FEUP at TREC 2017 OpenSearch Track

Graph-Based Models for Entity-Oriented Search

José Devezas
INESC TEC and DEI, FEUP
Porto, Portugal
jld@fe.up.pt

Carla Teixeira Lopes
INESC TEC and DEI, FEUP
Porto, Portugal
ctl@fe.up.pt

Sérgio Nunes
INESC TEC and DEI, FEUP
Porto, Portugal
ssn@fe.up.pt

ABSTRACT

We describe our participation in the TREC 2017 OpenSearch Track, where we explored graph-based approaches for document representation and retrieval. We tackled the problem as an entity-oriented search task over the SSOAR site (Social Science Open Access Repository), using the *title* and the *abstract* as a text block and the remaining metadata as a knowledge block. Our main goal for this edition was to compare the graph-of-word, a text-only representation, with the graph-of-entity, a combined data representation that we are working on. The proposal is that, by combining text and knowledge through a unified representation, we will be able to unlock novel weighting strategies capable of harnessing all available information and ultimately improving retrieval effectiveness. Unfortunately, due to a technical problem with the OpenSearch track infrastructure, we were unable to obtain feedback for the real round during August 2017. As an alternative, we were offered the opportunity to participate in a third extraordinary round, happening during October 2017, as well as available feedback from the period between the two official rounds, at the end of July 2017. We obtained an outcome of 0.375 for the graph-of-word and 0.167 for the graph-of-entity, based on only 29 impressions with clicks, out of a total of 4,683 impressions. According to this small number of clicked impressions, both models performed below the site's native search, with graph-of-entity performing below graph-of-word.

1 INTRODUCTION

Search has evolved from keyword-based approaches, as inspired by the back-of-the-book index, to entity-oriented approaches, where semantics has taken a central role. In entity-oriented search, natural language understanding — of queries and documents —, as well as the usage of structured data from knowledge bases, have become two fundamental tools to improve performance. This means that the better a user's information need is identified through query understanding and the better the information within a document is understood, the more likely the query will be matched with relevant documents or entities mentioned in those documents. This frequently results in improved retrieval effectiveness and, therefore, increased user satisfaction.

In the last few years, there has been work in graph-based approaches for information retrieval [2, 7], and also a growing need for unified models [3, 4, 8, 9]. While many solutions focus on the integration of signals obtained from text represented in an inverted index with signals obtained from external knowledge bases like Wikipedia [1], there have been few attempts at modeling text and knowledge in an unified manner, as a single data structure.

In this experiment, we build on the idea of the graph-of-word [7] to propose a novel graph-based model that combines text and knowledge within a single representation. The graph-of-word is a document-based graph [2] where terms are represented by nodes, providing directed links to the following $n = 3$ terms, as a way of capturing context. The graph-of-entity also captures the links between terms, still accounting for term dependence, but also the links between entities, similarly to an RDF[1] graph and, perhaps more importantly, it also captures the links between terms and entities, in an attempt to connect unstructured and structured data. On one side, the model is capable of representing the properties of terms in a text document, as well as the properties of entities and relations in a knowledge base. On the other side, it provides a way to cross reference all available information, independently of the source, as well as an opportunity to define a common set of operators that simultaneously work for text corpora and knowledge bases.

2 DATASET AND API INTERACTION

The dataset was provided to participants through the Living Labs API[2] in JSON format. In particular, an array of documents could be accessed through a GET request to `/api/v2/participant/docs`. Each document contained a *site_id* (i.e., "ssoar" for the 2017 occurrence), a *doc_id*, a *creation_time*, a *title*, and a *content* object. The *content* object contained relevant metadata about the document, including the *abstract* and other fields like *subject* or *type*. We divided all available metadata into a text block (*title* and *abstract*), and a knowledge block (*author*, *language*, *issued*, *publisher*, *type*, *subject* and *description*).

We also obtained the train and test queries from `/api/v2/participant/query`, generating rankings for each query based on our implementation of the graph-of-word and graph-of-entity (both detailed in the next section). Finally, we added any missing documents to the end of the results list, based on the provided rankings for each query, available at `/api/v2/participant/doclist/(qid)`. The `doclist` rankings corresponded to candidate documents, provided by the site, that could be used for instance for reranking and should be a part of the submitted runs — our approach did not, however, take advantage of this information. We submitted and activated three runs, one during the trial round (*goe_trec2017*) and two

TREC 2017, November 2017, Gaithersburg, Maryland, USA
2017.

