IT Company Atlas Upper Franconia

A Practical Application of Expert Search Techniques

Daniel Blank
Chair of Media Informatics
University of Bamberg
daniel.blank@uni-bamberg.de

Sebastian Boosz
Chair of Media Informatics
University of Bamberg
sebastian.boosz@uni-bamberg.de

Andreas Henrich
Chair of Media Informatics
University of Bamberg
andreas.henrich@uni-bamberg.de

ABSTRACT

Identifying a competent service provider or contractor is not an easy task. This is especially true in the IT sector with its short-lived trends and products. In this context an "IT Company Atlas" giving substantiated information about the IT companies in a certain region seems to be a promising idea. However, a B2B directory administered manually for this purpose is expensive to establish and to maintain. An alternative approach discussed in this paper is to apply expert search techniques and provide a search engine searching for companies instead of documents.

We propose a system searching for companies with expertise in a given field sketched by a keyword query. The system exploits the websites of the companies and covers all aspects: determining and representing the expertise of the companies, query processing and retrieval models, as well as query formulation and result presentation. In addition to the theoretical background and a system description we present experimental results comparing the effectiveness of different retrieval models adopted from expert search in the scenario we call "company search".

CCS Concepts

•**Information systems** → **Expert search;** *Language models; Combination; fusion and federated search; Enterprise search;*

Keywords

Company search; Expertise retrieval; Expert search; Domain specific search solutions; IR systems

1. MOTIVATION

The idea for the presented search engine came up when the association of IT companies in Upper Franconia tried to figure out the opportunities to set up a register of competencies for their members and associated companies. The basic idea was: Whenever a company with a demand for IT solutions is looking for a potential service provider or contractor the "register of competencies" will easily expose good matches.

The first idea was to conduct a survey among the companies and derive a brochure or website. However, the foreseeable problems with low return rates as well as the expected update and maintenance costs led to a broader consideration of potential approaches. The next thing that came into mind was "a special search engine". A closer look at commercial and freely available candidates made clear that these approaches were not convincing for the intended target group. The main reason based on small experiments was the inappropriate result presentation. The results consisted of documents and in many situations it was rather unclear what qualified the documents for top ranks in the result list and—even more important—which company was associated with the document.

Consequently, the next step was the design of a domain specific search engine optimized for the "company search task". In fact, company search can be considered as a special case of expertise retrieval—a topic intensely discussed in literature [3]. In this context the contribution of the paper at hand is twofold: (1) We present a domain-specific retrieval system with peculiar aspects regarding the retrieval model and especially the result presentation. (2) Experimental evaluations are presented comparing the applicability of different models known from the field of expert search in the company search scenario[1].

The remainder of the paper is organized as follows: In section 2 we present a rough overview of the proposed system to set the ground for the rest of the paper. Related work is addressed in section 3. Thereafter we discuss the four components identified to make up an expertise retrieval system according to Balog et al. [3, p. 145]: *modeling and retrieval*

[1]A preliminary version of this paper has been presented on a local workshop [4]. In contrast to [4] the current paper presents an enhanced version of the company search incorporating an optimized user interface as well as multiple retrieval models. Furthermore, experimental results are presented, which were missing in the preliminary version.

