Entity Ranking on Graphs: Studies on Expert Finding

Henning Rode[1], Pavel Serdyukov[1], Djoerd Hiemstra[1], and Hugo Zaragoza[2]

[1]CTIT, University of Twente, The Netherlands
{h.rode, p.serdyukov, d.hiemstra}@cs.utwente.nl

[2]Yahoo! Research, Barcelona, Spain
hugoz@yahoo-inc.com

ABSTRACT

Todays web search engines try to offer services for finding various information in addition to simple web pages, like showing locations or answering simple fact queries. Understanding the association of named entities and documents is one of the key steps towards such semantic search tasks. This paper addresses the ranking of entities and models it in a graph-based relevance propagation framework. In particular we study the problem of expert finding as an example of an entity ranking task. Entity containment graphs are introduced that represent the relationship between text fragments on the one hand and their contained entities on the other hand. The paper shows how these graphs can be used to propagate relevance information from the pre-ranked text fragments to their entities. We use this propagation framework to model existing approaches to expert finding based on the entity's indegree and extend them by recursive relevance propagation based on a probabilistic random walk over the entity containment graphs. Experiments on the TREC expert search task compare the retrieval performance of the different graph and propagation models.

1. INTRODUCTION

Most retrieval applications aim at ranking a set of documents according to a given query without taking into account whether the user's information need really requires to return complete documents. Current web search engines already try to offer further services. If a location or calculation query is recognized, the user gets instead of the normal list of webpages a map showing the place, respectively the result of the calculation. However, these services that try to understand the semantics of a given query are rather limited so far. The problem here is twofold: On the one hand, systems still need a better understanding of the query semantics. This issue is addressed in the field of question answering research [25, 19]. And on the other hand, once a system recognized a query correctly as searching for locations, dates, or persons, ranking methods have to be developed to serve the search for such entities. If someone is looking for important persons related to a certain historical event, it is necessary to directly recognize and rank all found persons in a collection.

While named entity recognition is a field of research since years [10], entity ranking has just started to attract the attention of researchers. We define the task as follows: *Given a keyword query, a text collection, and a set of entities occurring in the text collection, rank those entities according to the query.* The entities of interest could be specified in different ways. In the example query, we would just tell the retrieval system that we are interested in entities of the type *person*. In other cases, the user might even have a list of certain entities at hand, but still needs to rank them according to the query and corpus. This paper develops a general framework for entity ranking and compares different ranking techniques.

1.1 Expert Finding

A quite typical example of such an entity ranking task is the problem of *expert finding*. In expert finding, as performed in TREC's enterprise track [7], a system has to come up with a ranked list of experts with respect to a given topic of expertise, a corpus of enterprise documents, and a list of the employees of the company as possible candidates. Although we claim, that our approach in principle addresses entity ranking in general, we use the problem of expert finding in several ways. First, it serves as a source of inspiration for theory and methods that can be generalized to the entity ranking task. Furthermore, we have chosen the expert finding task for evaluation of our entity ranking approaches, since TREC's enterprise track provides data, topics and relevance assessments for this task, which allows to experimentally study the performance of our methods and to compare different approaches.

Expert finding is a young field of information retrieval research [11]. It has become popular after the upcoming of TREC's enterprise track [7]. Early approaches build query-independent profiles for each candidate expert by merging all documents related to the expert into one expert model. Experts are ranked then by measuring the similarity of their profile to the query [20]. Most effective approaches on the TREC task now measure instead the similarity between query and documents (usually emails), and infer thereafter an expert ranking from the top retrieved documents. When deriving expert ranks from related documents, we see again different strategies used. Algorithms of the one kind rank candidates by the aggregated relevance of all related top documents [2, 21]. Another kind of methods build query dependent social networks from the top retrieved documents [4, 5]. More precisely, so-called bibliographic coupling graphs are generated by using documents as links between persons (e.g. by utilizing `from` and `to` email fields). Candidates are ranked then on such social networks by popular *centrality measures*, such as Kleinbergs HITS algorithm [15].

However, these centrality based approaches have failed to show similar performance as the simpler aggregation methods so far. Both aggregated relevance and centrality based methods still ignore some properties of data. Methods using aggregated relevance do not reflect the relation between experts, whereas the centrality measures on the coupling graph simply model documents as unweighted links between candidates, neglecting their relevance to the query. We will show in this paper that graph-based approaches are able to incorporate both kinds of information.

