Company Search — When Documents are only Second Class Citizens

Daniel Blank, Sebastian Boosz, and Andreas Henrich

University of Bamberg, D-96047 Bamberg, Germany,
firstname.lastname@uni-bamberg.de,
WWW home page: http://www.uni-bamberg.de/minf/team

Abstract. Usually retrieval systems search for documents relevant to a certain query or—more general—information need. However, in some situations the user is not interested in documents but other types of entities. In the paper at hand, we will propose a system searching for companies with expertise in a given field sketched by a keyword query. The system covers all aspects: determining and representing the expertise of the companies, query processing and retrieval models, as well as query formulation and result presentation.

Keywords: Domain specific search solutions, expertise retrieval

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 provider of services or solutions 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 update and maintenance problems led to a broader consideration of potential approaches. The next thing that came into mind was "search engines". 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 special purpose search engine optimized for the "company search task". In fact, this scenario can be envisaged

 In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

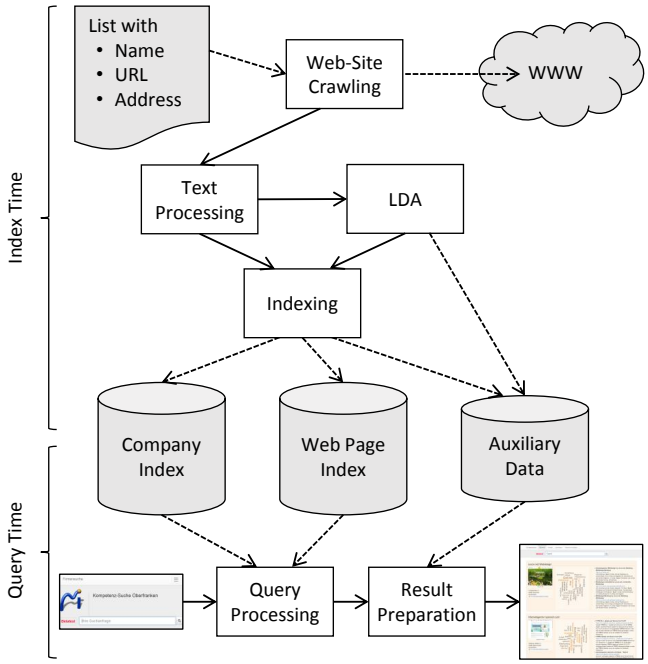

Fig. 1. System overview

as a special case of expertise retrieval—a topic intensely considered in literature [3]. The contribution of the paper at hand in this context is the design and reflection of a system based on state-of-the-art models and components adapted for a real world application scenario. As we will see this requires some peculiar design decisions and user interface aspects.

The remainder of the paper is organized as follows: In the two subsections of section 1 below we present a rough overview of the proposed system and we shortly address related work. 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* (section 2), *data acquisition* (section 3), *preprocessing and indexing* (section 4), as well as *interaction design* (section 5). Finally section 6 concludes the paper.

System Overview The basic assumption 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. A further 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 is represented by the concatenation of the content blocks of its web pages, called d_c. Of course, some text processing is necessary here for noise elimination, tokenizing, stemming, 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 "Typo3") we incorporated topic models (using Latent Dirichlet Allocation currently) 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 because we also want to deliver the best landing pages for the query within the websites of the ranked companies in the result. Finally some auxiliary data (for example the topic models generated via LDA) is stored since it is needed during query processing or result presentation.

When a query is issued the query is processed on the company index and on the web page index. Then 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 shortly address related work.

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 [10]. 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 search 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.

2 Modeling and retrieval

When thinking about the retrieval model for the given scenario on a higher level, a model of the competencies of a company has to be matched with the query representing the user's information need. From the requirements it was defined that a keyword query should be used. With respect to the representation of the company profiles interviews showed that an automatic extraction process is preferable to the manual definition of profiles because of the sheer creation effort and update problems. Due to the addressed domain of IT companies it can be assumed that all companies worth to be found maintain a website depicting their competencies. Of course other sources of evidence could also be addressed—such as newspaper reports, business reports, or mentions of the companies on other

pages in the Internet. These approaches are surely worth consideration in the future. Nevertheless, the concentration and restriction to the company's own website also has the advantage of predictability and clear responsibilities. Put simply, if a company complains that it is not among the best matches for a particular query, we can pass the buck back and encourage them to improve their website—what they should do anyway because of SEO considerations.