[1] The Resource Description Framework (RDF), is a metadata data model for representing information in the web. More details can be found at https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. RDF is frequently used to represent knowledge bases such as DBpedia (the structured version of Wikipedia) defining triples of subject, predicate and object, that can be seen as a graph.

[2] Living Labs is an open source framework for the evaluation of information retrieval systems based on an interleaving approach where the participant's results are combined with the results provided by the site. Living Labs is available at https://bitbucket.org/living-labs/ll-api.

during the real round (*gow_trec2017-real_round* and *goe_trec2017-real_round*). Run submission was done through a PUT request per query to `/api/v2/participant/run/(qid)`.

3 REPRESENTATION AND RETRIEVAL

In our experimental workbench, we implemented the graph-based models using a graph database per index (Neo4j[3]) and the ranking functions using the Gremlin DSL[4]. The goal of this work was to propose a graph-based representation for combined data (text and knowledge), while using the graph-of-word as a text-only baseline. Figure 1 illustrates the graph-of-word and graph-of-entity models, described in the following sections, based on the first sentence of the Wikipedia article for "Semantic Search" (i.e., our example collection consists of only one document with a single sentence):

```
Semantic search seeks to improve search
accuracy by understanding the searcher's
intent and the contextual meaning of terms
as they appear in the searchable dataspace,
whether on the Web or within a closed system,
to generate more relevant results.
```

3.1 Graph-of-word

Representation. The graph-of-word [7] is a document-based graph [2], where each node represents a term and each edge links to the following terms within a window of size n. The graph is unweighted, but directed, defying the term independence assumption of the bag-of-words approach. Figure 1a shows a graph-of-word instance for the first sentence of the Wikipedia article on "Semantic Search", using a window size of $n = 3$. The graph-of-word is thus able to capture the context of each term within a particular document.

Retrieval. In the original graph-of-word implementation, the term weight (TW) metric was precomputed based on the indegree of each term node and stored in the inverted index to be used in place of the term frequency (TF). In our implementation, however, this was done in real time by filtering over the union of all document-based graphs and selecting a given subgraph based on a *doc_id* attribute stored in the edge. This is a less efficient solution, but it simplified the process of exploring and developing the novel graph-of-entity model, based on the graph-of-word, by defining a common representation framework. Additionally, the focus of our experiment was retrieval effectiveness; we were not particularly concerned with index efficiency.

Equation 1 shows the ranking function used for retrieval over the graph-of-word.

$$TW\text{-}IDF(t,d) = \frac{tw(t,d)}{1 - b + b \times \frac{|d|}{avdl}} \times log\frac{N+1}{df(t)} \tag{1}$$

The formula was derived from the TF-IDF approach as defined by Lv and Zhai [5], replacing the $tf(t,d)$ function by the $tw(t,d)$ given by the term node indegree on the graph-of-word for document d. For example, in Figure 1a, we assume the query `[ web search system ]` and find that the largest term weight, $tw(t,d) = 3$, was assigned to "search", while "web" and "system" were tied in second place with $tw(t,d) = 2$. The parameter b was fixed at 0.003, since, according to the authors [7], it consistently produced good results across various collections, with $|d|$ representing the length of document d, $avdl$ the average length of all documents in the corpus, N the number of documents in the corpus, and $df(t)$ the document frequency of term t in the corpus. In our implementation, both $|d|$ and $avdl$ were approximated by the number of edges within the respective document-based graph, since we did all computations directly based on the graph.

3.2 Graph-of-entity

Representation. The graph-of-entity is a collection-based graph [2], where nodes can represent either terms or entities and edges can be of three types: term→term, entity→entity and term→entity. While the graph-of-entity was inspired by the graph-of-word, it only captures term sequence instead of term context in term→term relations, that is, the window size is always one[5]. Additionally, we also encode entity→entity relations in the graph as a way of representing knowledge associated with the document (e.g., obtained from an information extraction pipeline applied to the text, or simply consisting of Wikipedia concepts linked in some manner). Finally, term→entity relations are established based on a substring matching approach. The goal for the first version of this model was to keep it simple (e.g., refraining from using similarity edges), but highly connected (i.e., using weak, but abundant connections), while modeling knowledge and capturing text properties and the relations between text and knowledge.

Retrieval. We rank entities in the graph-of-entity based on the entity weight (EW) for an entity e and a query q. A set of seed nodes S_q are derived from query q, based on the links between query term nodes and entity nodes; when there are no entity nodes linked to a query term node, then the term node becomes its own seed node. This step provides a representation of the query in the graph, that will be used as the main input for the ranking function.