1.2 Graph-based Entity Ranking

Whereas document, passage, or XML retrieval employs standard retrieval models – passages or nodes are regarded as small documents in that case – the same models would fail for entity ranking. The simple reason is that query words in general do not occur as a part of a named entity. Therefore, entity ranking is always based on the association between entities and documents. In general we will speak of *text fragments* instead of documents to capture also approaches that perform sentence or text window based entity ranking. We have seen in related work on expert finding that the basic approach is to first rank those text fragments according to the keyword query and thereafter propagating the relevance of the text fragments to their included entities, respectively experts. This relevance propagation step will be the main issue of this paper. We show that graph-based approaches are most useful here for several reasons. Firstly, a graph makes the propagation process transparent. It becomes easy to describe and to visualize. It also allows to recognize and use indirect connections of paths longer than one. Secondly, we show that even non-graph-based approaches for expert finding can often be interpreted in terms of a graph-based equivalent.

For the rest of this paper, one should keep in mind the general processing model of our approach. While the named entity recognition can take place beforehand, the query dependent processing is divided in the following three steps:

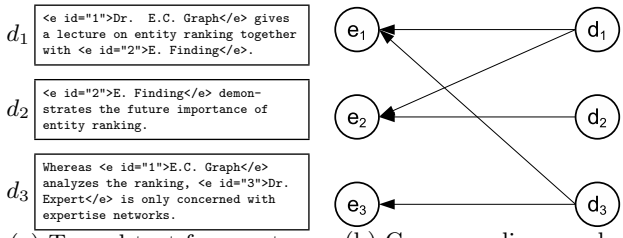
(1) Initial retrieval of text fragments,
(2) Building of an entity containment graph,
(3) Relevance propagation within the graph.

The first step remains a standard retrieval run on the entire text collection in order to select the most relevant text fragments. Those are used then for building a graph model that shows the containment of entities within retrieved text fragments (Section 2). In the third step we exploit the graph structure in order to rank the entities, respectively to propagate the relevance information (Section 3) within the graph.

2. ENTITY CONTAINMENT GRAPHS

This section proposes and discusses the modeling of appropriate graphs that represent the association between entities and documents. We will further on call them *entity containment graphs*.

Suppose we have a set of documents or sentences, in general *text fragments* D, with relevance scores from an initial retrieval run and a set of *entities* E, like persons, dates or



(a) Tagged text fragments (b) Corresponding graph

Figure 1: Entity Containment Graphs

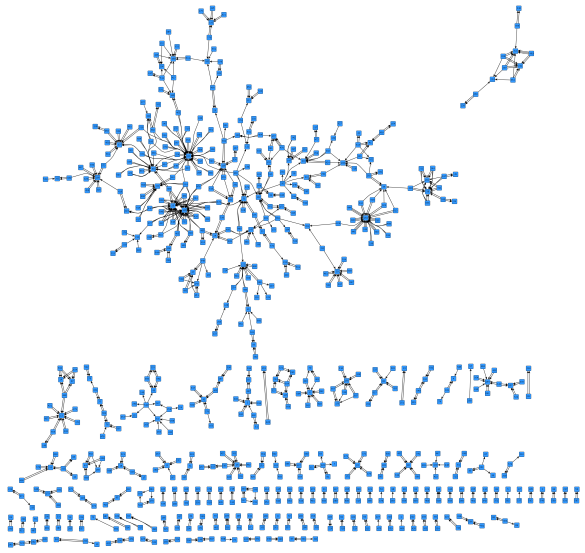
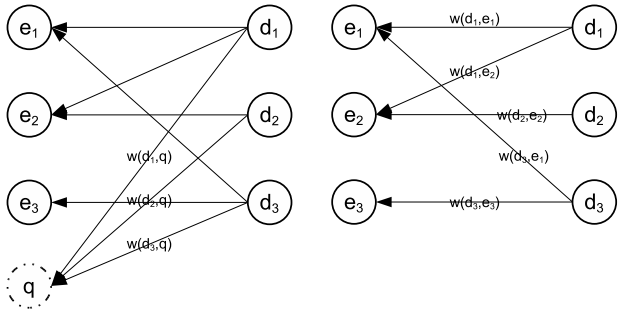


Figure 2: Real entity containment graph

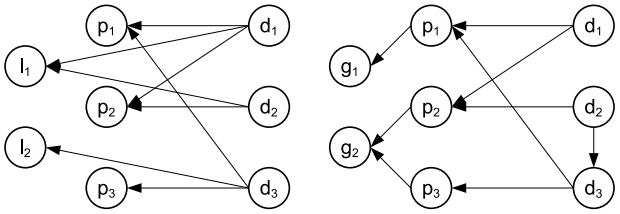
locations, that finally should get ranked according to the given query q. Furthermore, we know the containment relation of text fragments and entities, thus for each text fragment which entities are occurring inside. This relation can be represented in a graph, where both text fragments and entities become vertices and directed edges symbolize the containment condition. Such a graph is always bipartite, since all edges point from text fragments to entities.

