A PageRank like Approach for Ranking Archived Documents for Structured Queries

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

July 14, 2017

1 Introduction

1.1 Notions and Notations

Web Entities, Concepts and Events. In our problem, we define a *web entity* e as anything with a distinct, separate and meaningful existence that also has a "web identity" which is expressed through a unique URI u_e(e.g., a Wikipedia/DBpedia URI). This not only includes persons, locations, organizations, etc., but also concepts (e.g., *liberty*) and events (e.g., *Iranian elections of 2012*).[1]

Let E be a finite set of web entities, e.g., all Wikipedia entities, concepts and events expressed through a set of URIs U. Each entity $e \in E$ is associated with a unique URI, while several labels/names can be used to refer to this entity. For eg., for the entity *Donald Trump* (https://en.wikipedia.org/wiki/Donald_Trump), possible names are "Donald Trump", "Trump", "President Trump", etc.

Documents and extracted entities. Let D be a set of documents, e.g., a set of news articles. For a document $d \in D$, let $publ(d)$ denote its publication date. Let also $ents(d) \subseteq E$ be all entities mentioned in d. Inversely, for an entity $e \in E$, let $docs(e) \subseteq D$ be all documents that mention e, i.e., $docs(e) = \{d \in D \mid e \in ents(d)\}$. For an entity $e \in E$ and a document $d \in D$, let $count(e, d)$ be the number of occurrences of e in d. Finally, let E_D be all entities mentioned in documents of D, i.e., $E_D = \cup_{d \in D} ents(d)$.

Time periods. Let Δ be a fixed time period (e.g., *day*, *week* or *month*). Based on a time period Δ, let $T = (t_0, t_1, \ldots, t_n)$ be an ordered list of consecutive time points covering the publication dates of all documents in D. Note that

[1] From now on, when we say *entity* we refer to *entity*, *concept*, or *event*.

$t_i - t_{i-1} = \Delta$, for each $i \geq 1$. Now, let $\delta_i = [t_i, t_{i+1})$ be the time period of size Δ between two consecutive time points. We consider that a document d is published in the period δ_i, if $t_i \leq publ(d) < t_{i+1}$. For a document $d \in D$, let $timep(d)$ be the time period in which d was published. Now, let P_D denote the set of distinct time periods of all documents in D, i.e., $P_D = \cup_{d \in D}\{timep(d)\}$. For a time period $p \in P_D$, let $docs(p) \subseteq D$ be the set of all documents published within p, i.e., $docs(p) = \{d \in D \mid timep(d) = p\}$, and $ents(p) \subseteq E_D$ be the set of entities discussed in documents of D published within p, i.e., $ents(p) = \cup_{d \in docs(p)} ents(d)$.

2 Problem Definition

Given a corpus of documents D, a set of entities E_D mentioned in documents of D, and a SPARQL query Q requesting documents from D published within a set of *time periods* $P_Q \subseteq P_D$ and related to one or more *Entities of Interest (EoI)* $E_Q \subseteq E_D$ with logical `AND` (mentioning all EoI) or `OR` (mentioning at least one EoI) semantics, and an RDF Graph G, the problem is how to rank the documents $D_Q \subseteq D$ that (equally) match the query Q using a *PageRank-like algorithm.*

Figure 1 shows an example SPARQL query requesting documents published in 1990 discussing about the entities *Nelson Piquet* and *Ayrton Senna* (`AND` semantics), while the query in Figure 2 requests articles of 1990 discussing about *Ferrari Formula One drivers* (`OR` semantics).

```
SELECT DISTINCT ?article WHERE {
  ?article dc:date ?date FILTER(?date >= "1990-01-01"^^xsd:date &&
                                ?date <= "1990-12-31"^^xsd:date) .
  ?article oae:mentions ?entity1, ?entity2 .
  ?entity1 oae:hasMatchedURI  <http://dbpedia.org/resource/Nelson_Piquet> .
  ?entity2 oae:hasMatchedURI  <http://dbpedia.org/resource/Ayrton_Senna> }
```

Figure 1: SPARQL query for retrieving articles of 1990 discussing about *Nelson Piquet* and *Ayrton Senna* (`AND` semantics).

```
SELECT DISTINCT ?article WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?p dc:subject <http://dbpedia.org/resource/Category:Ferrari_Formula_One_drivers> }
  ?article dc:date ?date FILTER(?date >= "1990-01-01"^^xsd:date &&
                                ?date <= "1990-12-31"^^xsd:date)
  ?article oae:mentions ?entity .
  ?entity oae:hasMatchedURI  ?p }
```

Figure 2: SPARQL query for retrieving articles of 1990 discussing about *Ferrari Formula One drivers* (`OR` semantics).

