Ranking Methods for the Web of Data

Vaibhav Kasturia
L3S Research Center
Leibniz Universität Hannover
Hannover, Germany
kasturia@l3s.de

April 27, 2017

Abstract

Ranking resources is a typical task that occurs in Information Retrieval. Users performing Web Search tend to pay attention to only the top results and hence it becomes very important to have a good ranking approach. In document search link based analysis is commonly performed to establish relationship between documents. As there is a shift to the Web of Linked Data there needs to be an adaptation of the current ranking approaches for unstructured data for the Linked Data. RDF, RDFS and OWL allow us to express Semantic relationships in the Linked Data Web. These semantic relationships must be exploited for ranking the results of structured queries.

1 Introduction

A survey of the ranking approaches on the web of linked data shows that there is a lack of consensus in the problem of ranking structured data. This article categorises the existing ranking approaches and describes the features exploited by each one of them in performing ranking. We exploit some parts of the approaches in our ranking model for search on the semantic layer of the NYT archives. In general, most of the exploited features in existing ranking approaches fall into provenance and context. The related ranking approaches fall into two categories: ranking on results of unstructured data and ranking on results of structured data. The approaches ranking results of structured data are further categorized based on whether they permit keyword search.

2 Ranking on Unstructured Data

Ranking on unstructured data or documents in the existing search engines is performed by either Content-based analysis (ex. tf-idf based approach) or Link-based analysis (ex. PageRank approach, HITS). Ranking approaches on structured data typically adapt the analysis methods for unstructured data to structured data. Likewise, we exploit part of

the HistDiv approach for ranking NYT articles in our approach for ranking results for structured queries on semantic layer of NYT archives.

2.1 HistDiv Approach

Singh et al. [1] designed a system to model the need of Historians for documents related to the history of an Entity or an Event. The system deals with the ranking of such documents from the NYT Archives by retrieving important aspects from important points in time. As an example, consider a historian interested in the history of the US politician Rudolph Giulaini. Initially he would want to get an overview of the politicians life: his rise to prominence becoming mayor of New York City in 1994, his tough stance on crime for which he became famous in 1996, his drive against prostrate cancer in 2000, his run for president in the US Presidential Elections of 2008 and so on. Once he gets an overview and context, he can then delve deeper into the time periods he is interested in and move to more specific results. This system is precisely modelled on this behaviour and provides broad results initially with more focus on recall. The system does this by providing users NYT articles containing important aspects from important points in time. The notion of important aspects is based on the association of groups of entities in the articles and the probability distribution of these entities in a time interval. The system builds a Time-Aspect Space treating time diversity and topical diversity as separate and relies upon prior probabilities to return user important NYT articles. Our system for ranking documents given a list of entities and time period uses a similar approach for identifying important aspects for an entity. Further, the results of the HistDiv System could be used as baselines for our system.

3 Ranking on Structured Data

The systems ranking results on structured data are classified into two categories: those not supporting keyword search mentioned under Semantic search and those supporting keyword search mentioned under Keyword search. Most of the approaches described similar to unstructured data rely on Content-based analysis or Link-based analysis. RDF data too contains literals associated with text, for example, *rdfs:label* or *dbpprop:abstract*. Content-based analysis measures of ranking in structured data take the content associated with these literals to establish relationships based on string comparisons, tf-idf approach, and so on. For example, a way of establishing relationship between two entities would be to take into account the associated text labels like checking whether *rdfs:label* of one entity is occuring in the *dbpprop:abstract* of another entity. Link-based analysis measures on structured data are usually extensions of PageRank approach [2] or HITS approach [3]. These extensions in general perform three types of analysis on structured data: Weighted link analysis, Hierarchical link analysis and Semantic web link analysis. Methodologies based on Weighted link analysis assign more weights to certain kinds of links based on relevance during the ranking computation. In Hierarchical link analysis super node relationships are considered first and then relationships among

sub nodes are considered. Semantic web link analysis methodologies exploit semantic relationships during the ranking process.