Figure 2 shows a typical entity containment graph computed for one of the TREC queries. The graph representation provides several useful features of the entities. It shows in how many different documents they are occurring. Moreover, whether they are connected over common text nodes with other entities, or remain uncoupled (like all vertices in the lower part of the figure). Behind the last feature stays the hypothesis that entities mentioned in the same text fragment also have a stronger relation to each other than those which never appear together. Notice that in contrast to the bibliographic coupling graph, which models documents as edges between entities, such a bipartite graph of text and entity vertices captures both the direct containment relation as well as the indirect 2nd-degree neighborhood of entities to each other. In the following we show several modeling options and parameters:



(a) Initial retrieval scores (b) Associations weights

Figure 3: Weighting Models



(a) Further entity types (b) Further connections

Figure 4: Modeling additional information

Modeling Prior Scores. The initial retrieval run does not only return a ranked list of text fragments, but also their corresponding relevance score according to the given query. The simplest way to incorporate such prior knowledge into the graph model is to assign weights $w(v)$ to the vertices:

$$w(v) = \begin{cases} 0 & \text{if } v \models \text{ entity,} \\ P(d|q) & \text{if } v \models \text{ document } d. \end{cases}$$

$P(d|q)$ denotes here the probability that document d is relevant given query q. If the applied score function does not directly return probabilities, the raw scores should be transformed into probabilities. The assigned weights can later be used for relevance propagation through the graph.

For recursive propagation models, such as random walks or HITS, the initial setting of prior vertex weights has no influence on the final relevance of the vertices. For those models, we change the graph slightly by adding a further "virtual" query vertex q to the graph, which is connected with all text fragments. Instead of vertex weights, we can now similarly assign edge weights $w(d, q)$ (see Figure 3(a)):

$$w(d, q) = P(d|q) \text{ if } d \models \text{ document}, q \models query.$$

The additional query node is represented here as an additional "entity" contained in all documents. In order to motivate this modeling, one should think of the query as a set of terms, which is indeed contained in those documents to a certain degree, corresponding to the initial score.

Association Weights between Documents and Entities. The graph contains a directed edge from the document to each included entity, however, it does not provide so far any information about the strength of this association, in other words, how important the entity is for the document. To include such information an edge weight function $w(d, e)$ can be defined (see Figure 3(b)). Without any further domain knowledge, all occurrences of an entity should be treated equally. In a better known domain, like the expert finding task, occurrences of an expert in a document might be weighted differently. If an expert is the author of an email, she / he is probably more influential on the content than another expert who is just mentioned somewhere in the text.

Including Further Entity Types. In expert finding and many other cases of entity ranking the focus of interest will lie on a certain *entity type*. When speaking of entity types we mean a categorization of entities into semantically meaningful groups, such as persons, locations, and dates. Although a task like expert finding might only be interested in expert entities, it might still be useful to include nodes of other entity types into the graph (see Figure 4(a)). The motivation behind such a graph expansion would be to show the connection between entities of different types and to increase the relevance propagation between them. If one would search for instance for important dates in the live of the painter Pablo Picasso, it is probably useful to add more than date entities to the graph. In this case, further person or location entities might reveal important connections as well.

Including Further Edges. The suggested entity containment graph only models the relation of documents and included entities. One modification could be to include further document to document or entity to entity edges (see Figure 4(b)). The first ones for links between documents, the second ones if the found entities are standing in a known relation to each other. We think here for instance of exploiting known hierarchical ontologies, like *Cape Town* is part of *South Africa*, or *21 April 2006* and *2006* are date entities supporting each other. In case of expert finding, enterprises will often have a hierarchical organization overview of its personnel. By including such additional edges, the graph gains a higher density and enables more relevance propagation, but it looses its strict bipartite property.

Controlling Graph Size and Topical Focus. Apart from the graph modeling itself, the most influential parameter on the graph size and density is the number of top ranked documents taken into account while building the graph. Notice that for the unweighted graph only the restriction to the top ranked documents makes the graph model query dependent. Hence, by including more lower ranked documents more included entities are found and usually the graph's density increases with the drawback of loosing the topical focus.