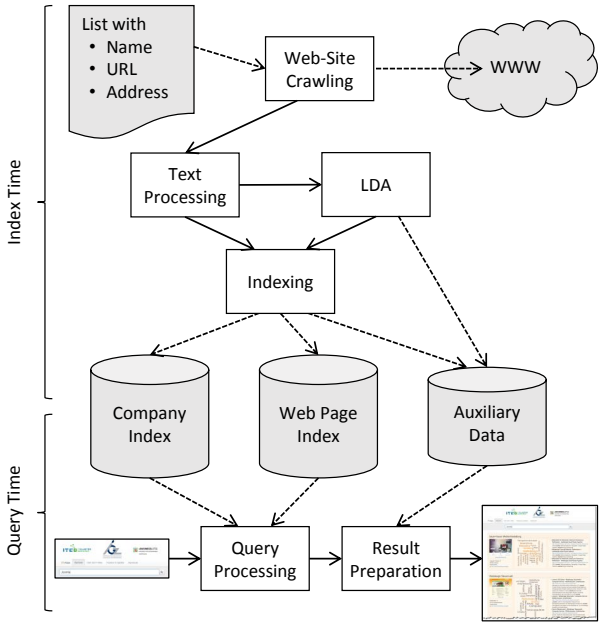

Figure 1: System overview

(section 4), *data acquisition* (section 5), *preprocessing and indexing* (section 6), as well as *interaction design* (section 7). In section 8 we evaluate our approach comparing the performance of a topic oriented version with document-based voting models. Finally section 9 concludes the paper.

2. SYSTEM OVERVIEW

The basic purpose of the system is that a manually defined set of companies should be searchable in the system. These are the members and associated companies of the IT-Cluster Upper Franconia. An assumption is that all these companies have a more or less expressive website. Hence, the starting point for the system is a list of companies, consisting of the name of the company, the URL of the corresponding website and the office address (represented in the upper left corner of Fig. 1).

The URLs are used to crawl the websites of the companies. Roughly spoken, each company c can be either represented by the set of the content blocks of its web pages or alternatively as the concatenation of these blocks, called d_c. Of course, some text processing is necessary here for noise elimination, tokenization, lemmatization, and so forth.

Since the corpus (consisting of about 700 companies at present) is rather small and queries might be specific (for example a search for "joomla") we incorporate topic models using Latent Dirichlet Allocation (LDA) [5] to boost companies with a broader background in the topic of the query. Using the terms and LDA-based boost-factors two indexes are build: In the first index the companies are indexed based on the pseudo documents d_c. In the second index the single web pages are indexed. Besides its potential use for the company ranking, we also want to use this web page index to deliver in the result the best landing pages for the query within the websites of the ranked companies. Finally some auxiliary data (for example the topic models generated via LDA) is stored since it is needed during query processing and result presentation.

When a query is issued the query is processed on the web page index. In case of the LDA-based topic oriented version the company index is queried first and the web page index is only used for finding the best landing pages. A result page is generated which represents companies as first class citizens. For each company a home page thumbnail, a characterization of its competencies and its relationship to the query, as well as up to three query related landing pages are presented. All these aspects will be considered in more detail in the remainder of this paper, but beforehand we want to address related work.

3. RELATED WORK

To our best knowledge, there is no directly related work on company search. The two most related areas are expertise retrieval [3] and entity search [14]. Many lessons can be learned from these areas. Nevertheless, there are some peculiarities with our company search scenario. In expert finding scenarios the identification of the experts is often a hard problem (see the expert finding task in the TREC 2007 enterprise track as an example [1]). Another aspect is the ambiguity of names or the vague relationship between persons and documents. On the other hand, representing experts by pseudo documents is also an established approach in expert search [2] and an elaborate result presentation is important here as well.

As already mentioned we use pseudo documents to represent companies in our approach similar to approaches in expert search [2]. We combine this approach with topic models to consider the broader expertise of companies with respect to the query. This is similar to the author-topic model introduced by Rosen-Zvi et al. [11], a generative model for documents that extends LDA to include authorship information.

Another approach considered in expert search is voting models [10]. Here no pseudo documents are used. Instead relevant documents for the given query are retrieved and these documents are allowed to vote for the associated authors. Similar to this approach we can search for single web pages and then the retrieved web pages can vote for their companies. This approach is also implemented for the company search and it will be used for comparison in section 8.