To avoid our arguments eventually turning against us, we have to exploit the information on the websites as good as we can. Besides crawling and preprocessing aspects addressed in the following sections 3 and 4, in particular we have to use an appropriate retrieval model. As a first decision we have to choose between a *company-based approach* (each company is represented by a pseudo document used directly for ranking) and a *document-based approach* (the documents are ranked and from this ranking a ranking of the companies is derived). A comparison of these approaches can, for instance, be found in [3], however not showing a clear winner. We plan to test both approaches in the future but we started with the *company-based approach* where each company c is represented by a pseudo document d_c generated as the concatenation of the content blocks of the web pages crawled from the respective website. In the future, we plan to test weighting schemes based on the markup information, the depth of the single pages, and other parameters.

For a more formal look we use the following definitions:

- $q = \{w_1, w_2, \dots w_n\}$ is the query submitted by the user (set of words)
- $C = \{c_1, c_2, \dots c_m\}$ 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 w appears in d_c
- cf_w is the number of companies for which d_c contains w
- λ and μ are design parameters

Following [6] 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]:

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

Currently, we use a constant for the company prior $P(c)$. 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 have specialized in support for this product were ranked higher. Interestingly, this problem is already an issue with expert finding, but an even bigger challenge in company search because of the heterogeneous company sizes.

Another point which became obvious in first experiments was the well known vocabulary mismatch. For example with the query "Typo3" the ranking did

not consider the broader competence of the companies in the topics of web applications or content management systems. As proposed in [4, 5] we decided to use Latent Dirichlet Allocation (LDA) 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 with a topic-based one would then lead 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 [5, 8] for more details). To simplify things further, we employed an idea presented in [9]. Here the Lucene 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 and inverse document frequency information we yield the following scoring function:

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

Of course, this is only a first pragmatic starting point and the above considerations point out various interesting aspects for future comparisons and optimizations.

3 Data acquisition

As a prerequisite documents representing companies have to be obtained first. For crawling company websites we chose to employ crawler4j, a lightweight Java web crawler (`https://github.com/yasserg/crawler4j`).

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 indexes. 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 Boilerpipe [7] (`https://code.google.com/p/boilerpipe`) to all documents in order to extract the main textual content from those pages first. This step aims to eliminate those page elements which do not contribute to the actual content of a page and are repeated very often: navigation menus, footer information, etc.

4 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 analyzer 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 (`http://jgibblda.sourceforge.net/`) implementation. The resulting model is represented in a Java class hierarchy, which 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.

The 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. To be able to show a query's top documents for a given company, we also create an index for the companies' web pages. All crawled web pages are considered and for each page we also preserve the information of the corresponding company. Both types of indexes are based on Lucene (`https://lucene.apache.org/core/`). Prior to indexing we apply the analyzing process described above.

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.

5 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 retrieval model described in section 2.