3. RELEVANCE PROPAGATION

Once having an entity containment graph, there are several relevance propagation models that can be used for ranking of the entity vertices. For abbreviation of the notation, $\Gamma^+(v)$ denotes the set of vertices adjacent to v over outgoing edges, respectively $\Gamma^-(v)$ marking those adjacent over incoming

edges. Furthermore, we use different letters to distinguish between a document vertex d and an entity vertex e.

All propagation methods are introduced in this section in their *weighted* version, that incorporates the initial query scores. However, a *unweighted* counterpart can always be obtained by simply setting all weights to 1. We will later compare the retrieval performance of the unweighted variants, depending purely on the structure of the graph with the weighted models that propagate the initial document weights through the graph network.

3.1 Maximal Retrieval Score

The simplest model of entity ranking can be described by the following process. Walking down the ranked list of documents, we add all included entities that have not been encountered before in that order to the final ranked list. The equivalent propagation model on the entity containment graph assigns to each entity vertex the weight of the highest ranked linked document node:

$$wMAX(e) = \max_{d \in \Gamma^-(e)} w(d).$$

Although the model is formalized within the graph-based framework, it ignores most of the features provided by the entity containment graph. We will refer to it later as a *baseline* ranking model in order to compare it to other relevance propagation models that consider more features of the graph.

3.2 Weighted Indegree

The theoretically most sound methods for expert finding proposed by Balog et al. [2] and Macdonald et al. [21] can be expressed as an expertise inference on a linear Bayesian network $q \rightarrow d \rightarrow e$:

$$P(e|q) = \sum_{d \in D} P(e|d)\ P(d|q).$$

It uses the query to find relevant documents and then candidate experts occurring in these documents. The higher the number of the most relevant documents mentioning a candidate expert, the higher its probability of being an expert. Thus the initial scores of documents related to candidate expert are aggregated with respect to the candidate. Talking in graph terms, this model calculates the *weighted indegree* of an expert candidate $wIDG(e)$ in the entity containment graph:

$$wIDG(e) = \sum_{d \in \Gamma^-(e)} w(d,e)\ w(d).$$

In contrast to the Bayesian network model, $w(d,e)$ and $w(d)$ are not necessarily probabilities here.

3.3 Probabilistic Random Walk

Although the linear Bayesian network provides a sound theoretical foundation for the direct inference from document to expert probabilities, such a propagation model does not take into account all features of the graph, yet. It is reasonable to extend the one-way inference model and to assume that (1) entities can also influence the relevance of documents and (2) entities affect the relevance of other entities if they are 2nd-degree neighbors, thus appear together in the same text fragment. In other words, relevance should not only be propagated from documents and accumulated at entity nodes, but should flow further through the graph.

Let us motivate the idea by outlining an expert finding scenario: We can easily imagine a searcher for expertise who got some list of highly relevant documents from a retrieval system. She realizes that the expertise, she is looking for, is partly contained in these documents and partly in the experts heads. Her search process can be seen as an (infinite) process involving the following actions at each step:

(1) At any time: (a) Randomly read a document, most probably from the top of the ranked list.

(2) After reading a document: (a) Consult a candidate expert mentioned in this document, or (b) check for other linked documents and read one of them.

(3) After consulting with an expert: (a) Consult another person which is recommended by the expert, or (b) read further documents mentioning this expert.

The above considerations can be modeled as a Markov process whose stationary probability distribution for candidate nodes should show the level of their expertise:

$$\begin{aligned} P(e) &= \lambda_1 \sum_d P(e|d)P(d) + \lambda_2 \sum_{e'} P(e|e')P(e'), \\ P(d) &= \lambda_0 P(d|q) + \lambda_1 \sum_e P(d|e)P(e) + \lambda_2 \sum_{d'} P(d|d')P(d'). \end{aligned}$$

with $\sum \lambda_i = 1$. The settings of the different λ_i steers the decision between the different optional steps in the above outlined process. The model does not contain a probability to consult an arbitrary expert at any time. For a uniform computation, one could add here zero probabilities to the model instead. Notice also that we implicitly added a query node (as in Figure 3(a)) and reverse edges to the graph model, since the document vertices are reachable from the query and from all entities. The entire graph is therefore connected.