3.1 Semantic search

The approaches mentioned in this section are adaptations of the Learning to Rank(LTR) approach and Language Model approach for unstructured data, among other approaches. Learning to Rank approach is a machine learning approach to deduce ranking model from training data. This approach has gained a lot of popularity as it is applicable for both structured and unstructured data without needed customization. However, one drawback of this approach is the requirement of training data which is often unavailable to authors. Language Model approach is query dependent approach as the ranker ranks a set of items according to the user input. The calculation performed is on the fly with no previous result or implicit structure of the data used.

3.1.1 Query-Independent Learning to Rank for RDF Entity Search

Dali et al. [4] propose a query independent LTR approach for RDF entity search. The query independent features extracted by them are can be categorized as: features extracted from RDF graph, search engine based features and centrality-based features which include PageRank and HITS. These features are represented as vectors and the ranking score is a summation of the weights of the feature vectors. This total score is then compared to the target features to estimate the closeness between extracted features and target features and derive a ranking score based on this closeness. The target features act as a ground truth and these are obtained from access logs. The authors propose the use of access logs as ground truth or training data based on showing that a direct co-relation exists between number of page visits and human relevance judgements. The classification of results is done using SVM with soft-margin approach. For our approach for ranking documents on NYT archives what we lack is access logs which would act as ground truth or training data if we built a system similar to this system. However, we could believe that the author's assumption of direct co-relation is correct and try to obtain ground truth from human relevance judgements of the NYT articles using crowdsourcing and use that as training data for the classifier.

3.1.2 Query-Independent Learning to Rank RDF Entity Results of SPARQL Queries

Latifi and Nematbakhsh [5] propose the same approach as Dali et al. [4] as outlined in the previous section. They claim that the centrality-based features used by Dali et al. do not use the semantics of links. They suggest the use of another query independent feature called Information Content(IC) which they define as the information associated with a given entity. They say that although the features obtained from RDF graphs in previous system provide a very good ranking, however, they take a huge amount of time to extract when compared to the use of Information Content(IC) as a feature and

hence suggest the use of Information Content feature for ranking. The authors made use of Wikipedia article traffic statistics[1] from July 2013 to June 2014 as ground truth. For our approach if we decide to build a query independent ranking model we could also make extract Information Content as a feature. Further, we could look whether we could make any possible use of the Wikipedia article traffic statistics in building our ground truth.

3.1.3 OntologyRank

This algorithm was first introduced in Finin et al. [6] and was subsequently mentioned in Ding et. al. [7] OntologyRank is the approach used by their Semantic web search engine *Swoogle*. This approach identifies Semantic Web Ontologies(SWOs) in Semantic Web Documents(SWDs) and is further able to rank terms in an Ontology based on their popularity. It is also able to list the most popular properties for a class by ranking class-property bonds which is useful for users desiring maximum data visibility. OntologyRank calculates relevance of SWDs taking into account the following relationships: *imports(A,B), uses-term(A,B), extends(A,B) and asserts(A,B)* where A and B are SWDs. The ranking score is computed based on these relationships and the algorithm is PageRank like but boosted to identify ontologies. Thus, the authors show that their variation of PageRank helps their Swoogle search engine detect ontologies better than Google. Another aspect of this algorithm is that it does not take into account provenance or context information. Our system would not be requiring ontology ranks and hence this approach is currently not useful. This approach could be of use in a scenario where we have multiple ontologies for the same thing and so there arises a need for ranking ontologies and terms used by the ontologies.

3.1.4 PopRank

Nie et al. [8] designed PopRank, another variation of PageRank that assigns weights to links among Web Objects depending on the relationship types between objects. The authors use Ranking Lists made by domain experts in this mechanism of assigning weights. Further, the authors do not assign weights to the whole graph as that would be expensive. They extract a subset of the graph based on domain and then assign the links weights to the subgraph. The authors say that due to their assignment mechanism of link weights, an accuracy increment of 50 per cent is observed over baseline PageRank. We could exploit their approach of extracting subset based on domain and using only the subset when performing Link-based analysis in order to return results faster and possibly with more accuracy.