Next, we present a formal definition for $EW(e,q)$, based on three main score components: coverage $c(e, S_q)$, confidence weight w_s for a seed node s, and the average weighted inverse length of the path between a seed node s and an entity node e to rank.

Let us assume a graph-of-entity represented by an attributed labeled multigraph $G_e = (V,E)$, similar to the one depicted in Figure 1b, and a set of operations over G_e to obtain a ranking of entity nodes with a *doc_id* attribute.

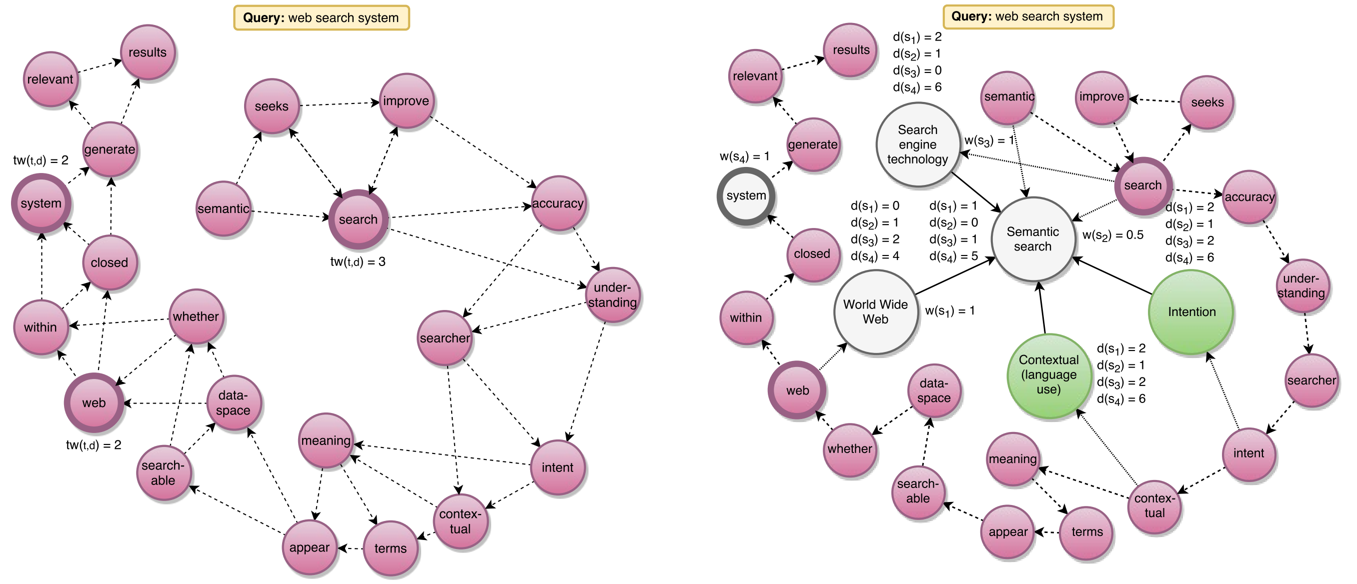
Let q be a query represented by a sequence of term nodes q_n and let e be an entity node that we want to rank (i.e., it has a *doc_id* attribute).

Let Sq be the set of seed nodes derived from query q. For each node q_n in the graph that represents a term in query q, we obtain the set of seed entity nodes S_{q_n} that are adjacent to term node q_n. Whenever q_n has no entity node neighbors, $S_{q_n} = \{q_n\}$. The set S_q of all seed nodes derived from query q is then given by $S_q = \bigcup_{q_n} S_{q_n}$. This means that S_q will contain all entity nodes adjacent to query term nodes, as well as query term nodes that are not adjacent to any entity node (i.e., they represent themselves).

[3] https://neo4j.com/

[4] Apache Gremlin is a domain-specific language for graph querying. More information at https://tinkerpop.apache.org/gremlin.html.

[5] As a side note, while outside of the scope of this work, this decision has a particularly high impact in reducing the number of edges and thus the necessary storage space.



(a) Graph-of-word (document-based graph; text-only). Nodes represent terms. Query term nodes are identified by a thicker border.

(b) Graph-of-entity (collection-based graph; text+knowledge). Smaller (pink) nodes represent terms, while larger (green) nodes represent entities. A seed node for the given query is displayed in white. Query term nodes are identified by a thicker border.

Figure 1: Graph-based representations for the first sentence of the "Semantic Search" Wikipedia article.

