated the performance of the compressed graphs on the EL and RL approach, FALCON [17].

For RL, FALCON uses the DBpedia Knowledge Graph to check whether the identified entities and candidate relations exist in the KG as triples. If a triple is found, the candidate relation is returned as the answer. We used two different settings for evaluation with FALCON. In the first setting, we used FALCON as it is and just changed the DBpedia endpoints to refer to the original graph and the compressed HITS and SALSA graphs. The results for this experiment are reported in Table IV. In the second setting, we replaced the EL component of FALCON with MAG and compared the results with different DBpedia endpoints, as stated above. We tested each setting on the QALD-8[18](50 questions) and the QALD-9[19](413 questions) datasets using GERBIL. Although QALD-8 and QALD-9 are datasets designed for QA, GERBIL is capable of measuring related approaches on RL. The results, are presented in Table IV.

Results Table IV shows that, for the original FALCON approach, the results on the QALD-8 dataset are the same for the summarized graphs as for the original DBpedia graph. Furthermore, the average time for a document decreased especially for the HITS-summarized graph. For QALD-9, the HITS-summarization algorithm improves the performance of the FALCON system, while summarization with the SALSA algorithm is a little lower compared to the original system. The reason for that might be, SALSA is more restrictive regarding the identification of Hubs and Authorities, so some rare entities that are used in QALD-9 queries are removed from the KB by filtering the entities based on the Authority-scores generated by the SALSA algorithm. Interestingly, the combination of MAG and FALCON decreased the performance of RL in FALCON, while for the EL task, the combined instances show the same results as the original approach. The reason for that lies in the incompatibilities of the implementations of the two systems. The entity and relation linking in FALCON are intertwined and hence, isolating the entity linking component results in a decreased performance. We plan to exploit how to cope with the communication issue in future research as well as handle domain-specific rare entities which are important for QA.

V. Evaluation on Question Answering

In this section, we describe our evaluation with our summarized graphs on Question Answering systems. Question Answering is the task of answering natural language questions, such as *Who is the president of the United States of America?* by querying a structured database, in the case of KGQA, the queries are over a KG. For our evaluation, we tested our summarized graphs on the two KBQA-Systems: QANARY and TeBaQA.

QANARY[20] consists of a pipeline, which enables the integration of different QA-Components in a QA System. This makes it easy to compare different QA-Components. For testing our compressed graphs, we integrated our MAG instances for the EL subtask in the QANARY-pipeline. Thus, we created three QANARY instances, one for each MAG instance and generated the results on the QALD-8[18](50 questions) and the QALD-9[19](413 questions) data sets with the QA-Benchmarking System GERBIL-QA[21].

Additionally, we tested the two summarized graphs on the Question Answering System, TeBaQA. TeBaQA is a system which is based on learning SPARQL templates from past benchmark challenges and filling them subsequently [19]. TeBaQA uses a joint approach for EL and RL based on a KG. Comparable to MAG and Falcon, it uses an inverted index for candidate generation on entities and relations and fills triple Templates based on the connection between entities in the KG.[3] For the evaluation, we created three instances of

[3]The implementation can be derived from https://github.com/dice-group/TeBaQA

TABLE IV: Results of Relation and EL with FALCON on multiple QA datasets with original and summarized graphs. In all the settings, FALCON was used as the Relation Linking component. The Average Time is in milliseconds.

Graph	Dataset	Entity Linking (EL)	EL-Precision	EL-Recall	EL-F-Measure	RL-Precision	RL-Recall	RL-F-Measure	Avg. Time
Original-DBpedia	QALD8	FALCON	0.95	0.95	0.95	0.24	0.34	0.28	387.47
	QALD8	MAG-Orig	0.95	0.95	0.95	0.06	0.07	0.06	84.68
	QALD8	MAG-HITS	0.95	0.95	0.95	0.06	0.07	0.06	81.07
	QALD8	MAG-SALSA	0.95	0.95	0.95	0.06	0.07	0.06	241.60
Original-DBpedia	QALD9	FALCON	0.76	0.79	0.77	0.50	0.51	0.50	2,861.39
	QALD9	MAG-Orig	0.74	0.73	0.73	0.39	0.37	0.38	324.00
	QALD9	MAG-HITS	0.72	0.71	0.72	0.39	0.37	0.38	536.09
	QALD9	MAG-SALSA	0.72	0.71	0.72	0.39	0.37	0.38	543.50
HITS-DBpedia	QALD8	FALCON	0.95	0.95	0.95	0.25	0.34	**0.29**	**198.18**
	QALD8	MAG-Orig	0.95	0.95	0.95	0.04	0.05	0.04	284.36
	QALD8	MAG-HITS	0.95	0.95	0.95	0.04	0.05	0.04	213.54
	QALD8	MAG-SALSA	0.95	0.95	0.95	0.04	0.05	0.04	223.19
HITS-DBpedia	QALD9	FALCON	0.74	0.77	0.75	0.45	0.62	**0.52**	**1,321.50**
	QALD9	MAG-Orig	0.74	0.73	0.73	0.37	0.36	0.37	665.55
	QALD9	MAG-HITS	0.72	0.71	0.72	0.37	0.36	0.37	727.92
	QALD9	MAG-SALSA	0.72	0.71	0.72	0.37	0.36	0.37	594.91
SALSA-DBpedia	QALD8	FALCON	0.95	0.95	0.95	0.25	0.34	**0.29**	331.33
	QALD8	MAG-Orig	0.95	0.95	0.95	0.04	0.05	0.04	129.15
	QALD8	MAG-HITS	0.95	0.95	0.95	0.04	0.05	0.04	325.52
	QALD8	MAG-SALSA	0.95	0.95	0.95	0.04	0.05	0.04	319.53
SALSA-DBpedia	QALD9	FALCON	0.74	0.77	0.75	0.45	0.46	0.46	2,526.58
	QALD9	MAG-Orig	0.74	0.73	0.73	0.37	0.36	0.37	530.46
	QALD9	MAG-HITS	0.72	0.71	0.72	0.37	0.36	0.37	838.74
	QALD9	MAG-SALSA	0.72	0.71	0.72	0.37	0.36	0.37	612.63