3.1.5 SemRank

Anwanyu et al. [9] designed SemRank system which is an approach which ranks relationships instead of entities. The final ranking score calculated by this system is based on the predictability or gain of information of a semantic association, the degree

[1] http://stats.grok.se/

of similarity of a keyword and property occurring in a semantic association and the amount of differences between properties in the original schema and the properties that compose a path. Result ordering can be changed by the user based on his need. There was no information provided about improvements over other baseline algorithms. Moreover, relationship ranking was performed only for objects from the following domains: Flight, University, Banking and Organization. This approach may not be useful to our approach as we do not plan to rank relationships.

3.1.6 ReConRank

ReConRank proposed in Hogan et al. [10], is a PageRank based algorithm that performs dynamic ranking and only analyses the result data that matches the user query. An advantage of this approach is that since the ranking is not static, the returned results would be more relevant to the user queries. Disadvantages of this approach are that the extraction of topical graph is a challenge. Since dynamic ranking is performed query time is a challenge as well. The change made to the PageRank algorithm is that the ratio of all links received is considered as in-links in this algorithm. Because of this change, the number of iterations reduce to one-third. Weightings can be manipulated as well. It can be tuned according to the user. The authors say that the goodness of ReConRank relies on the relationships between resources and its provenance. The using of ratio of links received as in-links from this approach when performing Link-based analysis might be useful in our approach.

3.1.7 AKTiveRank

AKTiveRank formulated by Alani et al. [11] ranks ontologies relying on their importance to a given query. It uses the semantic web search engine Swoogle [7] to get the list of ontologies that need to be ranked. For our system currently we do not require ontology ranking and hence this approach is not useful.

3.1.8 YAGO-NAGA

In Kasneci et al. [12] the authors have described the NAGA semantic search engine. This system is based on Language Model and performs ranking based on the notions of Informativeness, Compactness and Confidence. Informativeness is defined as the amount of information that is represented by a certain result. As an example of ranking results based on informativeness take a query about Albert Einstein. Albert Einstein was more a scientist than a politician and hence more results about his career as a scientist should be returned instead of results about his career as a politician. In Compactness the graph structure is used to rank the results, that is, we prefer direct connections between entities. Also, the more the facts the lesser will be the probability likelihood and hence the lesser the compactness of the graph. Confidence expresses certainty about a particular fact. For computation the authors use provenance of information through a PageRank like algorithm. An extension of this system supporting keyword based search was developed by Elbassuoni et al. [13] and will be described in the later sections.

3.1.9 Harth et al.

The algorithm by Harth et al. [14] is another variation of PageRank algorithm for ranking structured data in which the authors assign weights to all links in the graphs based on authority or provenance of data source and then calculate PageRank for the whole graph even before query is entered. This approach is a completely contrasting approach to the ReConRank approach [10] described in the earlier sections.

3.1.10 RareRank

Wei [15] introduced the RareRank ranking algorithm which is another modified version of the PageRank algorithm. In this algorithm, the author replaces the random component in PageRank model by a more deterministic component based on the domain of search in order to reduce the randomness. The final ranking scores are a summation of the link information and content information. The content information is modelled on an ontology that allows the navigation from one document to another documents through previously non-existing links in addition to citation links. For out approach it might be useful to exploit the idea of replacing the random component by a more deterministic component based on the domain of search. For example, for the domain Politician, we could find out a more deterministic component for random surfing if we perform a link-based analysis.

3.1.11 DBPediaRanker

Mirizzi et al. [16] proposed DBPediaRanker, an algorithm that given a query tries to first explore all nodes belonging to the same domain. It dereferences nodes in the DBPedia graph for the exploration of domain. During exploration, the algorithm tries to calculate similarity between pair of nodes in the domain. To calculate similarity between two resources r_1 and r_2 the algorithm first checks the number of pages that contain the *rdfs:label* of both r_1 and r_2. The algorithm then checks whether a strong relation exists between r_1 and r_2 by exploiting the DBPedia property *dbpprop:wikilinks*. The result is a contextualized weighted graph with weights of links between two resources based on the similarity between the nodes. This approach could be exploited in our approach when we are trying to rank entities given other entities and time period.

