Separating Wheat From the Chaff – A Relationship Ranking Algorithm

Sumit Bhatia, Alok Goel, Elizabeth Bowen, and Anshu Jain

IBM Watson, Almaden Research Centre, San Jose, CA, USA
{sumit.bhatia,alok.goel,elizabeth.bowen,anshu.n.jain}@us.ibm.com

Abstract. We address the problem of ranking relationships in an automatically constructed knowledge graph. We propose a probabilistic ranking mechanism that utilizes entity popularity, entity affinity, and support from text corpora for the relationships. Results obtained from preliminary experiments on a standard dataset are encouraging and show that our proposed ranking mechanism can find more informative and useful relationships compared to a frequency based approach.

1 Introduction

We are transitioning from the era of Big Data to Big Knowledge, and semantic knowledge bases such as knowledge graphs play an important role in this transition. A knowledge graph consists of real world entities (or concepts) as nodes. A given node may be connected with many other nodes and there could be multiple edges between two given nodes, with each edge representing a real world fact. Thus, a knowledge graph is an essential source for getting important facts about real world entities and how these entities are related to each other. Knowledge Graphs can be constructed either manually (facts authored by humans) or automatically (facts extracted from text using Machine Learning tools). Manually curated knowledge graphs such as DBPedia have little or no noisy facts as they are carefully authored, but they require very large human efforts. This problem is further exacerbated in enterprise domains and custom domains such as life sciences, finance, intelligence, etc. where domain expertise is also crucial to add good quality facts in the graph. As a result, efforts have been made for development of systems for automatic construction of semantic knowledge bases for domain specific corpora [1] and systems that use such domain specific knowledge bases [4] are gaining prominence. In such systems, a machine learning based annotator is used to extract entities and relationships from domain specific corpus. As a result, in such knowledge bases, for each discovered relationship, *associated text mentions* from the corpus are also available and number of times a relationship is observed in the corpus can be used as a proxy for that relationship's *evidence or strength*. Such information may or may not be available with manually curated knowledge graphs.

Typically, a given entity may be involved in many relationships and we argue that all such relationships are not equally important and informative and thus, there is a need for methods that can identify the most relevant and meaningful relationships for a given entity. There have been some efforts to rank the entities and relationships in knowledge bases. Li et al. [3] propose an entity-relationship structured query mechanism for

ranking Wikipedia entities and their relationships. However, their method is dependent on the hyperlinks structure derived from Wikipedia and thus, can not be applied for graphs constructed from generic corpora. Instead of finding most important relationships, Zhang et al. [5] propose an alternative way to cluster similar relationships together and then allowing the users to further explore clusters of interest. In this paper, we address the problem of *ranking relationships for an entity in an automatically constructed knowledge graph*. We propose a probabilistic framework that judges the relevance of a relationship by utilizing various measures such as entity popularity, strength of evidence for a relationship, and affinity between input and target entities. We use a semantic graph constructed from text of all articles in Wikipedia by automatically extracting the entities and their relations by using IBM's Statistical Information and Relation Extraction (SIRE) toolkit[1]. Even though there exist popular knowledge bases like DBPedia that contain high quality data, we chose to construct a semantic graph using automated means as such a graph will be closer to many practical real world scenarios where high quality curated graphs are often not available and one has to resort to automatic methods of constructing knowledge bases. Our graph contains more than 30 millions entities and 192 million distinct relationships in comparison to 4.5 million entities and 70 million relationships in DBpedia.

2 Proposed Relationship Ranking Algorithm

Let us consider a Knowledge Graph $\mathcal{G} = \{E, R\}$, where, $E = \{e_1, e_2, \ldots, e_n\}$ is the set of nodes (or entities) and $R = \{r_1, r_2, \ldots, r_m\}$ is the set of edges (or relationships). Further, let $w : R \rightarrow \mathbb{R}_+$ is a weight function that gives weight of any edge in the graph. This function can be defined in various ways to measure the importance of an edge. We chose the frequency of occurrence of the given fact in the text corpus (mention count) as the weight of the edge corresponding to that relation in the graph. Given an input entity e, and a set $R_e \in R$ of all the relations involving e, we want to produce an ordered list of all the elements of R_e, ordered by their importance/relevance. The task of selecting a relationship involving input entity e can be decomposed in two steps – first selecting a target entity e_t, and then selecting an edge that connects these two entities. For example, for input *Barack Obama*, we first select a target entity, say *United States*, and then decide which of the two relationships out of *citizenOf* and *presidentOf* should be picked. Mathematically,

$$P(r, e_t|e) = P(e_t|e)P(r|e_t, e) = P(r|e, e_t)\frac{P(e_t)P(e|e_t)}{P(e)} \tag{1}$$

In the above equation, $P(e)$ can be ignored for ranking purposes since this factor will remain same for all target entities and relationships. The above equation can then be written as follows:

$$P(r, e_t|e) \propto \underbrace{P(e_t)}_{\text{Entity Prior}} \times \underbrace{P(e|e_t)}_{\text{Entity Affinity}} \times \underbrace{P(r|e, e_t)}_{\text{Relationship Strength}} \tag{2}$$

The above equation represents the ranking function that can be used to rank all relationships of a given entity. We now discuss the three components in the above equation contributing to the overall relevance score of a given relationship.