For example, in Figure 1b, given query $q = q_1, q_2, q_3$ the seed nodes are given by $S_q = \{e_1, e_2, e_3, q_3\}$, where:

Vertex	Name	Source
	ENTITIES	
e_1	*Search engine technology*	q_2
e_2	*Semantic search*	q_2
e_3	*World Wide Web*	q_1
	TERMS	
q_1	*web*	–
q_2	*search*	–
q_3	*system*	q_3

Let p_{es} be a path between an entity node e and a seed node s, as defined by a sequence of vertices $e, v_1, \cdots, v_{(\epsilon-1)}, s$ in the undirected version of G_e. Let P_{es} be the set of all paths p_{es} between e and s. Assume the function $\epsilon(p_{uv})$ as the length of a given path p_{uv} between vertices u and v, representing the number of traversed edges[6].

Equation 2 can be read as the ratio between the number of paths linking entity node e and seed nodes s and the total number of seed nodes S_q. That is, the coverage represents the fraction of reachable seed nodes from a given entity.

$$c(e, S_q) = \frac{|\{s \in S_q | \exists p_{es} \in P_{es}\}|}{|S_q|} \tag{2}$$

Let e_{ts} be the edge incident to both a term node t and a seed node s. Equation 3 can be read as the confidence weight of seed node s. It represents the confidence that a seed node is a good representative of the query term it was derived from.

$$w_s = \begin{cases} \frac{|\{e_{ts} \in E(G_e) | \forall t \exists q (t = q_n)\}|}{|\{e_{ts} \in E(G_e)\}|} & \text{if } s \text{ is an entity node} \\ 1.0 & \text{otherwise} \end{cases} \tag{3}$$

Finally, Equation 4 shows the ranking function for a given entity e and query q.

$$EW(e, q) = c(e, S_q) \times \frac{1}{|S_q|} \sum_{s \in S_q} \left(\frac{1}{|P_{es}|} \sum_{p_{es} \in P_{es}} w_s \frac{1}{\epsilon(p_{es})} \right) \tag{4}$$

The query is only used to obtain the seed nodes S_q that best represent q in the graph. This is analogous to a query entity linking step. The remaining steps are quite straightforward. We obtain the average weighted inverse length of the path between each seed node s and each entity e. Assuming that the seed nodes are good representatives of the query in the graph, the closer an entity is from all seed nodes, the more relevant it is — closeness is measured by the inverse length of the path. Given there is a degree of uncertainty associated with the selection of seed nodes, we scale this value based on the confidence weight of the seed node — an entity close to a high confidence seed node is more relevant than an entity

[6] In practice, we also defined a maximum distance threshold to compute the length of a path between two nodes. That is, no paths above the given threshold were considered. For this particular experiment, we used a maximum distance of one, which is an extremely conservative value.

close to a low confidence seed node, but an entity further apart from a high confidence seed node might be on par, or even more relevant.

4 EVALUATION

The evaluation in TREC OpenSearch differs from classical TREC evaluation based on test collections. For this track, participants are provided with a Living Labs API, where they can register for a set of available sites. The API then provides documents to index and queries to generate rankings. Provided queries correspond to the most frequently issued within the site, thus increasing the chance they will appear again in the future. As one of the provided queries is issued by the site, a participant is selected and the results for participant and site are interleaved using Team Draft Interleaving [6]. Evaluation is then carried based on the clickthrough rate and, for the assessment, we account for the fraction of wins of the participant over the site. In the 2017 edition, only the Social Science Open Access Repository (SSOAR) was available as a site, providing 39,492 documents to index, along with 1,165 queries (676 train queries and 489 test queries). Our goal was to compare the graph-of-word with the graph-of-entity, based on the fraction of wins either model obtained against the site's results. While the evaluation was carried individually for each model and compared with the site's search model, using a different set of queries, this provides initial feedback as to whether the graph-of-entity is comparable or performs better than the graph-of-word — the hypothesis is that by including structured data and providing a combined data representation approach the results will improve.

4.1 Technical Issue

Unfortunately, there was a technical problem with the load balancer on the side of the OpenSearch track infrastructure that resulted in our team receiving no feedback for the real round, during August 2017. The criterion for a given run from any participant to be selected is based on the lowest number of impressions it has received so far. A rather unpredictable issue with the priority strategy of the load balancer led to our runs never being selected. This happened for two main reasons: (*i*) the early activation of our run in July 17, 2017, which by itself would have posed no issues; and (*ii*) the fact that the SSOAR site was kept active throughout the two rounds, even when no round was scheduled to run. The combination of these two events resulted in a total of 4,000 impressions for our runs, during July 2017, that were never surpassed by any other participant during August 2017, possibly due to lower traffic during the summer. Since the runs from every other participant were continuously lower in number of impressions, our runs were never selected to be displayed and thus received no feedback during August 2017.

