Efficient Computation of Relationship-Centrality in Large Entity-Relationship Graphs

Stephan Seufert[1], Srikanta J. Bedathur[2], Johannes Hoffart[1], Andrey Gubichev[3], and Klaus Berberich[1]

[1] Max Planck Institute for Informatics, Germany
{sseufert,jhoffart,kberberi}@mpi-inf.mpg.de
[2] IIIT Delhi, India
bedathur@iiitd.ac.in
[3] Technische Universität München, Germany
andrey.gubichev@in.tum.de

Abstract. Given two sets of entities – potentially the results of two queries on a knowledge-graph like YAGO or DBpedia– characterizing the relationship between these sets in the form of important people, events and organizations is an analytics task useful in many domains. In this paper, we present an intuitive and efficiently computable vertex centrality measure that captures the importance of a node with respect to the explanation of the relationship between the pair of query sets. Using a weighted link graph of entities contained in the English Wikipedia, we demonstrate the usefulness of the proposed measure.

1 Introduction

Consider a journalist researching the political relations between *France* and *Germany*. In order to gain insight into the underlying relationship, it is an important task to identify entities (e. g. events, organizations, etc.) that play an important role in the interactions between these countries. This task can be greatly simplified if the journalist could simply input two sets of entities – corresponding to the classes "French Politicians" and "German Politicians" – and the system automatically generates a ranking of entities in the knowledge base, reflecting their potential for characterizing the relationship between the two entity-sets. Variants of this *relationship characterization* problem can be found in settings ranging from political studies to analysis of relationships in computational biology and economics. With the availability of massive entity-relationship networks such as Wikipedia, DBLP, and BioCyc networks as well as large Semantic Web ontologies like YAGO2 [5], solutions not only need to be effective, but also scalable. State-of-the-art approaches for identifying important nodes in networks include various centrality measures (e.g., closeness- and betweenness-centrality) which operate on the entire network, without any specific input entity sets.

In this paper, we develop a novel centrality measure called *relationship centrality*, that assesses the 'strength' of a node in the relationship path between the given two sets of nodes. These scores can be computed exactly, or can be well-approximated to scale to networks as large as the entire Wikipedia graph, comprising tens of millions of edges. The resulting rankings can further be restricted to entities of certain types (e.g, `Organization` or `Location` etc.), leveraging semantic knowledge-bases such as YAGO2 [5] or DBPedia [1]. In the following section, we formally introduce our novel centrality measure.

2 Relationship Centrality

Graph centrality measures, which assign to every node a score reflecting its importance in the graph structure, are a valuable tool for analyzing different kinds of graphs. Although they have been studied extensively in the scope of social networks, the use of centrality measures in the context of Semantic Web is gaining importance only recently. The classical measures proposed in past include *closeness* [3] and *betweenness centrality* [2]. However, these measures become computationally expensive when we consider large networks. Also, their utility in ranking nodes with respect to an input set of entities is rather limited.

In contrast, the measure we introduce in this work, called *relationship centrality*, is easier to compute since it is designed to assign scores that reflect the centrality only with respect to the two input entity sets, rather than on a global scale. Formally, given two query entity sets S and T from the network, we define the relationship centrality of a node v as follows:

$$c_R(v) = \sum_{s \in S} \sum_{t \in T} \frac{1}{\rho(s, v, t)},$$

where $\rho(s, v, t)$ is a penalty function for a path connecting a node $s \in S$ and $t \in T$ passing through v, given by: $\rho(s, v, t) = (1 + d(s, v)) \cdot (1 + d(v, t))$. The distance $d(\cdot, \cdot)$ between two nodes connected by an edge can be customized based on the underlying network to measure the semantic distance between the corresponding entities. The corresponding edge-weighting schemes we envision can be based on the graph structure (Milne-Witten inlink overlap measure [6]) or textual representations of the entities (keyphrase overlap measure [4]), among others.

Relationship centrality takes into account different paths than betweenness centrality (which regards the shortest paths between all pairs of vertices). For every vertex in the graph and every pair $(s, t) \in S \times T$, the shortest path from s to t passing through v contributes to the centrality score of v.

The corresponding paths are computed as follows: For every vertex $s \in S$ and $t \in T$, the shortest distances to each vertex $v \in V$ are computed using Dijkstra's algorithm. Then, the centrality scores for every vertex can be computed from the resulting distance vectors. While the $O(m + n \log(n))$ time complexity induced by each of the $|S| + |T|$ required shortest path computations is rather lightweight, for very large graphs the corresponding computation time can be too demanding, especially for interactive applications. For this purpose, we have experimented with an alternative scoring scheme which only considers shortest path distances up to a value $\Delta \in \mathbb{R}$. Then, the distance from a query node q to a vertex v is approximated in the following way:

$$\tilde{d}(q, v) = \begin{cases} d(q, v) & \text{for } d(q, v) \leq \Delta, \\ \text{diam}(G) & \text{else,} \end{cases}$$

where $\text{diam}(G)$ denotes the diameter of the (weighted) graph. In our experimental evaluation in Section 4, we evaluate the quality of computed scores based on different choices of the *cutoff parameter*, Δ.