4. MODELING AND RETRIEVAL

Besides crawling and preprocessing which are discussed in sections 5 and 6, it is essential to find an appropriate retrieval model. In order to dig into the details of modeling and retrieval aspects and to take a more formal look we use the following definitions:

- $q = \{w_1, w_2, \ldots w_n\}$ is the query submitted by the user (set of words)
- $C = \{c_1, c_2, \ldots c_m\}$ is the set of companies

- d_c is the concatenation of the documents representing company c
- f_{w,d_c} is the number of times word w appears in d_c
- cf_w is the number of companies $c \in C$ for which d_c contains w
- λ and μ are design parameters

Following [7] we use a candidate generation model and try to rank the companies by $P(c|q)$, the likelihood of company c to be competent for query q. As usual, by invoking Bayes' theorem, this probability can be refactored as follows [3]:

$$P(c|q) = \frac{P(q|c)P(c)}{P(q)} \stackrel{\text{rank}}{=} P(q|c)P(c) \approx P(q|d_c)P(c)$$

Currently, we do not distinguish company priors $P(c)$. However, it turned out that this will be an interesting point for future optimizations highly correlated with aspects of document length normalization for the pseudo documents d_c. For test purposes we inserted big national and international IT companies in the list. In our current implementation these companies did not make it to the absolute top ranks even for queries simply consisting of a registered product name of the company. Instead, small service providers which are specialized in support for this product were ranked higher.

Another important aspect which became obvious in first experiments was the well known vocabulary mismatch. For example with the query "joomla" the ranking did not consider the broader competences of the companies in the field of web applications or content management systems. We thus decided to use LDA [5, 6] to address this problem. An independence assumption to calculate the probabilities wordwise by $P(q|d_c) = \prod_{w \in q} P(w|d_c)$ and a combination of the word-based perspective and a topic-based one leads to:

$$P(w|d_c) = \lambda \left(\frac{f_{w,d_c} + \mu \frac{cf_w}{|C|}}{|d_c| + \mu} \right) + (1 - \lambda) P_{lda}(w|d_c)$$

$P_{lda}(w|d_c)$ stands for the probability that a word w is generated by a topic which is generated by d_c (see [6, 12] for more details). To simplify things further, we employed an idea presented in [13]. Here Lucene[2] payloads are used to boost terms via LDA. The payload $lda(w, d_c)$ assigned to word w is determined as the weight of w according to the topic distribution of d_c. This means that $lda(w, d_c)$ is high when w fits well with the broader topics dealt with in d_c. Combining this boost factor with term frequency (tf) and inverse document frequency (idf) information we yield the following conceptual scoring function:

$$score(c, q) = \sum_{w \in q} \left(f(w, d_c) \cdot idf(w) \cdot lda(w, d_c) \right)$$

Of course, this is only a first pragmatic starting point and the above considerations point out various interesting aspects for future optimizations. For evaluating the performance of the LDA-based company ranking approach, we compare its effectiveness in section 8 with four instances of a document-based voting model. As retrieval models for the document ranking we apply Lucene's default TF-IDF model (*DocDef*), BM25 model (*DocBM25*), Jelinek-Mercer (*DocJM*)[3] and Dirichlet (*DocDir*) language models.

For the voting models CombMAX is chosen for score aggregation with the maximum score of a document determining the place of a company in the final ranking. The reason for choosing CombMAX is that length normalization becomes less important in this case and almost always normalization leads to a decrease in efficiency [10]. Normalization is a more important aspect when applying the voting model in combination with aggregation schemes such as CombSUM and expCombSUM. Furthermore, "the successful application of normalization also depends on the testbed and on how candidates and documents are associated" [3, p. 184]. We will deal with the important issue of normalization and thus the analysis of alternative combination schemes such as CombSUM and expCombSUM in future work. For the time being, we stick with CombMAX which offers good performance in prior experimental work [3, 10].

5. DATA ACQUISITION

As a prerequisite documents representing companies have to be obtained first. For crawling company websites we chose to employ crawler4j[4], a lightweight Java web crawler.

The crawling of each company is an individual process which allows us to crawl multiple companies at once. We start with the company's home URL as a seed and use a truncated form of that URL as a pattern to discard all links to external domains found during the process. For our first investigation, we crawled a maximum amount of 2000 documents per company in a breadth first manner where each document is a web page. We plan to leverage additional document types, such as PDF, in the future.