The organization of TREC 2017 OpenSearch Track acknowledged and detailed this issue and provided two options: (*i*) sharing the feedback from the end of July 2017, which is not available directly via the API, since there was no official round happening at the time; and (*ii*) run an extraordinary round during October 2017. While either option cannot be considered comparable with the approaches from other participants, our main focus was on comparing the two models we propose, making them both valuable.

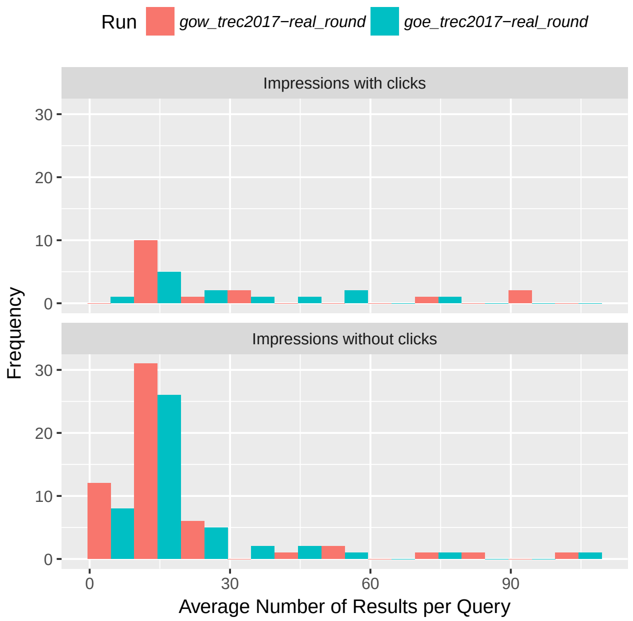


Figure 2: Result size distribution per run (bin width = 10).

Table 1: Outcome for the two graph-based models, during the dead period of July 17–31, 2017.

Run	Impressions		Outcome	Wins	Losses	Ties
	Total	Clicked				
gow	2,342	16	0.375	6	10	0
goe	2,341	13	0.167	2	10	1

4.2 Feedback for July 17–31 2017

Feedback collected between July 17, 2017 and July 31, 2017 corresponds to what we call a dead period, as no official round was scheduled to run at that time. This is not usually supplied to participants, however, given the technical issue described in Section 4.1, such data was provided to us as a JSON dump.

Figure 2 shows the average number of results per query, as provided to the users, based on the feedback for the dead period. Analyzed results were automatically generated by the Living Labs system from the interleaving of documents provided by the participant and the site. We distinguish between each run with different colors and separately analyze impressions with and without clicks. As we can see, the number of results varies between 0 and 100 and there is a significantly lower number of impressions with clicks. We also find that lists of results with less than 10 documents were never clicked, which might be an indicator of a poor precision at 10 for both models. Overall, the graph-of-word retrieves a slightly larger number of documents when compared to the graph-of-entity.

Figure 3 shows the rank distribution for clicked results. As we can see, most of the clicked results were at the top of the ranks and, overall, the graph-of-word achieved a higher number of clicks for the top ranks than the graph-of-entity.

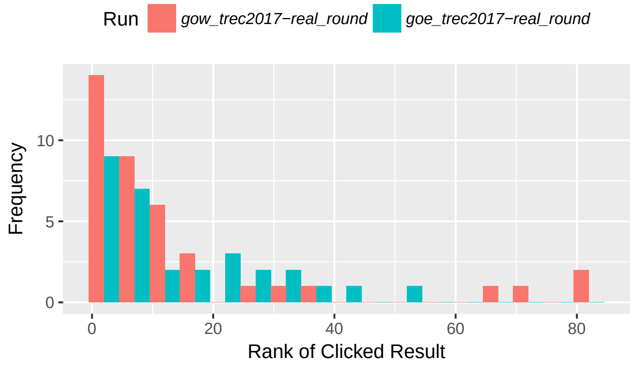


Figure 3: Rank distribution per run (bin width = 5).