3.1.12 DING

Delbru et al. [17] designed Dataset rankING or DING which is another adaptation of the PageRank algorithm. The algorithm calculates rank in three steps. First, it calculates the global dataset rank. Then it calculates entity rank. Finally, it calculates the global ranking score which is a combination of both global dataset rank and entity rank. For the global dataset ranking, it assigns high weights to certain kinds of links within dataset (links with high authority) and low ranking to more frequently occurring links within the dataset. These link weights are calculated using *lf-idf* approach. Entity ranks are calculated based on a version of PageRank adapted for entities. The computation of global rank is performed once dataset graph and entity

graph have been ranked. For our approach we could possibly make use of a similar approach for weight assignment of links when performing link-based analysis.

3.1.13 NOC-ORDER

Graves et al. [18] proposed NOC-ORDER to rank nodes in an RDF Graph. NOC-ORDER ranks nodes based on centrality measure. The approach can be understood by an example of considering a traveller with a limited amount of money wanting to explore maximum number of places in a city but having familiarity with the city. Naturally, the traveller would like to be in a central location in a city which connects to the maximum number of places with low cost to reach each of them. The algorithm tries to rank nodes in a graph based on this same idea. It checks the connectedness and distance of each node to the other nodes. The algorithm is an adaptation of *All-Pairs Shortest Path* algorithm for RDF Graph. the authors say that the direction of edge does not matter in case of an RDF Graph since for every predicate it is always possible to find an inverse predicate. This approach is an alternative way to compute rank of nodes in an RDF graph instead of using PageRank algorithm and could be looked at for our system.

3.2 Keyword search

This section provides a description of the ranking algorithms that permit keyword search. Most of these approaches add some extensions to the Semantic search approaches described in the previous section.

3.2.1 Language-model-based Ranking for Queries on RDF-Graphs

Elbassuoni et al. [13] extended the NAGA system [12] to allow for keyword based searches and approximate matches for results of SPARQL queries. The idea was that it may be desirable to allow users to narrow search by specifying certain keywords in queries and getting approximate matches when less or no exact matches exist. The authors performed query dependent ranking using Language Model(LM). The LM is used to construct query and result graph and then ranking is performed using Kullback-Liebler(KL) Divergence. A critical features in such ranking is the witness count. The authors considered witness count using keywords while finding keyword matches. Query relaxation was permitted to allow approximate matches with assignment if less weight to relaxed queries. To judge relevance of results Crowd-workers were used and the system was evaluated to similar systems using Normalized Discounted Cumulative Gain(NDCG) measure. Among similar systems, BANKS [19] described in the next sub-section, was shown to have lesser perfomance than this system. We could make use of witness count or the number of times a triple gets seen or extracted from a corpus with/without specified keywords from this approach in our approach. This system could also act as one of the baselines to ranking keyword queries when we extend our ranking from NYT archives in particular to Web archives in general.

3.2.2 BANKS

Bhalotia et al. [19] developed BANKS, a system that tries to find nodes matching terms of a keyword query. For establishing matches string comparison of keywords and data and metadata are considered. The relevance of an answer is determined based on a minimization of edge weights and node weights. This is done by reducing the problem to Steiner Tree problem which is considered to be NP Hard. The algorithm, therefore, is able to find optimization only for a subset or small dataset. It would not work for a big dataset that is not able to fit in memory. This approach has shown less performance than the approach described in previous subsection [13] and has the drawback of not being able to work for a large dataset and hence we should be careful if we consider to model our approach to keyword search in a similar fashion to this approach.

3.2.3 DISCOVER

Hristidis and Papakonstantinou [20] implemented DISCOVER, which is a system similar to the BANKS system. The difference between DISCOVER and BANKS is that they do not include backward links unlike BANKS where backward links were added to nodes to prevent the creation of hubs and they do not assign node weights and edge weights. The minimizing of distance between nodes is done by a greedy algorithm. The same authors proposed a further extension to this system [21] in which support of boolean AND and OR in keyword search was added. We could look at this system for their method of support of boolean AND and OR for keyword search.