Rank	Entity
1	2009 G-20 Pittsburgh sum.
2	2010 G-20 Toronto summit
3	37th G8 summit
4	Iraq War
5	35th G8 summit
6	2009 G-20 London Summit
7	36th G8 summit
8	2010 G-20 Seoul summit
9	Presidency of G. W. Bush
10	2009 Nobel Peace Prize

(a) Q1

Rank	Entity
1	The Expendables (2010 film)
2	Crouching Tiger, Hidden Dragon
3	The Forbidden Kingdom
4	Rush Hour 2
5	Police Story (1985 film)
6	Once Upon a Time in China II
7	Fist of Fury
8	Romeo Must Die
9	Kung Fu Hustle
10	Fearless (2006 film)

(b) Q2

Rank	Entity
1	Iraq War
2	War in Afghanistan
3	Gulf War
4	Op. Enduring Freedom
5	Yom Kippur War
6	War on Terror
7	Battle of Karameh
8	Palestinian diaspora
9	Operation Opera
10	Suez Crisis

(c) Q3

Table 1. Top ranked entities for example queries

3 Application

In this section, we briefly present the application scenarios we envision. We target analytical tasks at the downstream of Semantic Web applications. In particular, we consider a large knowledge-base (such as YAGO or DBPedia), over which S and T sets are derived as results of SPARQL queries.

Example: As a concrete scenario, the following two queries retrieve all organizations conducting research on (variants of) lung cancer, and all tobacco companies respectively:

```
SELECT ?p WHERE { ?p <rdf:type> <Organization> . ?p <worksOn> <Lung_Cancer> }
SELECT ?c WHERE { ?c <rdf:type> <American_Tobacco_Company> }
```

An analytics task could be to identify legal cases that played an important role in the relationship between the entity sets corresponding to the query results. We can utilize the relationship centrality measure developed above, and then use a type hierarchy, e. g. Wikipedia categories or the WordNet lexical database, to retain only the relevant entity types in the generated ranking.

4 Experimental Evaluation

In this section, we provide an overview over our experimental evaluation of the relationship centrality measure. In order to empirically validate the assigned scores, we have compiled several example queries over an edge-weighted entity-relationship graph obtained from Wikipedia: The vertices of the graph correspond to Wikipedia pages that represent an entity contained in the YAGO knowledge base. Two vertices u, v are connected via a weighted, undirected edge if there exists an internal Wikipedia link in either direction between the corresponding articles, $A(u), A(v)$. The weight we assign to the edge (u, v) should capture the semantic relatedness of the respective concepts. For this purpose, we employ the inlink overlap measure originally proposed by Milne and Witten [6]. For nodes u, v the weight (inverse semantic relatedness) is given by

$$d(u,v) = \frac{\log(\max\{|I_u|, |I_v|\}) - \log(|I_u \cap I_v|)}{\log(n) - \log(\min\{|I_u|, |I_v|\})},$$

where I_u and I_v denote the set of pages linking to $A(u)$ and $A(v)$, respectively and n corresponds to the overall number of pages. The resulting weights lie in the interval $[0, \infty]$. Using this measure, vertices u and v exhibit a high semantic relatedness if the weight of the edge (u, v) is close to zero. Finally, we discard all edges with $d(u, v) \geq 1$.

Query	$\Delta = \infty$	$\Delta = 1$		$\Delta = 0.5$	
	Time	Time	τ	Time	τ
Q1	41,960.40 ms	18,629.80 ms	1.0	4,616.05 ms	0.55
Q2	48,174.80 ms	15,002.70 ms	1.0	5,117.02 ms	0.60
Q3	71,162.50 ms	32,028.50 ms	1.0	7,858.39 ms	0.87

Table 2. Computation time and ranking quality

The resulting graph structure contains around 2.5 million vertices (entities) and roughly 37 million edges.

In order to empirically evaluate our ranking, we use three example queries where the sets of entities correspond to a collection of

Q1: *Events* between European politicians (S) and US American politicians (T)[4]
Q2: *Movies* between US action movie stars (S) and Asian action movie stars (T)[5]
Q3: *Events* between countries from the Middle East/Central Asia (S) and Western countries (T)[6]

In Table 1 we present the top-10 ranked results by relationship centrality for each of the queries. The resulting rankings suggest that our measure is useful for the explanation of the relationship between the sets of query entities. Regarding the computation time, we give an overview over the effect of pruning the shortest path computation using different cutoff parameters Δ, as well as the resulting rank correlation (measured by Kendall's τ) for the top 10 entities in Table 2.

5 Conclusions & Outlook

In this work we have presented the *relationship centrality* measure, a vertex centrality score that reflects the potential of an individual vertex for the explanation of the relationship between two sets of query nodes. Our preliminary experimental results over the edge-weighted Wikipedia entity-relationship graph indicate that our measure can provide valuable insights into the relationship between sets of real-world entities. In future work, we plan to conduct a large-scale evaluation of our result ranking in a user study. In addition, we plan to use our centrality measure as a building block for extracting interesting subgraphs between the query entities.

References

1. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *J. Web Sem.*, 7(3):154–165, 2009.
2. L. C. Freeman. A set of measures of centrality based upon betweenness. *Sociometry*, 40:35–41, 1977.
3. L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3):215–239, 1979.
4. J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. KORE: Keyphrase Overlap Relatedness for Entity Disambiguation. In *CIKM'12: Proceedings of the 21th ACM International Conference on Information and Knowledge Management*, pages 545–555. ACM, 2012.
5. J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 2013.
6. D. Milne and I. H. Witten. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *WIKIAI'08: Proceedings of the 2008 AAAI Workshop on Wikipedia and Artificial Intelligence*. AAAI, 2008.

[4] S = {Angela Merkel, Nicolas Sarkozy, David Cameron, Silvio Berlusconi},T = {Barack Obama, Hillary Clinton}
[5] S = {Chuck Norris, A. Schwarzenegger, Sylvester Stallone,Bruce Willis},T = {Jet Li, Jackie Chan, Chow Yun-Fat}
[6] S = {Iraq, Iran, Israel, Palestine, Afghanistan, Pakistan},T = {Germany, France, Spain, Italy, Netherlands, Portugal}