Table 1 shows the outcome for the *gow_trec2017-real_round*, where we tested the graph-of-word (*gow*), and the *goe_trec2017-real_round*, where we tested the graph-of-entity (*goe*), based on feedback for the dead period. This enabled us to evaluate the two graph-based models, by comparing each of them, individually, with the existing model used by SSOAR. This comparison was based on the outcome given by the fraction of wins for the participant (not including ties). For an outcome of 0.5, the two retrieval models would be equivalent, while for a value higher than 0.5, the participant's model would be better than the site's model. The results were not particularly encouraging, with both graph-based models achieving an outcome under 0.5 — 6 wins versus 10 losses for the graph-of-word and 2 wins versus 10 losses for the graph-of-entity. While there were over 2,300 impressions for each run, only a small fraction of about 15 impressions contained clicked results ($\sim 0.5\%$).

5 CONCLUSIONS

We have successfully participated in TREC 2017 OpenSearch Track, despite some technical issues with the Living Labs framework (that have since been corrected by the development team). We implemented a document-based graph (graph-of-word) and a collection-based graph (graph-of-entity) using a common approach supported on a graph database. The graph-of-entity is a novel graph-based model for representation and retrieval that we proposed and described in detail. This model is able to represent combined data (text and knowledge), while capturing the properties of text through the sequences of terms, the properties of knowledge through the relations between entities, and the relationships between terms and entities through the occurrence of terms within entity names. We were able to assess the two models, but only based on feedback from a period between the trial round and the real round, because of the technical issue. Despite the lack of strong evidence, according to our evaluation data, both models underperformed SSOAR's native search and, when comparing the models amongst themselves, we found no evidence of graph-of-entity performing better than the graph-of-word.

5.1 Future Work

There is a lot of work ahead, in particular regarding the development of the graph-of-entity as a representation model. We would like to experiment with different types of edges, linking the same two types of nodes (terms and entities). Our goal is to build a graph-based model that is able to be extended with novel links without the requirement to change the ranking function to accompany the semantics of the included links. We also aim at building a model that supports different tasks beyond entity ranking, in order to prove its generality.

Regarding the graph-of-word, we would like to correctly compute the document length and average document length based on the actual number of words in a document, changing our representation of the model (supported on a graph database) to include that information and providing a more accurate comparison.

Additionally, we expect to analyze the feedback from the extraordinary round of October 2017, in order to verify whether the performance of the graph-based models is actually below the native search for the SSOAR site and, moreover, whether the graph-of-entity in fact underperforms the graph-of-word. If this happens, it might indicate that modeling term context might be a fundamental step of graph-based indexing solutions. Finally, there is external evidence that the graph-of-word can outperform BM25, which leads us to believe that either 29 impressions with clicks is not enough to assess the effectiveness of the models, or that the dataset is not ideal to index with this type of strategy. In that case, we might want to investigate whether there is a particular type of collection that benefits from graph-based over inverted file based strategies.

ACKNOWLEDGMENTS

José Devezas is supported by research grant PD/BD/128160/2016, provided by the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

REFERENCES

[1] Hannah Bast, Björn Buchhold, Elmar Haussmann, et al. 2016. Semantic Search on Text and Knowledge Bases. *Foundations and Trends in Information Retrieval* 10, 2-3 (2016), 119–271.

[2] Roi Blanco and Christina Lioma. 2012. Graph-based term weighting for information retrieval. *Information Retrieval* 15, 1 (2012), 54–92. https://doi.org/10.1007/s10791-011-9172-x

[3] José Devezas and Sérgio Nunes. 2017. Graph-Based Entity-Oriented Search: Imitating the Human Process of Seeking and Cross Referencing Information. *ERCIM News. Special Issue: Digital Humanities* 111 (Oct. 2017), 13–14.

[4] Pedro Domingos. 2015. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World.* Basic Books. https://books.google.pt/books?id=glUtrgEACAAJ

[5] Yuanhua Lv and ChengXiang Zhai. 2011. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011.* 7–16. https://doi.org/10.1145/2063576.2063584

[6] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. 2008. How Does Clickthrough Data Reflect Retrieval Quality?. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08).* ACM, New York, NY, USA, 43–52. https://doi.org/10.1145/1458082.1458092

[7] François Rousseau and Michalis Vazirgiannis. 2013. Graph-of-word and TW-IDF: new approach to ad hoc IR. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management.* ACM, 59–68.

[8] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-Entity Duet Representations for Document Ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017.* 763–772. https://doi.org/10.1145/3077136.3080768

[9] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Ed Hovy. 2017. JointSem: Combining Query Entity Linking and Entity based Document Ranking. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM 2017).*