3 Probabilistic Analysis

We *dynamically* construct a graph of documents and identified entities and then analyze it probabilistically for identifying the important document and entity nodes. For the graph analysis and its node scoring, we follow a Random

Walk-based (PageRank-like) [?]ethod because the theoretical framework is quite solid and also according to the need and application we can bias or customize the method. The probabilistic analysis is mentioned in detail after we define some aspects and present an user side exploratory search scenario which better motivates the PageRank-like approach that we propose.

3.1 Aspects

We define the following aspects: i) the *relativeness* of a document with respect to the EoI, ii) the *timeliness* of a document with respect to its publication date, and iii) the *relatedness* of a document with respect to its reference to other entities related to the EoI. iii) the *time difference importance* between the publication dates of two documents.

Relativeness

We consider that if the EoI are mentioned multiple times within a document, the document should receive a high score (since the document's topic may be about these entities). The term frequency (in our case entity frequency) is a classic numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

For the case of AND semantics ("$\wedge$"), the *relativeness* score of a document $d \in D_Q$ based only on the *frequency* of the EoI is defined as:

$$ScoreD^f_\wedge(d) = \frac{\sum_{e \in E_Q} count(e,d)}{\sum_{e' \in ents(d)} count(e',d)} \tag{1}$$

Notice that the score of a document will be 1 if it contains the EoI and no other entity.

For the case of OR semantics ("$\vee$"), we can also consider the number of EoI mentioned in the document (since a document does not probably contain all the EoI as in the case of AND semantics). In that case, the *relativeness* score can be defined as follows:

$$ScoreD^f_\vee(d) = \frac{\sum_{e \in E_Q} count(e,d)}{\sum_{e' \in ents(d)} count(e',d)} \cdot \frac{|ents(d) \cap E_Q|}{|E_Q|} \tag{2}$$

where $\frac{|ents(d) \cap E_Q|}{|E_Q|}$ is the percentage of EoI discussed in the document. The score of a document will be 1 if it contains all the EoI and no other entity. This formula favors documents mentioning multiple times many of the EoI.

Timeliness

We consider that a time period $p \in P_Q$ is important for the entities in E_Q, if there is a relatively big number of documents in D_Q discussing about these entities during p. For example, a big number of articles about *Nelson Mandela* was published the period 11-13 of February 1990 because in February 11 *Nelson*

Mandela was released from prison. Thus, articles published during that period should be promoted since they are probably related to this important event of *Nelson Mandela*'s life.

For the case of `AND` semantics, we define the following importance score of a *time period* $p \in P_Q$:

$$ScoreP_{\wedge}(p) = \frac{|docs(p) \cap D_Q|}{|D_Q|} \tag{3}$$

For the case of `OR` semantics, in a time period p there may be a big number of documents discussing only for one of the EoI, while in a time period p' there may be a smaller number of documents discussing though for many of the EoI. For also taking into account the number of EoI discussed in documents of a specific time period, we consider the following formula:

$$ScoreP_{\vee}(p) = \frac{|docs(p) \cap D_Q|}{|D_Q|} \cdot P_{EoI}(p) \tag{4}$$

where, $P_{EoI}(p)$ is the average percentage of EoI discussed in articles of $p \in P_Q$, i.e.:

$$P_{EoI}(p) = \frac{\sum_{d \in docs(p) \cap D_Q} \frac{|ents(d) \cap E_Q|}{|E_Q|}}{|docs(p) \cap D_Q|} \tag{5}$$

By considering only timeliness, the score of a document can be determined by the score of its publication date, i.e.:

$$ScoreD^t(d) = ScoreP(timep(d)) \tag{6}$$

Relatedness

Entities that are co-mentioned frequently with the EoI in important time periods are probably important for the EoI. For example, *Apartheid* was an important concept related to *Nelson Mandela* during 1990, thus articles discussing for both *Apartheid* and *Nelson Mandela* should be promoted. However, there may be also some general entities (e.g., *South Africa* in our example) that co-occur with the EoI in almost all documents (independently of the time period). Thus, we should also avoid over-emphasizing documents mentioning such "common" entities.