3.2.4 ObjectRank

Balmin et al. [22] proposed ObjectRank which is an extension of PageRank for keyword search. The difference to PageRank is that provenance of the relationship between objects is taken into account and that the user can adjust the system according to the domain or his requirements. In the first stage, the algorithm computes a global ranking similar to PageRank. In the second stage, a keyword based ranking is computed. The final ranking is a combination of both the rankings. The authors believe that when performing keyword search, it is substantially more efficient to calculate global ranks first and then use these scores as initial values for keyword-specific computations. This approach could be compared against the approach by Elbassuoni et al. [13] to check which of these is better for keyword search.

3.2.5 BLINKS

He et al. [23] designed BLINKS, a system which follows the same strategy as BANKS [19] for performing keyword search over graph structures. The authors just improve the index structure for lower memory consumption and better query processing. There is no improvement in ranking. Since it has already been shown by Elbassuoni et al. that their approach [13] for keyword search performs better than BANKS approach, it might not be useful to look into this approach in developing our approach.

3.2.6 A Hybrid Approach for Searching in the Semantic Web

Rocha et al. [24] proposed a hybrid approach combining classical search engine techniques with spread activation techniques. The motivation behind this approach was that usually in systems supporting keyword searches only the results where the keyword actually occurs get returned to the user because of the use of string comparison techniques. Due to this, results which are conceptually related to the results but where the keyword does not occur are not returned. Spread activation techniques work as a concept explorer identifying concepts which are closely related to the initial set of activated concepts. The authors say that though better performance could be achieved if for every domain a domain expert sets the spread activation configuration still pretty good results are achieved even without setting domain-specific configuration. Because this approach returns close conceptually related results for keyword searches, this could be exploited in our system for better keyword search as well.

3.2.7 Post-Analysis of Keyword-based Search Results using Entity Mining, Linked Data and Link Analysis at Query Time

Fafalios and Tzitzikas [25] developed an approach that integrates classical Web with the Web of Linked Data. For the top-100 results BING search engine for keyword search, the system first detects the entities in the snippets of the results. Then for the top-k entities derived from a PageRank like algorithm, using the information available on the LOD tries to show the user how the top detected entities are related. The semantic graphs for the entities using which subsequent identification of the top relationships among entities can be performed at query time. Hence, this approach is useful as it provides users with semantic context and thus help users save time in exploratory search scenarios. In our approach to rank entities given a set of entities and a time period we could first check the top relations among the entities using this approach and then rank new entities based on their sharing of top relationships with given entities.

4 Conclusion

This article provides an overview of different systems for ranking on the web of data. It categorizes each system and describes the ranking approach followed by each system. It further highlights the useful parts of each approach which could possibly be exploited in our ranking approach on the semantic layer of the NYT archives and tries to check if any of the approaches could help in establishing the ground truth for our case.

References

[1] Jaspreet Singh, Wolfgang Nejdl and Avishek Anand. History by Diversity: Helping Historians search News Archives. In *Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval*. ACM, 2016.

[2] Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*. 30(1-7), 107-117, 1998.

[3] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th annual ACM-SIAM symposium on discrete algorithms*. 1998.

[4] Lorand Dali, Blaž Fortuna, Thanh Tran and Dunja Mladenić. Query-Independent Learning to Rank for RDF Entity Search. In *Extended Semantic Web Conference*. pp. 484-498, Springer Berlin Heidelberg, 2012.

[5] Sara Latifi and Mohammadali Nematbakhsh. Query-Independent Learning to Rank RDF Entity Results of SPARQL Queries. In *Proceedings of the 4th International Conference on Computer and Knowledge Engineering(ICCKE)*. IEEE, 2014.

[6] T. Finin, Y. Peng, R. Scott, C. Joel, S. A. Joshi, P. Reddivari. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM conference on information and knowledge management*. pp. 652-659, ACM Press, 2004.