Although the motivating scenario was described in terms of expert search, it is obviously not restricted to this type of entity ranking. The Markov process describes here a *random walk* on an entity containment graph. Apart from the edge transition probabilities $P(e|d), P(d|e)$, all necessary information for the iterative calculation of the stationary probabilities is already included in the entity containment graph. Adopting the work of Balog et al. [2], the association weights w(e,d) between documents and entities get simply normalized to real probabilities:

$$\begin{aligned} P(e|d) &= w(d,e) / \sum_{e' \in \Gamma^+(d)} w(d,e'), \\ P(d|e) &= w(d,e) / \sum_{d' \in \Gamma^-(e)} w(d',e). \end{aligned}$$

The above proposed generic model of the walk even assumes entity to entity edges and links between documents (as in Figure 4(b)). Unfortunately, the TREC data used in the experimental study does not have such additional links, but

for the completeness of the approach we wanted to show how to incorporate this type of relevance propagation as well. Our propagation model stands in close relation to the random walk on web graphs proposed by Shakery and Zhai [24] that also controls the walk to different *neighborhoods* of a node by a set of steering probabilities λ_i.

3.4 HITS

We have chosen the HITS algorithm in order to compare our random walk with other known graph-based weight propagation models. HITS was originally used to characterize a hyperlinked network of web-pages consisting of portal pages with a high number of outgoing links, so-called *hubs*, on the one hand, and cited content bearing pages with a higher number of in-links on the other hand, so-called *authorities*. We can easily transfer this distinction to our entity containment graphs with text fragments (hubs) pointing to entities (authorities). HITS can be seen as a straightforward extension of the weighted indegree model. Instead of a 1-step propagation from hub to authority nodes, the hub weights are equivalently defined by their weighted out-degree, which leads to a mutually recursive definition of hubs and authorities:

$$Auth(e) = \sum_{d \in \Gamma^-(e)} w(e,d)\ Hub(d),$$

$$Hub(d) = \sum_{e \in \Gamma^+(d)} w(d,e)\ Auth(e).$$

In order to incorporate the query dependent weighting of the documents, we use here weighting model that includes the query as an additional (entity) vertex in the graph (Figure 3(a)). Since we have a bipartite graph with links from document to entities only, documents will only have hub scores and entities will get authority scores only.

We have to make two remarks about the HITS algorithm as it is applied here. First, including the query node as an additional entity introduces an indirect "random jump" possibility. Since all document nodes are connected to the query, each of them is reachable from the others. Secondly, notice that the HITS algorithms performs a normalization step on its hub and authority values after each iteration not mentioned in the above given definition. Therefore, the edge transition weights do not have to be probabilities. We can for instance set them uniformly to 1 with the implication that a document does not divide its importance among its contained entities but propagates its full weight to all of them, vice versa for the propagation from entities to documents.

The HITS algorithm was adapted in other ways to incorporate a prior vertex weighting (among others by Bharat and Henzinger [3]). Since we use HITS mainly for comparison, we stick here to the more basic propagation model but compute it as mentioned on the entity graph extended by the additional query node.[1]

4. EXPERIMENTAL STUDY ON EXPERT FINDING

We used TREC's expert finding task to evaluate our entity ranking approach, since it is a typical entity ranking task and the required evaluation data (collection, topics, and assessments) are available. Although we cannot claim that all observations made for expert finding will be valid also in other entity ranking domains, we expect to find similar results there.

The corpus used in TREC for expert finding is the *W3C-corpus* consisting of emails and web documents from the W3C working groups. A list of potential experts for all topics, in this case people participating in the W3C working groups, is provided with the TREC data. We preprocessed the corpus data in order to convert it to proper XML format with the least possible changes to the data itself, and secondly for tagging all occurrences of experts within the corpus. A simple string-matching tagger marked a candidate when it either matched the complete candidate name or her/his email-address. We disregarded abbreviations of the names since they could also mislead to different persons. All experiments in this section are performed on the email part of W3C corpus, which is the most clean and structured part of the corpus. Using the entire W3C corpus yielded in slightly worse results in general, however the order of compared techniques with respect to their retrieval performance remained the same.

For the initial ranking as well as for the graph generation the PF/Tijah retrieval system [14] was employed. It allows to rank the text content of an arbitrary set of XML nodes, and also provides the full functionality of the XQuery language to specify the output. For this experiment, we generated XQueries that directly output entity containment graphs in *graphml* format given a title-only TREC query.[2] A standard language modeling retrieval model was employed for the initial scoring of text nodes. The generated graphs were later analyzed with a Java graph library, adapted by our own weighted propagation models.

4.1 Testing the Models of Relevance Propagation Models

Before analyzing the influence of single parameters, Table 1 gives an overview on the performance of all discussed relevance propagation models. Mean average precision (MAP) is reported in all experiments since it was used for the TREC evaluation as well. The results are based solely on the expert ranking, not taking into account the supporting documents also used in the TREC evaluation. Wherever appropriate, we also show how the precision on the top of the ranked list (P10) is influenced by the different parameter settings. Here, and in all other cases where not stated otherwise, we used the 1500 top ranked documents for building the entity containment graph and included only expert entities; domain specific edge weighting schemes are not considered. In case of the random walk, we set the random jump probability to a small value, $\lambda_0 = 0.1$, in order to emphasize the walk on the graph structure. Such a setting is also typical for the use of random walks for web retrieval.