For the case of `AND` semantics, we consider the following *relatedness* score for an entity $e \in E_D \setminus E_Q$:

$$\begin{aligned} ScoreE_{\wedge}(e) &= idf_{\wedge}(e) \cdot \sum_{p \in P_Q} (ScoreP_{\wedge}(p) \cdot \frac{|docs(p) \cap D_Q \cap docs(e)|}{|docs(p) \cap D_Q|}) \\ &= idf_{\wedge}(e) \cdot \sum_{p \in P_Q} \frac{|docs(p) \cap D_Q \cap docs(e)|}{|D_Q|} \end{aligned} \tag{7}$$

where $idf_{\wedge}(e)$ is the inverse document frequency of entity e in the set of documents discussing about the EoI in the entire corpus, which can be defined as follows:

$$idf_{\wedge}(e) = 1 - \frac{|docs(e) \cap (\cap_{e' \in E_Q} docs(e'))|}{|\cap_{e' \in E_Q} docs(e')|} \tag{8}$$

The formula considers the percentage of documents in which the entity co-occurs with all of the EoI in each time period, as well as the importance of the time period.

For the case of `OR` semantics, we define the inverse document frequency $idf_{\vee}(e)$ to include documents mentioning one(or more) EoI as follows:

$$idf_{\vee}(e) = 1 - \frac{|docs(e) \cap (\cup_{e' \in E_Q} docs(e'))|}{|\cup_{e' \in E_Q} docs(e')|} \tag{9}$$

For the case of `OR` semantics, the above formula does not consider the number of different EoI discussed in documents together with the entity e. To also handle this aspect, we consider the following *relatedness* score for the case of `OR` semantics:

$$\begin{aligned} ScoreE_{\vee}(e) &= idf_{\vee}(e) \cdot \sum_{p \in P_Q} (ScoreP_{\vee}(p) \cdot \frac{|docs(p) \cap D_Q \cap docs(e)|}{|docs(p) \cap D_Q|}) \cdot P_{EoI}(e) \\ &= idf_{\vee}(e) \cdot \sum_{p \in P_Q} (P_{EoI}(p) \cdot \frac{|docs(p) \cap D_Q \cap docs(e)|}{|D_Q|}) \cdot P_{EoI}(e) \end{aligned} \tag{10}$$

where $P_{EoI}(e)$ is the average percentage of EoI discussed in articles together with $e \in E_D \setminus E_Q$, i.e.:

$$P_{EoI}(e) = \frac{\sum_{d \in docs(e) \cap D_Q} \frac{|ents(d) \cap E_Q|}{|E_Q|}}{|docs(e) \cap D_Q|} \tag{11}$$

This formula favors related entities that i) co-occur frequently with many of the EoI, ii) are discussed in documents published in important (for the EoI) time periods.

Time Difference Importance

The time difference importance score of a document d' relative to a document d can be defined as follows for the case of both `AND` and `OR` Semantics:

$$ScoreD^m(d) = 1 - \frac{(|publ(d) - publ(d')| + 1)}{\max_{d'' \in D_Q, d'' \neq d}(|publ(d) - publ(d'')| + 1)} \tag{12}$$

where $|publ(d) - publ(d')|$ is defined as the absolute difference (in *days*) of the publication dates of the documents d and d'.

3.2 Modeling a Random Walker

The exploratory search process is modelled as a *Random Walker* of the graph defined by the documents, the entities of interest EoI, the mined entities and their connections.

Specifically, whenever the walker is at an entity of interest e:
(i) With probability p_1 he moves to a document $d \in D_Q$.
(ii) With probability 1-p_1 he moves to an entity e' related to the entity of interest e.

When the walker is at a document $d \in D_Q$:
(i) With probability p_2 he moves to another document $d' \in D_Q$.
(ii) With probability 1-p_2 he moves back to the entity of interest e.

Whenever at a related entity e':
(i) With probability p_3 he moves back to an entity of interest e.
(ii) With probability p_4 he moves to a document $d' \in docs(e')$.
(iii) With probability 1-p_3-p_4 he moves to another entity e'' related to e'.

Lastly, when the walker is at a document d':
(i) With probability p_5 he moves back to an entity of interest e.
(ii) With probability 1-p_5 he moves back to a related entity $e' \in ents(d')$.