[7] Li Ding, Rong Pan, Tim Finin, Anupam Joshi, Yun Peng and Pranam Kolari. Finding and ranking knowledge on the semantic web. In *International Semantic Web Conference*. Springer, 2005.

[8] Z. Nie, Y. Zhang, J.R. Wen and W.Y. Ma. Object-level ranking: Bringing order to web objects. In *World Wide Web Conference*. pp. 567-574, ACM, 2005.

[9] K. Anyanwu, A. Maduko, and A.P. Sheth. Semrank: Ranking complex relationship search results on the semantic web. In *World Wide Web Conference*. pp. 117127, ACM, 2005.

[10] Aidan Hogan, Andreas Harth and Stefan Decker. ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In *SSWS 2006(ICSW'06 Workshop)*. 2006.

[11] H. Alani, C. Brewster, and N. Shadbolt. Ranking ontologies with aktiverank. In *International semantic web conference, lecture notes in computer science*. Vol. 4273, pp. 115, Springer Berlin, 2006.

[12] G. Kasneci, F.M. Suchanek, G. Ifrim, M. Ramanath, G. Weikum. Naga: Searching and ranking knowledge. In *G. Alonso, J.A. Blakeley and A.L.P.Chen(Eds.), ICDE*. pp. 953-962, IEEE, 2008.

[13] S. Elbassuoni, M. Ramanath, R. Schenkel, M. Sydow and G. Weikum. Language-model-based ranking for queries on RDF-graphs. In *Proceedings of the 18th ACM conference on Information and knowledge management*. pp. 977-986, ACM, 2008.

[14] A. Harth, S. Kinsella and S. Decker. Using naming authority to rank data and ontologies for web search. In *A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, et al. (Eds.), International semantic web conference, lecture notes in computer science*. Vol. 5823, pp. 277292, Springer Berlin, 2009.

[15] W. Wei. Semantic search: Bringing semantic web technologies to information retrieval. Ph.D. thesis, University of Nottingham, 2009.

[16] R. Mirizzi, A. Ragone, T.D. Noia and E.D. Sciascio. Ranking the linked data: The case of Dbpedia. In *ICWE, Lecture Notes in Computer Science*. pp. 337-354, Springer Berlin, 2010.

[17] R. Delbru, N. Toupikov, M. Catasta, G. Tummarello and S. Decker. Hierarchical link analysis for ranking web data. In *Proceedings of the 7th international conference on the semantic web: Research and applicationsvolume part II, ESWC10*. pp. 225-239, Berlin, Springer Heidelberg, 2010.

[18] Alvaro Graves, Sibel Adali and James Hendler. A method to rank nodes in an RDF graph. In *2007 ISWC, Posters and Demonstrations*. 2008.

[19] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti and S. Sudarshan. Keyword searching and browsing in databases using banks. In *ICDE*. pp. 431-440, IEEE Computer Society, 2002.

[20] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *VLDB*. pp. 670-681, Morgan Kaufmann, 2002.

[21] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient ir-style keyword search over relational databases. In *VLDB*. pp. 850-861, 2003.

[22] A. Balmin, V. Hristidis and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *M.A. Nascimento, M.T. Özsu, D. Kossmann, R.J. Miller, J.A. Blakeley, K.B. Schiefer(Eds.), VLDB*. pp. 564-575, Morgan Kaufmann, 2004.

[23] H. He, H. Wang, J. Yang, and P.S. Yu. Blinks: Ranked keyword searches on graphs. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. pp. 305-316, New York, ACM Press, 2007.

[24] Christiano Rocha, Daniel Schwabe and Marcus Poggi Aragao. A hybrid approach for searching in the semantic web. In *13th international conference on World Wide Web*. ACM, 2004.

[25] Pavlos Fafalios and Yannis Tzitzikas. Post-Analysis of Keyword-based Search Results using Entity Mining, Linked Data and Link Analysis at Query Time. In *IEEE International Conference on Semantic Computing*. IEEE, 2014.