For each page the corresponding company, page title and full URL are stored in a database. This information is reused later when creating the web page index. To obtain d_c (the pseudo document describing company c) the contents of all crawled pages of the company are concatenated. To reduce noise, we apply the Boilerpipe library[5] [9] to all documents in order to extract the main textual content from those pages first. This step aims at eliminating those page elements which do not contribute to the actual content of a page and are repeated very often: navigation menus, footer information, etc.

6. PREPROCESSING AND INDEXING

Early experiments have shown that data quality plays a seminal role for the quality of a topic model learned. That is why we utilize a customized Lucene Analyzer before applying the LDA to d_c or indexing the company documents. The ana-

[2] `http://lucene.apache.org/` (all URLs visited: 12/2/15)

[3] DocJM is the only Lucene model which relies on an external parameter λ. We set λ to 0.1 since the API documentation states that an optimal value should be around 0.1 in case of title queries which distinguish themselves from long queries.

[4] `https://github.com/yasserg/crawler4j`

[5] `https://code.google.com/p/boilerpipe`

lyzer filters German stop words, applies a Porter Stemmer for the German language and uses a series of regular expressions to remove or modify tokens. As an example, digit-only tokens are removed, while tokens of the form *word1:word2* are split into two tokens, *word1* and *word2*. Consistently, incoming user queries are processed by the same analyzer.

After the analyzing step, an LDA topic model of all company representations d_c is created, utilizing the jgibbsLDA[6] implementation. The resulting model enables us to directly access the distribution of topics for each company as well as the word probability distributions within topics. Therefore the payload function $lda(w|d_c)$ for each word w in d_c can be computed immediately. Another representation of d_c is created, where each term is enriched with its determined LDA payload. The generated LDA model is reused for result preparation (see section 7).

In addition to the document index for the companies' web pages, a company index is created from all pseudo documents d_c enriched with payloads. When executing a query it is examined by the index searcher and consequently determines the ranks of the companies in the result set according to the chosen retrieval model.

With the creation of a company index representing companies and their competencies, a web page index for the companies as well as the overall topic model, all steps necessary to enable searching are completed.

7. INTERACTION DESIGN

As usual, the query is given as a simple keyword query. In the future more sophisticated variants are conceivable, for example allowing for geographic filter constraints. Nevertheless, the simple and familiar keyword solution has its advantages and the use of geographic filter constraints is debatable as long as only companies located in Upper Franconia are listed, anyway.

With respect to the result presentation the situation is more demanding. Discussions with potential users disclosed the following requirements: (1) Companies are the main objects of interest. (2) Address information, a first overview, and a visual clue would be nice. (3) The general company profile as well as the relationship to the query should become obvious. (4) Entry points (landing pages) for the query within the website of the company are desired.

The result page depicted in Fig. 2 directly implements these requirements. Companies are ranked with respect to the chosen retrieval model described in section 4. For each company in the result list a row with the name and three information blocks is shown. The company name is directly taken from the input data as well as the address information (Fig. 1 upper left corner). A screenshot of the homepage (captured with Selenium[7]) and a prepared link to OpenStreetMap[8] complete the left overview block for each company. In the middle block a word cloud is given. Here the size of the terms represents their importance for the company profile (based on $tf \cdot idf$ information). The color represents the relationship of the terms to the query. Orange represents a strong relationship. The relationship is calculated based on a company's prevalent LDA topics. Currently, we consider the five top terms of the five topics with the highest correlation to the query. At most thirty terms are shown in the word cloud taking terms important for the company profile and important for the relationship to the query in a round robin procedure. Finally, the right block consists of up to three most relevant landing pages within the company website represented by their title, the URL, and a query-dependent snippet.

[6] `http://jgibblda.sourceforge.net/`
[7] `http://www.seleniumhq.org/`
[8] `http://www.openstreetmap.org/`

8. EVALUATION

For evaluating the performance of our approach, we decided to compare the effectiveness of the LDA-based search with 100 topics to a document-based voting model as described in section 4.

In order to establish a ground truth we conducted a pooling approach: Considering the current system's log files, we chose the ten most frequent queries (omitting company names and the term *computer* for being inexpressive in an IT domain). For each query we collected the top ten hits of all compared approaches removing duplicates. Four independent raters with an IT background were then chosen to score the hits for each query using a three-valued scale: 0 (not relevant), 1 (partially relevant) and 2 (highly relevant).