The Markov chain of the corresponding process is depicted in Figure 3. This process models the user behaviour in our search environment: the user submits a SPARQL query containing entities EoI that he is interested in some time period of interest and the system a list of documents mentioning the entities of interest EoI. The user can then open results(documents) that contain the EoI. Otherwise, the user can either click on some the related entities. From the related entities the user can now : (a) browse documents that contain the related entity, or (b) click on an entity related to the related entity, or (c) display back the initial results mentioning the EoI since the original objective of the user was to find interesting documents mentioning the EoI in some period of interest.

3.3 Creating the Graph of States and Transitions

First, we provide a definition of a semantic graph of documents and entities, and then we describe in detail the construction of the graph of states and transitions corresponding to the modelling.

The semantic graph of documents and entities. Both the documents and entities are considered as vertices in $\mathcal{X}$, and for drawing edges we take into account documents in which an entity has been detected. Specifically, an edge is drawn starting from an entity e and ending at a document d, if $e \in ents(d)$ (i.e., e was extracted from d). By exploiting a Semantic Knowledge Base (in our case *DBpedia*), interesting related entities linked as triples with the entity can be fetched (performed during *entity enrichment* step. In the RDF Graph G_i, let $T(u_e) \subseteq G_i$ be triples that describe information about u_e. Now, for two

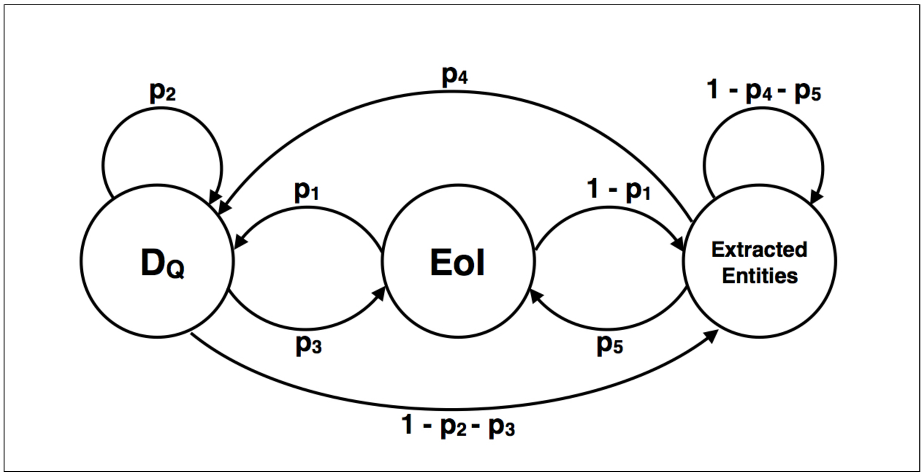

Figure 3: The Markov chain of the process.

entities $e_1, e_2 \in E$, if $(u_{e_1}, p, u_{e_2}) \in T(u_e 1)$, then we draw an edge starting from the entity e_1 and ending at the entity e_2 and we denote this edge by $edge(e_1, e_2)$.

The State Transition Graph (STG). From χ we now define a STG $\mathcal{G} = (\mathcal{E}, \mathcal{P})$. For each node n in χ, we create a node in $\mathcal{G}$. For each directed edge $(n \rightarrow n')$ in χ we create two directed edges in $\mathcal{G}$; one of the same direction $(n \rightarrow n')$ and one of the opposite direction $(n' \rightarrow n)$. This is done because in our context, if a property connects two nodes in χ, then these nodes are *semantically biconnected*. For example, in case of two entities $e_1 =$ `Sebastian_Vettel` and $e_2 =$ `Scuderia_Ferrari`, we can either say that $(e_1, "team", e_2)$ or that $(e_2, "driver", e_1)$, the difference only being in how we name the property.

Weighting the edges. We consider the different scenarios where the walker can lie and move to, and specify the weights of the edges.

Case 1: The walker lies at an entity of interest e and he moves to a document $d \in D_Q$, i.e., the movement is along the edge $n \rightarrow n'$, where $n = e, e \in E_Q$ and $n' = d, d \in D_Q$.

