Medtree: A Search Engine for Medical Professionals

Neel Guha, Errol Ozdalga, Matthew Wytock

Medtree: A Search Engine for Medical Professionals

Neel Guha
Stanford University
nguha@stanford.edu

Errol Ozdalga
Stanford University
eozdalga@stanford.edu

Matthew Wytock
Google
mwytock@cs.cmu.edu

Abstract

Users struggle with keyword based search engines like Google or Bing because queries can have multiple interpretations and search engines fail to understand the context in which the user is looking for information. This failure leads to search results that are either inappropriate or contextually irrelevant. In this paper we describe algorithms which, utilizing information about the user's context, scrape the web and process/filter candidate sites that could be used to create a customized context specific search engine for the user. We used the algorithm to create Medtree, a customized medical search engine for doctors at Stanford Hosptial. In evaluations we demonstrate Medtree's superiority to Google for medical queries.

I. The Problem

Users of web search tools (such as Google or Bing) often look for information in a predefined context. This context is shaped by factors such as:

1. The user's background (age, location, etc)
2. Prior information the user may know about the subject
3. Information need, e.g. what the user intends to do with the information

Unfortunately, many keyword based search engines struggle because they fail to understand the user's context and provide results that are not contextually appropriate. In order for a result to be contextually appropriate, it must fulfill the following criteria:

1. **It must be topically relevant to the user.** The result must be relevant to the user's query within the user's context. Because queries can have multiple interpretations and contain ambiguity (the query "Lincoln" could refer to the car or the American President) a search engine may return a result that is related to the query but outside the scope of the user's context.
2. **Results must be authoritative.** The user may require the result meets a certain threshold of accuracy and prefer more authoritative sources (research publications, acknowledged websites) to other sources (forums, personal blogs, etc).

We would like to create a system that can avoid the listed drawbacks of common keyword based search engines by providing the user with contextually appropriate and authoritative results. In this paper we use the intuition that rather than adjusting ranking algorithms to promote appropriate sites and demote inappropriate sites, we can instead identify a corpus of appropriate sites and limit results to user queries to those sites. We describe an algorithm that automatically generates a corpus of relevant sites for specific context. Using the Google Custom Search Engine infrastructure, we also show how this corpus can be used to create a search engine.

We tested our algorithm with doctors in the Stanford Department of Internal Medicine. Doctors frequently struggle with generic keyword search engines as most results are targeted solely towards patients and fail to meet their needs. We used our algorithm to create Medtree, a medical search engine (medtree.stanford.edu) for these doctors. This search engine was demonstrated to be superior through comparative evaluations against Google.

In the rest of this paper we describe the methodology of our algorithm and its performance. For clarity, we use the following terminology:

- **Context**. The circumstances in which a user is looking for information as shaped by factors such as their background, prior information known, and need for information.
- **User(s)**. Unless otherwise specified, we define the user as an individual searching for information only in the stated context.
- **Appropriate (or Good) sites**. Site's that meet the user's threshold for authority and are topically relevant to the context.

II. Related Work

There are a number of topic/task specific search engines, especially in medicine, such as ([22], [2], [11]). However, these examples typically restrict their attention to a particular kind of information (in this case, scholarly articles and patents). In contrast, we are interested in searching across a large number of sites that contain relevant information.

Earlier work ([9], [8]) introduced the need for course specific search engines and the failure of Google to provide adequete results to queries from history students. In this work we generalize the problem to the failure of keyword based search engines to understand user context. In addition, we provide a much more generalized algorithm that differs significantly in its approach from ([9], [8]).

Though the use of context in search is not new [13], the term "context" is used to refer to many different phenomena. Indeed, everything outside of the query terms that is possibly relevant is lumped under the term "context". We now examine some of the well known kinds of context and see how they relate to our notion of context.

Often, queries are drawn from a document (for example, by highlighting a phrase). Earlier authors ([4], [12]) study the problem of using either the whole document or the words surrounding the query to bias the search results. Our work can be seen as extending this notion of 'neighborhood' context from beyond a single document to a whole corpus that characterizes the context.

Another notion of context is rooted in the idea that the link structure of a certain locality of the web could capture the context. This is seen in the work [10] that tries to create a topic or context sensitive version of Page Rank [17]. We, too, try to capture the notion of topic in research contexts, but do it by analyzing the text in pages.

Search history has been used to personalize search results ([21], [16]). [3] uses both the user's history and meta data to personalize search results. Our work can be seen as an extension of some of this work, where, instead of using a single user's search history, we use a whole community's search patterns and judgments to create a corpus for that community. Social tagging systems such as del.icio.us [23] and folksonomies [15] have tried to collaboratively filter web pages. Most of the prior work in these systems has focused on the problem of getting a community of users to use a consistent vocabulary to label a set of pages with tags. In contrast, we use a combination of reference texts and a small number of labeled examples to build our corpus. We hope that in future, we can use techniques similar to those described in these papers to help our system improve with continued usage.

III. Methodology

Our algorithm first identifies a candidate set of sites and scrapes their relevant pages. We construct a model to classify these candidate sites as appropriate and inappropriate. Using the Google Custom Search Engine Infrastructure, we then create a search engine out of the appropriate sites.