The table reveals the three main results of the evaluation:

[1]The basic model was even showing better results in initial experiments on expert ranking.

[2]Title + description queries were tested as well without any improvements in the results.

Table 1: Performance of the relevance propagation models

Model	*unweighted*	*weighted*
MAX		0.352
IDG	0.342	0.371
HITS	0.343	0.376
PRW	0.340	0.386

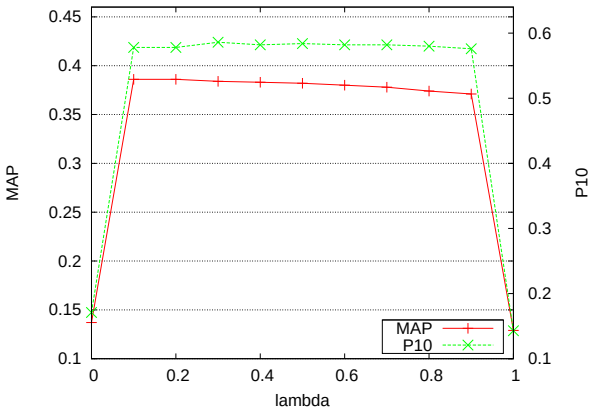


Figure 5: Influence of interpolation factor λ in HITS and PRW

(1) By adding more graph-based features to the relevance propagation model, the results improve. From the baseline model (MAX), followed by the weighted indegree (IDG) towards the probabilistic random walk (PRW) we can see a slight but constant improvement on each step. (2) Even pure unweighted graph structure provides useful features for entity ranking – otherwise the MAP values for the unweighted models would be by far lower – but incorporating the probabilities of the initial ranking clearly improves the results. (3) The non-recursive indegree performs reasonably well compared to the more sophisticated relevance propagation models like HITS or the random walk. Differences are still visible, but in many applications the fast to compute weighted indegree might be sufficient.

Setting of the Interpolation Factor. The random walk requires to set the probabilities λ_i for taking a step to the different neighborhoods of a node. Since we do not have direct edges between documents and entities here, the parameter space can be reduced to one variable: $\lambda_0 = \lambda, \lambda_1 = 1 - \lambda$. Figure 5 shows how different settings of λ influence the retrieval performance of the random walk. The MAP curve suggests to set the interpolation factor close to 0, however, the relative flat gradient shows that the relevance propagation is rather insensitive to this setting. Not only the mean average precision, also the precision on top of the ranked list (P10) remains unchanged by different settings of λ.

Limiting the Number of Iterations. The recursive propagation models, HITS and the random walk, are computed iteratively until the overall change of the probabilities in the graph between two consecutive iterations is smaller than ϵ, usually given by the numerical precision of a system. After roughly 50 iterations the values converge in our experiments. Since we observed already good performance for the non-iterative indegree model, the question arises, whether less iterations would be enough to achieve the wanted probability propagation within the graph. In the random walk model, a limitation to n iterations can be interpreted as an n-step random walk. Figure 6 shows the retrieval performance for such n-step walks. Obviously, the first few steps yield the highest improvements, and the precision gain for rest of the iterations remains minimal.

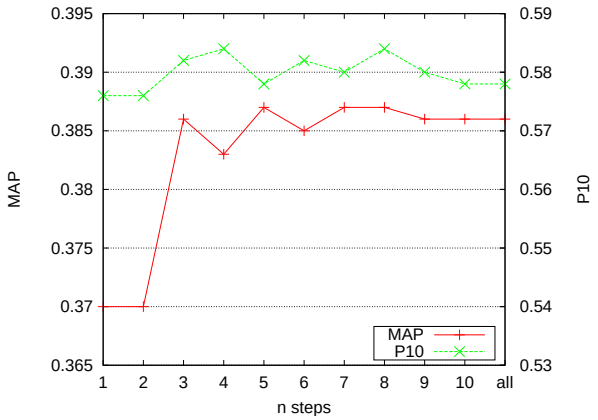


Figure 6: Retrieval performance for n-step random walks

Furthermore, we want to remark here, that the weighted versions of the propagation models converge in general faster than their unweighted counterparts relying on the pure graph structure, which makes them computational less cost intensive.