Considering relativeness and timeliness, the score of a document $d \in D_Q$ can be defined as:

$$ScoreD^{f,t}(d) = Score^f(d) \cdot Score^t(d) \tag{13}$$

The weight of the edge $(n \rightarrow n')$ in this case $(n = e, n' = d)$ for both `AND` and `OR` semantics can be defined as:

$$weight(e \rightarrow d) = \frac{ScoreD^{f,t}(d)}{\sum_{d' \in D_Q}(ScoreD^{f,t}(d'))} \tag{14}$$

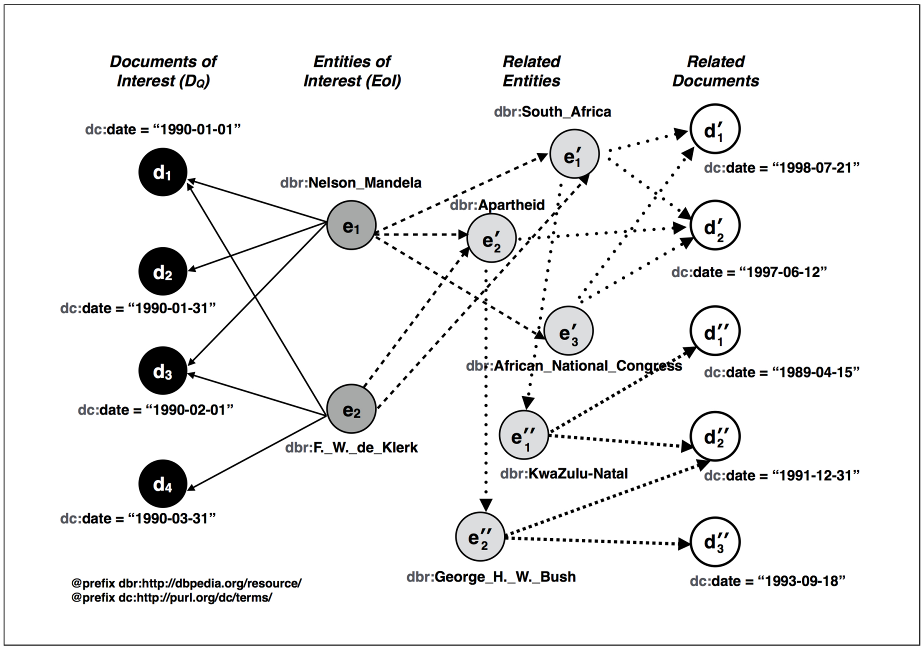

Figure 4: An example of a semantic graph of documents and entities.

Case 2: The walker lies at an entity of interest e and moves to a related entity e'. Movement is along the edge $(n \rightarrow n')$ where $(n = e, n' = e')$. Considering relatedness, for both `AND` and `OR` Semantics we use the following criteria for assigning weights to the edges:

$$weight(e \rightarrow e') = \frac{ScoreE(e')}{\sum_{e' \in E_D \setminus E_Q}(ScoreE(e''))} \tag{15}$$

We also have to take into consideration that the weight of the outgoing edges must represent transition probabilities, i.e., they must sum to 1. Thus, the weight from an entity-node e to a connected node n can be generally defined as:

$$weight(e \rightarrow n) = \begin{cases} p_1 \cdot \frac{ScoreD^{f,t}(d)}{\sum_{d' \in D_Q}(ScoreD^{f,t}(d'))} & n = d \\ (1 - p_1) \cdot \frac{ScoreE(e')}{\sum_{e' \in E_D \setminus E_Q}(ScoreE(e''))} & n = e' \end{cases} \tag{16}$$

Figure 5 shows an example of an STG of documents of interest, entities of interest and related entities along with the edges and some of the edge weights for the entities of interest. The criteria for assigning weight in the first and second case is based upon the notion that from an entity of interest a surfer is more likely to move to a document that mentions the entity of interest many times and is published in important time periods (for EoI). It is also likely to move to a related entity that is important to the EoI.

