Unsupervised Approach to Generate Informative Structured Snippets for Job Search Engines

Nikita Spirin
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61820, USA
spirin2@illinois.edu

Karrie Karahalios
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61820, USA
kkarahal@illinois.edu

ABSTRACT

Aiming to improve user experience for a job search engine, in this paper we propose an idea to switch from query-biased snippets used by most web search engines to rich *structured snippets* associated with the main sections of a job posting page, which are more appropriate for job search due to specific user needs and the structure of job pages. We present a very simple yet actionable approach to generate such snippets in an unsupervised way. The advantages of the proposed approach are two-fold: it doesn't require manual annotation and therefore can be easily deployed to many languages, which is a desirable property for a job search engine operating internationally; it fuses naturally with the trend towards Mobile Web where the content needs to be optimized for small screen devices and informativeness.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Abstracting methods, Linguistic processing

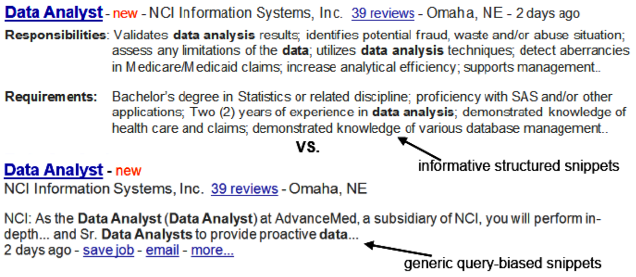
Keywords

search snippet, information extraction, summarization

1. INTRODUCTION

Search is one of the most important services on the Web. However, due to complexities of natural language, existing systems cannot guarantee perfect retrieval results. To address this issue, search engines "cooperate" with the users via intelligent search interfaces: on a Search Engine Result Page (SERP) each link is accompanied with a corresponding *snippet* that helps users judge whether a page is relevant for a query. The de facto method to generate snippets is based on extraction of sentences containing the query terms [5]. This method is used across most search verticals, which focus on specific segments of online content. However, such an egalitarian approach might be suboptimal and some search verticals might benefit from a dedicated solution. One such example is job search.

In this paper we propose using *structured snippets*, associated with the specific sections of a job posting (responsibilities and requirements), rather than the generic query-biased snippets. To contrast, we show the examples of both in Figure 1. By summarizing and bringing the most important information directly to the SERP, we help users complete the search task faster and, therefore, improve an overall user experience (UX). This work complements the research on good abandonment, a phenomenon studied in the context of web search, when a satisfied user leaves a SERP without making any clicks since the answer is already present [1, 3]. In our case, however, snippets help users in making the right clicks rather than not making them at all.



Data Analyst - new - NCI Information Systems, Inc. 39 reviews - Omaha, NE - 2 days ago

Responsibilities: Validates **data analysis** results; identifies potential fraud, waste and/or abuse situation; assess any limitations of the **data**; utilizes **data analysis** techniques; detect aberrancies in Medicare/Medicaid claims; increase analytical efficiency; supports management..

Requirements: Bachelor's degree in Statistics or related discipline; proficiency with SAS and/or other applications; Two (2) years of experience in **data analysis**; demonstrated knowledge of health care and claims; demonstrated knowledge of various database management..

Data Analyst - new
NCI Information Systems, Inc. 39 reviews - Omaha, NE

NCI: As the **Data Analyst** (**Data Analyst**) at AdvanceMed, a subsidiary of NCI, you will perform in-depth... and Sr. **Data Analysts** to provide proactive **data**...
2 days ago - save job - email - more...

vs.
informative structured snippets
generic query-biased snippets

Figure 1: Structured snippets provide more details about a job directly on a SERP and therefore save time users spend looking for relevant results.

The key question then is *how to generate such snippets*? One can build an information extraction model following ideas in [4, 6]. The problem here is that despite aiming to minimize labelling efforts, this approach still requires some training to be conducted manually. The problem becomes more severe if we take into account that major search engines operate internationally and hence training sets must be created for each language. To handle this complication we propose an unsupervised approach for structured snippets generation that doesn't require manual data annotation.

2. OUR APPROACH

The inspiration for our unsupervised approach comes from the paper [2], where the authors discuss how very large amounts of data make it possible to tackle complex problems, like machine translation, scene completion, and natural language disambiguation, with relatively intuitive algorithms. The wisdom of the paper is that *one should define a non-parametric model and statistically fit it with very large amounts of data. In this case, since "most" of the variability of the phenomenon is captured in the data, the model will have quite high predictive ability.* We first present a method for fully automatic training set construction from a collection of job postings, which is the key trick in this work, and

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

then we describe a machine learning algorithm, which uses this training set to build a model for structured snippets generation. Since we want to generate snippets, which contain information from responsibilities and requirements sections, our training set must contain positive examples from these sections (target classes) and also have some negative examples representing the irrelevant content.