A company's relevance related to a query is constituted by the average rating of the evaluators. Table 1 gives an overview of all query terms. n_c denotes the number of companies in the pool. n_r is the number of relevant companies (a company is deemed to be relevant if its average rating is ≥ 1.5). σ_m is the mean of the standard deviations in the raters' relevance decisions. Higher values indicate that rating the companies for the particular query has been more controversial than for other queries. As a measure for the difficulty of a query n_r can also be taken into account: while for *webdesign* nearly all (26 from 29) companies found are relevant, *test* (2 from 26) is a much harder and unspecific query: while the term *test* occurs frequently on company websites, only a few companies in our sample have expertise in conducting professional software/security tests or developing tools for testing.

Query	Term(s)	n_c	n_r	σ_m
1	open source	23	9	0.3447
2	cloud	16	12	0.2068
3	java	20	9	0.4253
4	e-procurement	14	5	0.5030
5	joomla	17	12	0.2716
6	programmierung	28	18	0.4830
7	webdesign	29	26	0.2787
8	test	26	2	0.3049
9	erp	24	18	0.1684
10	online marketing	25	18	0.3340
	average	22.2	11	0.3019

Table 1: Frequent queries which are used for the experiments

Figure 2: Result page for the query "joomla"

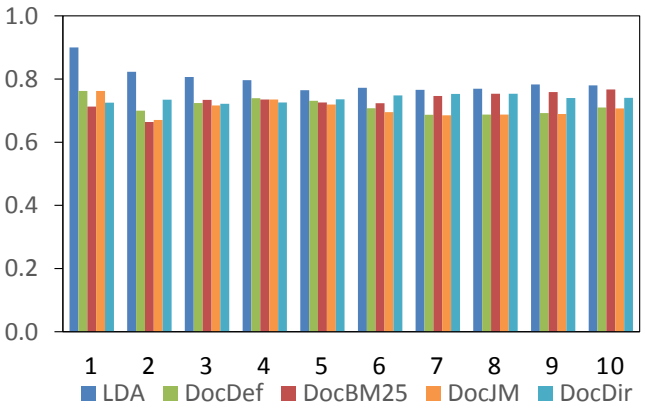

Figure 3: Aggregated NDCG values (ordinate) for each rank (abscissa)

For analyzing the performance of the approaches we use the *normalized discounted cumulative gain* (NDCG) [8], well established in the field of IR. This measure emphasizes the importance of relevant hits appearing early in the result ranking. Figure 3 displays NDCG values at ranks 1 to 10. In addition to the LDA-based approach, the four document-based techniques are shown. It can be observed that the LDA-based approach outperforms the competing approaches for all ranks, indicating a higher retrieval performance at ranks 1 to 4 (however, without statistical significance). Ignoring LDA, none of the document-based approaches can outperform the others when all ranks are considered.

Method	NDCG@1	NDCG@3	NDCG@10
LDA	**.9000**	**.8065**	**.7801**
DocDef	.7625	.7240	.7098
DocBM25	.7125	.7341	.7671
DocJM	.7625	.7164	.7069
DocDir	.7250	.7218	.7405

Table 2: NDCG values at ranks 1, 3 and 10

At rank 1 the effectiveness of LDA is 18% higher than the best document-based models as shown in table 2. When moving away from the aggregated view and instead focusing on the individual queries, figure 4 shows the differences of NDCG values between the LDA model and the best performing document-based model for ranks 1, 3 and 10 for the single queries. The differences are sorted in descending order. On the x-axis the corresponding query identifier from table 1 is provided. The diagram on the left shows that the LDA-based version outperforms DocDef at rank 1 only in 5 out of 10 queries. DocDef's lower aggregated NDCG@1 score in comparison to LDA is highly influenced by query 7 ("webdesign"). Accordingly, diagrams for ranks 3 and 10 comparing LDA and DocBM25 show more heterogeneous re-