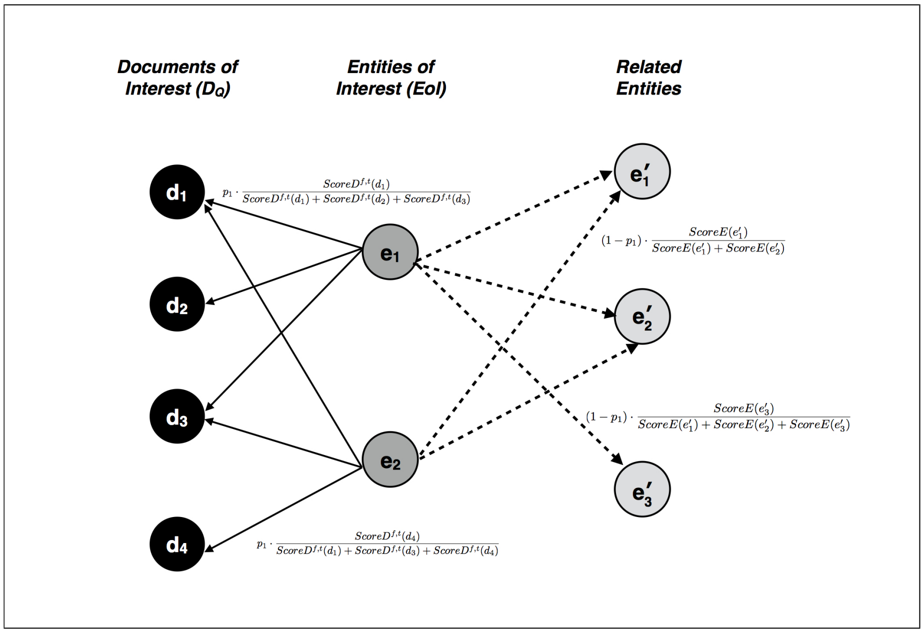
Documents of Interest (D_Q)
Entities of Interest (EoI)
Related Entities
d_1
d_2
d_3
d_4
e_1
e_2
e'_1
e'_2
e'_3
$p_1 \cdot \frac{ScoreD^{f,t}(d_1)}{ScoreD^{f,t}(d_1) + ScoreD^{f,t}(d_2) + ScoreD^{f,t}(d_3)}$
$(1-p_1) \cdot \frac{ScoreE(e'_1)}{ScoreE(e'_1) + ScoreE(e'_2)}$
$(1-p_1) \cdot \frac{ScoreE(e'_3)}{ScoreE(e'_1) + ScoreE(e'_2) + ScoreE(e'_3)}$
$p_1 \cdot \frac{ScoreD^{f,t}(d_4)}{ScoreD^{f,t}(d_1) + ScoreD^{f,t}(d_3) + ScoreD^{f,t}(d_4)}$

Figure 5: Biasing the link selection from an EoI-node.

Case 3: The walker is at a document d and moves to another document $d' \in D_Q$, i.e., the walker moves along the edge $n \rightarrow n'$ where $n = d$ and $n' = d'$. In this case, we use the following criterion for assigning weight to the edge:

$$weight(d \rightarrow d') = \frac{ScoreD^m(d')}{\sum_{d'' \in D_Q, d'' \neq d}(ScoreD^m(d''))} \tag{17}$$

Case 4: The walker is at a document d and moves to an entity of interest e, i.e., the movement is along the edge $(n \rightarrow n')$ where $n = d$ and $n' = e$. We do the weight assignment in this case as follows:

$$weight(d \rightarrow e) = \frac{1}{|ents(d) \cap E_Q|} \tag{18}$$

This means that we define that it is equiprobable for a walker to move to any of the entities of interest from a document containing the EoI. In case the document mentions only one EoI, then weight becomes equal to 1.

Like in the first and second case, in these cases as well, the weight of the outgoing edges must represent transition probabilities and sum to 1. Hence, we define the weight from an entity-node d to a connected node n as:

$$weight(d \rightarrow n) = \begin{cases} p_2 \cdot \frac{ScoreD^m(d')}{\sum_{d'' \in D_Q, d'' \neq d}(ScoreD^m(d''))} & n = d' \\ (1 - p_2) \cdot \frac{1}{|ents(d) \cap E_Q|} & n = e \end{cases} \tag{19}$$

Figure 6 shows an example of an STG graph of documents of interest and entities of interest along with the edges and some of the edge weights for a document of interest. In the third and fourth case, the criteria for assigning weight is based upon the notion that if the surfer is at a document he finds interesting, he would more likely move to another document closely published (before or after) to it to find out more about the event that occurred at the time period. For example, a historian stumbling upon an article mentioning about Nelson Mandela's release from Victor Vester prison might want to know more about the details of his release or the events that occurred shortly before or after his release. And if the document is not interesting enough or does not need to be explored further, the surfer moves back to one of the entities of interest.