[1] http://ibmlaser.mybluemix.net/siredemo.html

Entity Prior: This component captures the intuition that in absence of any other information, the input entity has a higher chance of having a relationship with a popular entity in the graph as compared to a rare entity in the graph. It can be computed as follows:

$$P(e_t) \propto relCount(e_t) \tag{3}$$

where, $relCount(e_t)$ is the number of relationships entity e_t is involved in.

Entity Affinity: A target entity that has most of its relationships (or most of its strongest relationships) with the input entity is more important as compared to a target entity that has very few (or very weak relationships) with the input entity. For example, in our knowledge graph, both "Florida" and "France" have almost equal number of relationships in the graph, however, "Florida" has a larger fraction of its relationships with "USA" and some of its strongest relationships are with "USA". Hence, compared to "France", "Florida" is a more specific entity to "USA". Mathematically, it can be expressed as follows:

$$P(e|e_t) = \frac{\sum_{r_i \in R(e,e_t)} w(r_i) \times r_i}{\sum_{r_i \in R(e_t)} w(r_i) \times r_i} \tag{4}$$

where, $R(e_t)$ is the set of all relationships e_t is involved in and $R(e, e_t)$ is the set of all relationships between e and e_t.

Relationship Strength: While the previous two components were concerned with capturing the relevance of target entities for a given input entity, this component tries to measure the relative importance of different relationship types once we have selected the target entity. Given two entities, there could be multiple relationships between them. For example, "Barack Obama" is connected to "USA" with multiple relationships such as *presidentOf*, *citizenOf*, *livesAt*, *bornAt*, etc. In absence of any other information, we hypothesize that a relationship having more support/evidence from the corpus is more important than a relationship that has little supporting evidence. Mathematically,

$$P(r|e, e_t) = \frac{mentionCount(r, e, e_t)}{\sum_{r \in R_{e,e_t}} mentionCount(r, e, e_t)} \tag{5}$$

where, $mentionCount(r, e, e_t)$ represents the number of times relationship r connecting e and e_t was mentioned in the text corpus, and R_{e,e_t} is the set of all relationships between entities e and e_t.

3 Evaluation

For evaluating our proposed relationship ranking approach, we use the set of entities provided in the KORE entity relatedness dataset [2]. This dataset provides 21 seed entities from various domains. However, the dataset does not provide a ranked list of important relationships for the seed entities. Hence, we took help of two human evaluators to assess the quality of results produced by proposed ranking approach. For each input entity, we generated top 10 relationships ranked by our proposed ranking function and also top 10 most popular relationships as a baseline. Each evaluator was asked to rate the resulting relationships using a three point scale – 0 for an incorrect/noisy relationship, 1 for a correct but not useful relationship, and 2 for a correct and highly interesting relationship. As an example,"Brad Pitt" is *spouseOf* "Angelina Jolie" is a

much more useful and informative relationship when compared to a generic relationship "Brad Pitt" is a *partOfMany* "Actors", even though both relationships are correct. One evaluator provided judgments for relationships for 11 entities and one provided for 10 entities. The results are tabulated in Table 1. We observe that while almost all the relationships produced by both the approaches were correct (corresponding to scores of 1 and 2), the proposed approach was much better at finding more informative relationships (70.48% relationships with score 2 compared to only 47.14% for popularity based ranking). The results of this preliminary evaluation are encouraging and provide strength to our hypothesis that not all facts about an entity are equally important and hence, the need for an appropriate relationship ranking algorithm.

Method	Score 0	Score 1	Score 2
Popularity	8 (3.8%)	103 (49.05%)	99 (47.14%)
Proposed Approach	10 (4.76%)	52 (24.76%)	148 (70.48%)

Table 1. Results for relationship ranking as provided by human evaluators.

4 Conclusions and Future Work

We discussed the problem of ranking facts about a given entity in a knowledge graph and proposed a probabilistic framework to rank relationships/facts. Results of a preliminary evaluation study are encouraging and our future work will focus on enhancing the evaluation, both qualitatively and quantitatively. One major limitation of proposed approach is its inability to find facts that are customized to a user's requirements. For example, for the input entity *Barack Obama*, a user researching about *presidential elections* will be interested in different facts than a user interested in his *education history*. Therefore, our future research work will focus on *context sensitive* ranking of relationships so that users can get facts that are most important to their information needs.

References

1. V. Castelli, H. Raghavan, R. Florian, D.-J. Han, X. Luo, and S. Roukos. Distilling and exploring nuggets from a corpus. In *SIGIR*, pages 1006–1006, 2012.
2. J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *CIKM '12*, pages 545–554, 2012.
3. X. Li, C. Li, and C. Yu. Entity-relationship queries over wikipedia. *ACM Trans. Intell. Syst. Technol.*, 3(4):70:1–70:20, Sept. 2012.
4. M. Nagarajan et al. Predicting future scientific discoveries based on a networked analysis of the past literature. In *KDD*, pages 2019–2028, 2015.
5. Y. Zhang, G. Cheng, and Y. Qu. *JIST 2013, Selected Papers*, chapter Towards Exploratory Relationship Search: A Clustering-Based Approach, pages 277–293. 2014.