4.2 Analyzing the Graph Modeling

Number of Included Text Fragments. The first parameter to investigate here is the number of top ranked text fragments that are taken into account when building the entity containment graph. Notice that not all of these top ranked documents will become a vertex in the graph, but only those containing at least one expert node. In fact, this filters out a large part of the retrieved document list. E.g. from 1500 top ranked documents on average 650 contain expert entities with a high variation among queries.

Figure 7 shows the influence of the number of top ranked text fragments on the retrieval performance. Remarkably here is the difference between the weighted (wPRW) and unweighted version (PRW) of the random walk. The weighted random walk gains higher MAP values and does not loose precision on top of the ranking. Apparently it does not suffer from the weakened topical focus. However, the precision at 10 retrieved entities degrades rather soon for the unweighted variant relying on pure graph statistics. Also the mean average precision deteriorates in the unweighted walk when more than 1500 top ranked documents are considered for

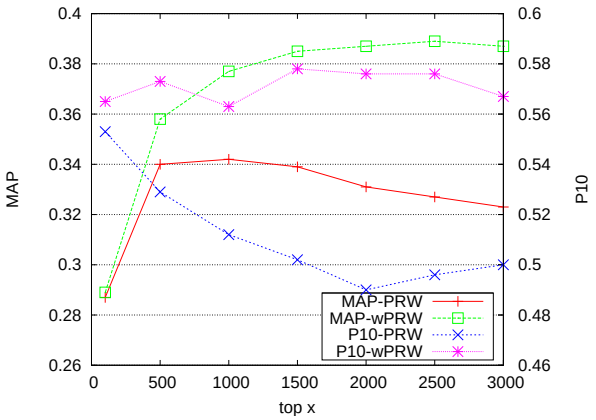


Figure 7: Influence of the number of top x documents taken from the initial retrieval

Table 2: Including Person Entities

Model	*unweighted*	*weighted*
HITS	0.343	0.375
PRW	0.342	0.388

the graph building.

Including Person Entities. In order to achieve a denser graph we tried to include further entity types apart from experts. One straightforward idea was to tag all other person occurrences throughout the whole corpus from those persons known as an author of at least one mail in the email sub-collection. Similar to the experts we used the full name or email-address for identification. Experts who also wrote emails were not tagged twice. The additional person entities increased the graph-sizes by far, since also documents containing a person but no experts were included in the graph network as well.

Since the indegree of experts nodes will not be influenced by the additional entity vertices, we take a look only on the recursive propagation models. Table 2 provides a result overview as for the expert-only graphs before (Table 1). Unfortunately, the overall changes remain minimal here for both tested propagation models. Initial experiments on the full W3C-corpus have shown higher improvements for the inclusion of person entities, but cannot be approved on the email corpus.

5. RELATED WORK

Graph-based ranking methods are first of all known from web retrieval. Among them, Pagerank [22] and HITS [15] are probably the most popular, and their usage is widely studied in the field of hypertext retrieval. Similar to our work here, more recent graph-based approaches try to incorporate as much information into the graph as possible. Pagerank can be regarded in general as a Markov process, or a random walk on the web graph [12]. Several attempts have been made in the last years to make this walk query and content dependent. The *intelligent surfer* [23] walks to linked pages biased by their relevance to the query. The surfer model proposed by Shakery and Zhai uses a similar, but bi-directional walk considering both out-links and in-links of a node [24].

Graph-based ranking methods often find applications beyond the bounds of hyperlinked corpora. They were recently adapted for spam detection [6] and blog search [16]. Kurland and Lee experimented with structural re-ranking for ad-hoc retrieval, first using Pagerank [17] and later HITS [18] in bipartite graphs of documents and topical clusters. Erkan and Radev use implicit links between similar sentences to compute their centrality for text summarization [9]. More close to our work, Zhang et al. studied query-independent link analysis in post-reply networks for expert detection comparing Pagerank and HITS centralities [27]. Another new study by Agarwal et al. generalizes completely from the application and tries to learn the best edge weighting function for a Markov walk from relevance assessments [1]. Their notion of entities here is even broader than ours and also includes the documents itself.

6. CONCLUSIONS

Formulating the task of entity ranking as a graph-based relevance propagation has shown to be a fruitful theoretical model. It does not only motivate and justify the suggested propagation models, but we could also show that exploiting more and more graph features improved the expert ranking in our empirical study. Since our experiments only study the problem of expert finding, we cannot claim that all suggested techniques will work similarly in other domains of entity ranking, but the general propagation model will be useful there as well.