Our approach to generate the training set in an unsupervised way is based on the observation that despite being free text documents, job postings mainly consist of the following sections *in the presented order*: short attributes (job title, company, location, and etc.), responsibilities, requirements, and company description. Moreover, some job postings are so well-structured that just one declarative rule is enough to extract information from them. Specifically, we noticed that a reasonable percentage of pages (about 10% based on experiments) contain a section header[1] followed immediately by a set of sentences wrapped in a "<li>" tag. We can extract such sentences and use them during training as positive instances for the corresponding class. Negative instances ("Others" class) are generated by taking sentences preceding the responsibilities and following the requirements sections from pages where these sections were detected by the strict rule defined above. The more job postings are crawled and added to a search index, the more training instances can be generated. In the limit, we can potentially "remember" all things companies look for in candidates and achieve very high extraction quality.

Once the training set is available, the learning task can be formally stated as follows. Given a set of labelled sentences from "Responsibilities", "Requirements", and "Others" sections of job postings, create an algorithm that can identify important sentences from a new job posting and assign them to the corresponding sections. Each sentence is used independently as an instance for a machine learning algorithm. As a model, we use a 2-level ensemble similar to stacking. At a base level we train 3 textual classifiers based on unigrams, bigrams, and trigrams. At a meta level we mix binarized predictions of the base algorithms as well as linguistic features, e.g. sentence length, capitalization, POS tags. As a learning algorithm, at both levels we use SVM with a linear kernel, which scales to very large datasets. For a new page, having extracted the sentences, we assemble the snippet by selecting a mutually diverse set of sentences following the maximum marginal relevance principle. For the responsibilities section we additionally break sentences into "*VP NP*" fragments in order to make snippets more informative and representative of activities associated with the potential job.

3. EXPERIMENT

To evaluate the proposed approach we crawled a corpus of 1,101,482 pages representing job postings. The set of URLs was generated by submitting queries for top job titles[2] to Indeed.com and web scraping the corresponding SERPs. The destination pages for each URL were downloaded and stored locally. We then selected 10 distinct job titles spanning multiple industries (16,054 pages total) and annotated 100 randomly sampled pages for each. The annotation speed with the custom-built annotation tool was about 2 minutes per page, which demonstrates the cost one must accommodate if they follow a classical supervised approach. We used 10-fold cross validation for the supervised approach evaluation. The proposed unsupervised approach was trained on sentences automatically extracted from 16,054 pages excluding the labelled pages to make sure we evaluated the performance fairly. Since snippets generation is a precision-oriented task (due to limited space available on a SERP), we tracked 2 metrics: *precision* averaged across job titles and *page coverage* representing percentage of pages for which at least one relevant sentence was extracted. The corresponding results are presented in Table 1 and Figure 2.

Section	Metric	Supervised	Unsupervised
Resp.	Precision	0.86 ± 0.09	0.84 ± 0.08
	Page Coverage	0.80 ± 0.12	0.92 ± 0.07
Req.	Precision	0.91 ± 0.07	0.88 ± 0.07
	Page Coverage	0.94 ± 0.05	0.97 ± 0.04

Table 1: Unsupervised approach achieves similar values for precision while having higher coverage due to much bigger training set generated automatically.

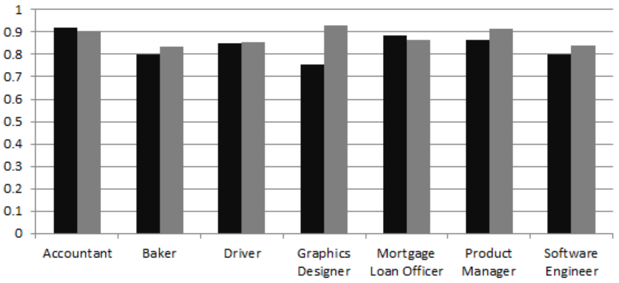


Figure 2: Precision (graphed for unsupervised case) stably achieves high values across job titles. Black bars represent responsibilities, gray – requirements.

To conclude, in this work we demonstrated that informative structured snippets can be generated in an unsupervised way. In future work we plan to conduct a study with real users and evaluate how such snippets help searching.

4. REFERENCES

[1] L. B. Chilton and J. Teevan. Addressing people's information needs directly in a web search result page. In *Proceedings of WWW 2011.*

[2] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 2009.

[3] J. Li, S. Huffman, and A. Tokuda. Good abandonment in mobile and PC internet search. In *Proceedings of ACM SIGIR 2009.*

[4] U. Y. Nahm and R. J. Mooney. A mutually beneficial integration of data mining and information extraction. In *Proceedings of AAAI 2000.*

[5] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of ACM SIGIR 1998.*

[6] T.-L. Wong, W. Lam, and B. Chen. Mining employment market via text block detection and adaptive cross-domain information extraction. In *Proceedings of ACM SIGIR 2009.*

[1] Identifiable with just one seed word per section, like "Responsibilities:" and "Requirements:" used in our experiments.

[2] http://www.indeed.com/find-jobs.jsp