TeBaQA. One for each summarized Graph and one for the original KB.

Results On the two test datasets, the three instances based on QANARY achieved the same results w.r.t. Recall, Precision and QALD-F-Measure. The average time is nearly the same for all datasets. Moreover, on the larger QALD-9 dataset, QANARY takes lesser time with the summarized graphs. For the TeBaQA system, the influence of the summarization is much larger compared to QANARY. Especially on QALD-9, the overall performance of TeBaQA is 20% lesser with the HITS-Graph and 5% lesser with Salsa-Graph. However, the average time decreased significantly for both of the summarized Graphs. The results can be derived from table V. The reason for different influence of the summarization on TeBaQA and QANARY is, that TeBaQA uses information of the KB not only in the EL-step, but also for RL and for Query Generation. QANARY only uses the KB in the EL step. The results show that, summarization techniques support the performance of QA-Systems with regard to time efficiency and hardware requirements. In our experiments, the performance of the tested QA-Systems decreased only for TeBaQA on the QALD-9 datset, for all other experiments the performance with the summarized Graphs is comparable or equal as for the orignal version of the graphs. We plan to investigate this in further research.

VI. Conclusion

Our results show that with knowledge graph summarization, we not only achieve comparable results but also an improvement in the processing time of the queries. Across all evaluation tasks, the compressed SALSA graph almost consistently outperformed the HITS and the original graph in terms of processing time. In our future work, we plan to further evaluate the performance of these graphs on QA tasks in low-resource environments, such as cellphones, wherein efficient handling of hardware limitations would play an important role. Moreover, we plan to implement more graph summarization techniques as well as exploit other large KGs such as Wikidata and YAGO.

Acknowledgments

Research funded by the German Federal Ministry of Economics and Technology (BMWI) in the project RAKI (no. 01MD19012D) and by the H2020 KnowGraphs (GA no. 860801). This work also has been supported by the German Federal Ministry of Education and Research (BMBF) within the project DAIKIRI under the grant no 01IS19085B as well as by the German Federal Ministry for Economic Affairs and Energy (BMWi) within the project SPEAKER under the grant no 01MK20011U.

References

[1] H. Seo, K. Park, Y. Han, H. Kim, M. Umair, K. Khan, and Y. Lee, "An effective graph summarization and compression technique for a large-scaled graph," *J. Supercomput.*, vol. 76, no. 10, pp. 7906–7920, 2020. [Online]. Available: https://doi.org/10.1007/s11227-018-2245-5

[2] N. Shah, D. Koutra, T. Zou, B. Gallagher, and C. Faloutsos, "Timecrunch: Interpretable dynamic graph summarization," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, L. Cao, C. Zhang, T. Joachims, G. I. Webb, D. D.

TABLE V: Results of QANARY and TeBaQA for multiple Question Answering Datasets on the original and the summarized graphs.

System	Graph	Dataset	Precision	Recall	QALD F-Measure	Avg. Time(milliseconds)
QANARY	ORIGINAL-DBpedia	QALD-8	0.24	0.23	0.37	**3,422.83**
QANARY	HITS-DBpedia	QALD-8	0.24	0.23	0.37	3,446
QANARY	SALSA-DBpedia	QALD-8	0.24	0.23	0.37	3,480.68
QANARY	ORIGINAL-DBpedia	QALD-9	0.14	0.15	0.25	4,019.60
QANARY	HITS-DBpedia	QALD-9	0.14	0.14	0.24	**3,955.34**
QANARY	SALSA-DBpedia	QALD-9	0.14	0.14	0.24	4,017.52
TeBaQA	ORIGINAL-DBpedia	QALD-8	0.48	0.49	**0.56**	15,068.20
TeBaQA	HITS-DBpedia	QALD-8	**0.50**	**0.51**	**0.56**	12,140.73
TeBaQA	SALSA-DBpedia	QALD-8	0.48	0.49	0.54	11,620.83
TeBaQA	ORIGINAL-DBpedia	QALD-9	**0.27**	**0.27**	**0.35**	9,524.67
TeBaQA	HITS-DBpedia	QALD-9	0.22	0.23	0.33	8.843.98
TeBaQA	SALSA-DBpedia	QALD-9	0.20	0.21	0.30	8,314.00

Margineantu, and G. Williams, Eds. ACM, 2015, pp. 1055–1064. [Online]. Available: https://doi.org/10.1145/2783258.2783321

[3] Z. Shen, K.-L. Ma, and T. Eliassi-Rad, "Visual analysis of large heterogeneous social networks by semantic and structural abstraction," *IEEE transactions on visualization and computer graphics*, vol. 12, no. 6, pp. 1427–1439, 2006.