The main findings of the experimental study on expert finding can be summarized as follows. The pure unweighted graph structure of the entity containment graph provides useful additional hints for the expert ranking, but cannot come up with high quality rankings on its own. Similarly, our baseline ranking approach that relies solely on the initial ranking of documents remains beaten by all graph-based relevance propagation models. It is thus necessary to combine both the structural features of the graph as well as the initial document ranking to yield the best retrieval performance. From the three relevance propagation models that make use of graph features, the simplest one – the weighted indegree model – is already showing a rather good performance. The more sophisticated random walk seems slightly superior, but comes with the disadvantage of a more costly computational model and further parameters that require to be set appropriately. However, even a very limited number of random steps on the graph yields in the best performing model found here.

We currently see two interesting directions of future work. Since the paper claims to provide an approach for entity ranking in general, but carries out its experiments on expert finding only, it will be necessary to study other entity ranking tasks. Especially working with multiple types of entities, as with candidate experts and other persons, needs more exploration to deliver improving results. Another challenging task linked to expert ranking in general is the problem of

finding useful supporting text fragments for the given ranking. Although we explained in the introduction that the added value of entity ranking compared to passage or XML retrieval is that it directly returns the extracted ranked list of entities, it is important to notice that such a result list is in many cases only useful in combination with links to supporting sentences or passages. Our graph-based propagation models might be useful here as well, since they also rank the text fragments with respect to the included entities.

7. REFERENCES

[1] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 14–23, New York, NY, USA, 2006. ACM Press.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In Efthimiadis et al. [8], pages 43–50.

[3] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR*, pages 104–111. ACM, 1998.

[4] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *CIKM*, pages 528–531. ACM, 2003.

[5] H. Chen, H. Shen, J. Xiong, S. Tan, and X. Cheng. Social network structure behind the mailing lists: Ict-iiis at trec 2006 expert finding track. In *Proceedings of the 15th Text REtrieval Conference Proceedings (TREC)*, 2006.

[6] P. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In Herzog et al. [13], pages 373–380.

[7] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec-2005 enterprise track. In *Proceedings The 14th Text REtrieval Conference (TREC 2005)*, 2005.

[8] E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors. *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*. ACM, 2006.

[9] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)*, 22:457–479, 2004.

[10] R. Grishman and B. Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[11] D. Hawking. Challenges in enterprise search. In K. Schewe and H. E. Williams, editors, *ADC*, volume 27 of *CRPIT*, pages 15–24. Australian Computer Society, 2004.

[12] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. Measuring index quality using random walks on the web. *Computer Networks*, 31(11-16):1291–1303, 1999.

[13] O. Herzog, H. Schek, N. Fuhr, A. Chowdhury, and W. Teiken, editors. *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*. ACM, 2005.

[14] D. Hiemstra, H. Rode, R. van Os, and J. Flokstra. Pftijah: text search in an xml database system. In *Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR), Seattle, WA, USA*, pages 12–17. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2006.

[15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, pages 668–677, 1998.

[16] A. Kritikopoulos, M. Sideri, and I. Varlamis. Blogrank: ranking weblogs based on connectivity and similarity features. In *AAA-IDEA '06: Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications*, page 8, New York, NY, USA, 2006. ACM Press.

[17] O. Kurland and L. Lee. Pagerank without hyperlinks: structural re-ranking using links induced by language models. In R. A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, and J. Tait, editors, *SIGIR*, pages 306–313. ACM, 2005.

[18] O. Kurland and L. Lee. Respect my authority!: Hits without hyperlinks, utilizing cluster-based language models. In Efthimiadis et al. [8], pages 83–90.

[19] J. Lin and B. Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 116–123, New York, NY, USA, 2003. ACM Press.

[20] X. Liu, B. W. Croft, and M. B. Koll. Finding experts in community-based question-answering services. In Herzog et al. [13], pages 315–316.

[21] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In Yu et al. [26], pages 387–396.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library, 1999.

[23] M. Richardson and P. Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *NIPS*, pages 1441–1448. MIT Press, 2001.

[24] A. Shakery and C. Zhai. A probabilistic relevance propagation model for hypertext retrieval. In Yu et al. [26], pages 550–558.

[25] E. M. Voorhees and H. T. Dang. Overview of the trec 2005 question answering track. In *The Fourteenth Text REtrieval Conference (TREC) Proceedings*, 2005.

[26] P. S. Yu, V. J. Tsotras, E. A. Fox, and B. Liu, editors. *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*. ACM, 2006.

[27] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 221–230, New York, NY, USA, 2007. ACM Press.