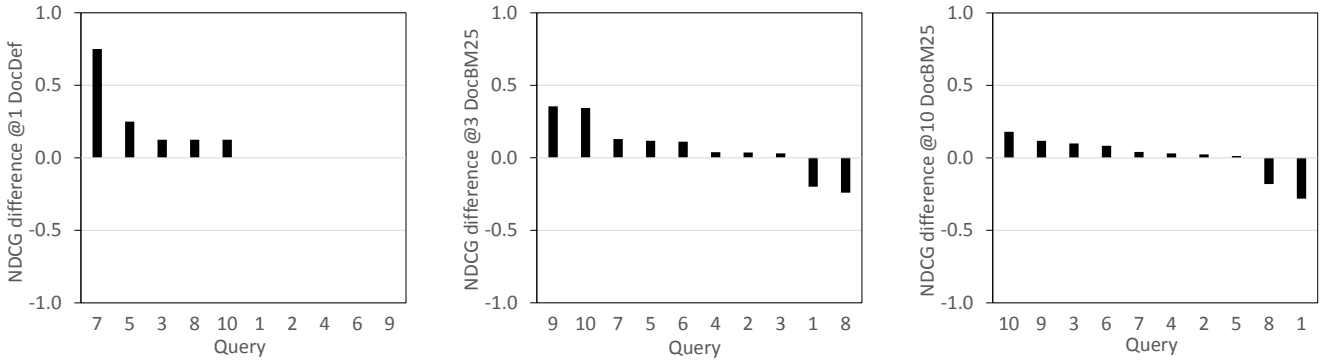

Figure 4: Differences of NDCG values at ranks 1, 3 and 10

sults and for 2 out of 10 queries ("open source" and "test") DocBM25 even performs better than LDA. The diagrams also show that the differences between LDA and DocBM25 are diminishing when higher ranks are considered.

9. CONCLUSION

In this paper we delved into the company search problem. We presented a solution based on pseudo document ranking, the use of LDA to incorporate topical relevance, and a suitable result presentation. The evaluation of company ranking techniques focused on a comparison between the LDA-based approach and instances of a document-based voting model. Although the preliminary analysis with only 10 queries indicates a promising effectiveness, future work will have to address the issue of company ranking in a more comprehensive way. Another important aspect in this regard is the appropriate ranking of companies of different sizes and thus document normalization which was not considered here. Other remaining tasks are the visualization of the company profiles in the result page and a reasonable modeling and presentation of topics (number of topics in LDA and also alternative approaches). In addition to NDCG-based comparisons of different retrieval models, user studies investigating the user interface are planned. The current prototype is available on the project web page[9].

10. REFERENCES

[1] P. Bailey, A. P. De Vries, N. Craswell, and I. Soboroff. Overview of the TREC-2007 enterprise track. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*. NIST Special Publication: SP 500-274, 2007.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 43–50, New York, NY, USA, 2006. ACM.

[3] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Found. Trends Inf. Retr.*, 6(2–3):127–256, Feb. 2012.

[9] `www.uni-bamberg.de/minf/forschung/firmensuche/`

[4] D. Blank, S. Boosz, and A. Henrich. Company search — when documents are only second class citizens. In R. Bergmann, S. Görg, and G. Müller, editors, *Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB.*, Trier, Germany, 2015. published at http://ceur-ws.org.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[6] W. B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Pearson Education, 2009.

[7] H. Fang and C. Zhai. Probabilistic models for expert finding. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 418–430, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[9] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 441–450, New York, NY, USA, 2010. ACM.

[10] C. Macdonald and I. Ounis. Searching for expertise: Experiments with the voting model. *Comput. J.*, 52(7):729–748, 2009.

[11] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[12] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *Proc. of the 29th international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

[13] M. Zhang and C. Luo. A new ranking method based on latent dirichlet allocation. *Journal of Computational Information Systems*, 8(24):10141–10148, 2012.

[14] M. Zhou. *Entity-centric search: querying by entities and for entities*. Dissertation, University of Illinois at Urbana-Champaign, dec 2014.