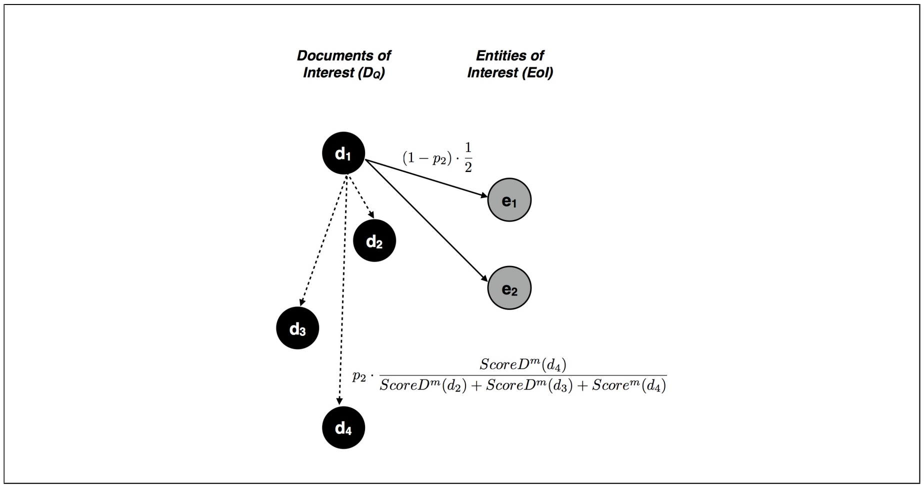
Documents of Interest (D_Q)
Entities of Interest (EoI)
d_1
$(1-p_2)\cdot\frac{1}{2}$
e_1
d_2
e_2
d_3
$p_2 \cdot \frac{ScoreD^m(d_4)}{ScoreD^m(d_2)+ScoreD^m(d_3)+Score^m(d_4)}$
d_4

Figure 6: Biasing the link selection from a document of interest-node.

Case 5: The walker is at a related entity e' to the EoI and he moves to an entity of interest e, that is, it travels along the edge $(n \rightarrow n')$ where $n = e'$ and $n' = e$. We define the weight assignment as:

$$weight(e' \rightarrow e) = \frac{1}{|edge(e', e'')|} \qquad e'' \in E_Q \tag{20}$$

This equation also translates as that there is equal probability for a walker to move from a related entity e' to any of the entities of interest connected to it by an edge. In case the related entity is connected to only one entity of interest, the weight of the edge becomes 1.

Case 6: The walker is at a related entity e' and moves to a document $d' \in docs(e')$, that is, movement is along the edge $(n \rightarrow n')$ where $n = e'$ and $n' = d'$. We take into consideration relativeness and define the weight assignment as:

$$weight(e' \rightarrow d') = \frac{ScoreD^f(d')}{\sum_{d'' \in docs(e')}(ScoreD^f(d''))} \qquad d' \in docs(e') \tag{21}$$

This equation also translates as that there is equal probability for a walker to move from a related entity e' to any of the entities of interest connected to it by an edge. In case the related entity is connected to only one entity of interest, the weight of the edge becomes 1.

Case 7: The walker is at a related entity e' and moves to another entity e'', that is, movement is along the edge $(n \rightarrow n')$ where $n = e'$ and $n' = e''$. Using relatedness we define the weight assignment as:

$$weight(e' \rightarrow e'') = \frac{ScoreE(e'')}{\sum_{e''' \in E_D \setminus E_Q}(ScoreE(e'''))} \tag{22}$$

Here we use the same way to assign weight as we did in the second case where the movement was from an EoI e to a related entity e'.

Because the weight of the outgoing edges must sum to 1 as they represent transition probabilities, we define the weight from a related entity-node e' to a connected node n as:

$$weight(e' \rightarrow n) = \begin{cases} p_3 \cdot \frac{1}{|edge(e',e'')|} & e'' \in E_Q, n = e \\ p_4 \cdot \frac{ScoreD^f(d')}{\sum_{d'' \in docs(e')}(ScoreD^f(d''))} & n = d', d' \in docs(e') \\ (1 - p_3 - p_4) \cdot \frac{ScoreE(e'')}{\sum_{e''' \in E_D \setminus E_Q}(ScoreE(e'''))} & n = e'', p_3 + p_4 \leq 1 \end{cases} \tag{23}$$

Figure 7 shows an example of an STG of entities of interest, related entities and related documents along with the edges and edge weights for a related entity. Also, in the case where the walker is at further related entity e'' and wants to make a transition, the same cases 5, 6 and 7 apply.

Case 8: The walker is at a document d' which he has reached from a related entity e' and now he wants to move back to an entity of interest e and display the initial results.