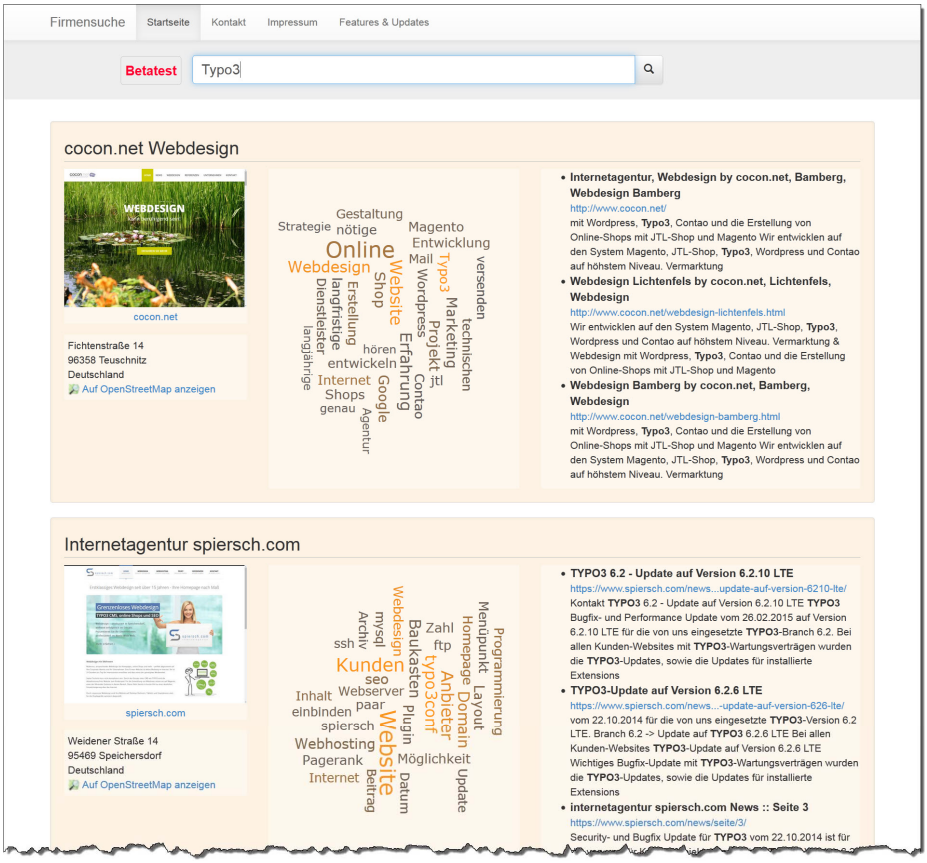

Fig. 2. Result page for the query "Typo3"

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; `http://www.seleniumhq.org/`) and a prepared link to OpenStreetMap complete the left overview block for each company. In the middle block a word cloud is given. Here the size of the terms represents the importance of the terms 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 Conclusion

In this paper we have described the company search problem and presented a solution based on pseudo document ranking, the use of LDA to incorporate topical relevance, and a suitable result presentation. Currently the prototype implementation is tested. It turned out that the effectiveness and the efficiency are promising in preliminary interviews with representatives of local IT companies. Current response times of the system are below two seconds. The most obvious challenges are the appropriate ranking of companies with different sizes, 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). The current prototype is available on the project web page[1].

References

1. Bailey, P., De Vries, A.P., Craswell, N., Soboroff, I.: Overview of the TREC-2007 enterprise track. In: The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings. NIST Special Publication: SP 500-274 (2007)
2. Balog, K., Azzopardi, L., de Rijke, M.: 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, pp. 43–50. ACM, New York, NY, USA (2006). DOI 10.1145/1148170.1148181
3. Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., Si, L.: Expertise retrieval. Found. Trends Inf. Retr. **6**(2–3), 127–256 (2012). DOI 10.1561/1500000024
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research **3**, 993–1022 (2003)
5. Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice. Pearson Education (2009)
6. Fang, H., Zhai, C.: Probabilistic models for expert finding. In: Proceedings of the 29th European Conference on IR Research, ECIR'07, pp. 418–430. Springer-Verlag, Berlin, Heidelberg (2007). URL http://dl.acm.org/citation.cfm?id=1763653.1763703
7. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10, pp. 441–450. ACM, New York, NY, USA (2010). DOI 10.1145/1718487.1718542
8. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 178–185. ACM (2006)
9. Zhang, M., Luo, C.: A new ranking method based on latent dirichlet allocation. Journal of Computational Information Systems **8**(24), 10,141–10,148 (2012)
10. Zhou, M.: Entity-centric search: querying by entities and for entities. Dissertation, University of Illinois at Urbana-Champaign (2014). URL http://hdl.handle.net/2142/72748

[1] `http://www.uni-bamberg.de/minf/forschung/firmensuche`