[4] C. Dunne and B. Shneiderman, "Motif simplification: improving network visualization readability with fan, connector, and clique glyphs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 3247–3256.

[5] Y. Liu, T. Safavi, A. Dighe, and D. Koutra, "Graph summarization methods and applications: A survey," *ACM Comput. Surv.*, vol. 51, no. 3, pp. 62:1–62:34, 2018. [Online]. Available: https://doi.org/10.1145/3186727

[6] T. Safavi, C. Belth, L. Faber, D. Mottin, E. Müller, and D. Koutra, "Personalized knowledge graph summarization: From the cloud to your pocket," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 528–537.

[7] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, p. 604–632, Sep. 1999. [Online]. Available: https://doi.org/10.1145/324133.324140

[8] R. Lempel and S. Moran, "The stochastic approach for link-structure analysis (salsa) and the tkc effect1abridged version1," *Computer Networks*, vol. 33, no. 1, pp. 387 – 401, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128600000347

[9] R. Usbeck, M. Röder, A. N. Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, and L. Wesemann, "GERBIL: general entity annotator benchmarking framework," in *Proceedings of the 24th International Conference on World Wide Web, WWW, May 18-22*, Florence, Italy, 2015, pp. 1133–1143.

[10] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[11] D. Moussallem, R. Usbeck, M. Röder, and A. N. Ngomo, "Entity linking in 40 languages using MAG," in *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*, 2018, pp. 176–181. [Online]. Available: https://doi.org/10.1007/978-3-319-98192-5_33

[12] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to Wikipedia," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 1375–1384. [Online]. Available: https://www.aclweb.org/anthology/P11-1138

[13] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*. New York, NY, USA: ACM, 2008, pp. 509–518. [Online]. Available: http://www.cs.waikato.ac.nz/~dnk2/publications/CIKM08-LearningToLinkWithWikipedia.pdf

[14] P. N. Mendes, M. Jakob, A. Garcia-Silva, and C. Bizer, "DBpedia Spotlight: shedding light on the web of documents," in *Proceedings of the 7th international conference on semantic systems*. ACM, 2011, pp. 1–8.

[15] S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 708–716. [Online]. Available: https://www.aclweb.org/anthology/D07-1074

[16] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both, "N3 - a collection of datasets for named entity recognition and disambiguation in the nlp interchange format," 03 2014.

[17] A. Sakor, I. O. Mulang, K. Singh, S. Shekarpour, M.-E. Vidal, J. Lehmann, and S. Auer, "Old is gold: Linguistic driven approach for entity and relation linking of short text." in *NAACL-HLT (1)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 2336–2346. [Online]. Available: http://dblp.uni-trier.de/db/conf/naacl/naacl2019-1.html#SakorMSSV0A19

[18] R. Usbeck, A. N. Ngomo, F. Conrads, M. Röder, and G. Napolitano, "8th challenge on question answering over linked data (QALD-8)," in *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018)*, ser. CEUR Workshop Proceedings, vol. 2241. CEUR-WS.org, 2018, pp. 51–57. [Online]. Available: http://ceur-ws.org/Vol-2241/paper-05.pdf

[19] R. Usbeck, R. H. Gusmita, A. N. Ngomo, and M. Saleem, "9th challenge on question answering over linked data (QALD-9)," in *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018)*, ser. CEUR Workshop Proceedings, vol. 2241. CEUR-WS.org, 2018, pp. 58–64. [Online]. Available: http://ceur-ws.org/Vol-2241/paper-06.pdf

[20] A. Both, D. Diefenbach, K. Singh, S. Shekarpour, D. Cherix, and C. Lange, "Qanary – a methodology for vocabulary-driven open question answering systems," in *The Semantic Web. Latest Advances and New Domains*, H. Sack, E. Blomqvist, M. d'Aquin, C. Ghidini, S. P. Ponzetto, and C. Lange, Eds. Cham: Springer International Publishing, 2016, pp. 625–641.

[21] R. Usbeck, M. Röder, M. Hoffmann, F. Conrads, J. Huthmann, A. N. Ngomo, C. Demmler, and C. Unger, "Benchmarking Question Answering Systems," *Semantic Web*, vol. 10, no. 2, pp. 293–304, 2019. [Online]. Available: http://www.semantic-web-journal.net/system/files/swj1578.pdf