This forms a special case because although there may not exist an edge between the EoI e and the document d' as it may be the case that $d' \notin docs(e)$. But still we allow this transition as the walker after travelling to a document mentioning a related entity he may always choose to reset and display initial results as his original objective was to find interesting documents related to the entities of interest EoI in a specific time period. In this case, if $\nexists edge(d', e)$ then we create an $edge(d', e)$ where $e \in E_Q$ and the edge direction is from d' to e. The weight in this case is defined as:

$$weight(d' \rightarrow e) = \frac{1}{|E_Q|} \tag{24}$$

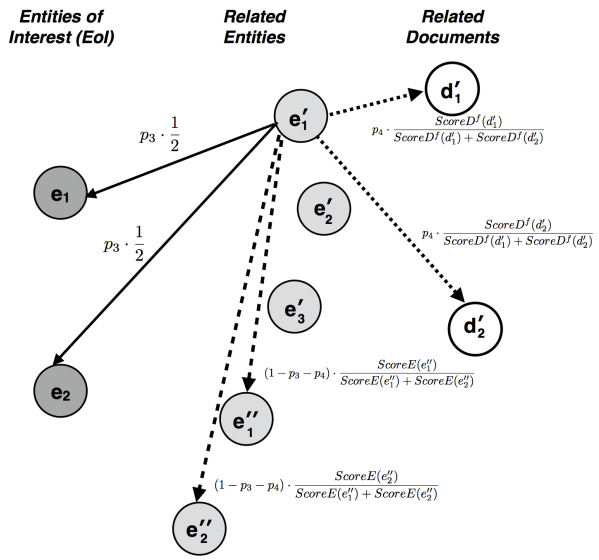
Entities of Interest (EoI)
Related Entities
Related Documents
d'_1
e'_1
$p_4 \cdot \frac{ScoreD^f(d'_1)}{ScoreD^f(d'_1) + ScoreD^f(d'_2)}$
$p_3 \cdot \frac{1}{2}$
e_1
e'_2
$p_4 \cdot \frac{ScoreD^f(d'_2)}{ScoreD^f(d'_1) + ScoreD^f(d'_2)}$
$p_3 \cdot \frac{1}{2}$
e'_3
d'_2
$(1 - p_3 - p_4) \cdot \frac{ScoreE(e''_1)}{ScoreE(e''_1) + ScoreE(e''_2)}$
e_2
e''_1
$(1 - p_3 - p_4) \cdot \frac{ScoreE(e''_2)}{ScoreE(e''_1) + ScoreE(e''_2)}$
e''_2

Figure 7: Biasing the link selection from a related entity-node.

Case 9: The walker is at a document d' which he has reached from a related entity e' and now he moves to an entity e' extracted from it. The criterion for assigning weight in this case will be as follows:

$$weight(d' \to e') = \frac{1}{|ents(d') \setminus E_Q|} \tag{25}$$

This means that there is an equal probability of the walker to move from a document d' to any of the extracted entities except for the entities of interest. In case the walker moves from EoI e to a related entity e' and further to related entity e'' before moving to a document d'', the cases 8 and 9 apply.

For the transition probabilities in the eighth and ninth case, we need to make the sum of all the outgoing edges from d' to be equal to 1. Thus, we modify the edge weights from document-node d' to a node n as:

$$weight(d' \to n) = \begin{cases} p_5 \cdot \frac{1}{|E_Q|} & n = e \\ (1 - p_5) \cdot \frac{1}{|ents(d') \setminus E_Q|} & n = e' \end{cases} \tag{26}$$

Figure 8 shows an example of an STG of related documents, related entities and entities of interest along with the edges and their weights for a related document.

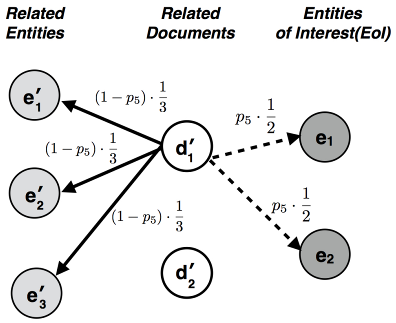

Figure 8: Biasing the link selection from a related document-node.

3.4 Analyzing the STG

For a node n, let $in(n)$ be the set of nodes that point to n. The *PageRank-like* value $r(n)$ is defined as:

$$r(n) = d \cdot Jump(n) + (1-d) \cdot \sum_{n' \in in(n)} (weight(n \to n') \cdot r(n')) \tag{27}$$

References