I. Identifying Candidate Sites

In order to capture the context of Stanford doctors we start with an intial set of their web queries. In order to capture more of the context we expand this set to include similar, co-occuring queries (provided by Google Trends). For example, from the seed query [myotonic dystrophy], we can identify 48 more related queries, including [friedreich's ataxia], [homocystinuria], [tuberous sclerosis], etc. We issue each of these queries to Google, collecting the top ten results for each and group the results by site. This set of sites forms our candidate pool. Our goal is to curate this set and determine which are appropriate for the user.

II. Need for automation

If the majority of results are contained by a handful of sites it would be viable to manually curate the candidate set of sites. However, as we later describe, the distribution of results across sites is fairly uniform, necessitating automatic curation.

III. Constructing the model

In order to train the model we require an initial set of known classified sites. We randomly selected 100 sites which appeared in 5 or more results and had doctors at Stanford Hospital label them as either "appropriate" or "inappropriate". These sites formed the training data for our model. The features used in our model fell into two broad classes: textual similarity features and site metadata features.

IV. Textual Similarity Features

We can evaluate candidate sites based on their textual similarity to labelled sites. By preferring sites that are more similar to appropriate sites and less similar to inappropriate sites, we identify sites related to the context. When analyzing sites, we restrict our attention to the pages returned from queries we issued. This ensures that only portions of a site relevant to a context are captured and prevents unrelated sections of the site (which would not appear in the result stream) from biasing the analysis of the site.

IV.1 Word Frequency

A naive approach is to include every term on a site as a feature for that site (with the value being the term's frequency). Candidate sites that contain the same words (with similar frequencies) as the labelled appropriate sites are more likely to be preferred. This approach also has several drawbacks as term frequency isn't an accurate representation of the importance of a term on a page.

IV.2 Cosine Similarity

To address this we compute a TF-IDF weighted cosine similarity score for candidate sites. We weight the words in a document by their TF-IDF score (the product of the term's frequency in the document (TF) and the inverse document frequency (IDF) of the term). We then calculate the cosine similarity between each candidate site and the set of appropriate sites and the set of inappropriate sites.

However, to account for the (sometimes) significant overlap between medical and non medical terms we also determine the 500 terms with the greatest frequencies across all appropriate sites and the 500 terms with the greatest frequencies across all inappropriate sites. We weight each term using a fractional score based on its frequency (we call this set S_g for appropriate sites and S_b for inappropriate sites).

Then for each candidate site S_i we similarily weight the terms on the site and calculate its similarity to S_b and S_g. By preferring sites more similar to S_g and less similar to S_b, we can identify appropriate sites.

V. Metadata

Textual analysis alone is not sufficient to determine if a site is appropriate as it only captures the content on the site. Site metadata however can provide information about the site that is helpful in determining site authority.

V.1 Generic Top Level Domains

A naive approach is to use the top level domain (TLD) of a site (.com, .edu, etc) as a model feature. The TLD of a site broadly correlates to the site's content (.edu is used for education organizations, .mil is used for the military, etc). However, this alone isn't sufficient as TLD's are incredibly broad.

V.2 Wikipedia Categories

Since many candidate sites have corresponding Wikipedia articles, we can also use the Wikipedia category hierarchy to help classify candidate sites. We map each candidate site to its respective Wikipedia article. All Wikipedia articles are organized in a hierarchical structure. For example, the Wikipedia article A (corresponding to a website) may be a member of the category B. Because B is a child of the category C, we can say that A is a member of both B and C. Since Wikipedia categories can often be quite specific, it is important to capture the broader parent categories (recursively) an article/site belongs to. We include all the categories (and their parents) of the Wikipedia article associated with a site as candidate features in our model. Because there are some sites which don't map to Wikipedia categories, our model incorporates the Wikipedia categories as binary features. A site S_i is either part of a category C_i (value = 1) or isn't (value = 0).

VI. Feature Selection

We now have the following candidate features:

1. A TFIDF weighted score for cosine similarity to the reference text (one score per site).
2. Scores for a sites similarity to the top 500 common appropriate terms and top 500 common inappropriate terms (two scores per site)
3. Each term in the corpus with its frequency on the site as its value .
4. The top level domain for each site.
5. A binary feature for each Wikipedia category, specifying whether the Wikipedia article corresponding to the site is a member of that category.

Both (3) and (5) result in a large number of features and using all will lead to overfitting. We need to pick a subset of the features which is small enough that overfitting is unlikely, but which has enough information to be able to do the prediction. This problem of feature selection, especially for text classification, has been studied extensively ([19], [24], [5], [14]).

There are many metrics available for evaluating the quality of a candidate set of features. As discussed in [24], the most commonly used metrics in text classification include Inverse Document Frequency (IDF), Mutual Information, χ^2 statistic, and Information Gain. Given the sparsity of some of our features, Information Gain is the best suited metric (IDF struggles performance wise and Mutual Information and χ^2 statistic are unreliable for low frequency terms).

The goal is to pick a subset of features of size M ($\ll$ N) that provides sufficient predictive capability. [24] presents a simple algorithm in which only features with more than a certain threshold of IG are selected. We adapt this algorithm for our purposes. We run this algorithm separately for the textual terms and for the Wikipedia categories to select a subset of each.

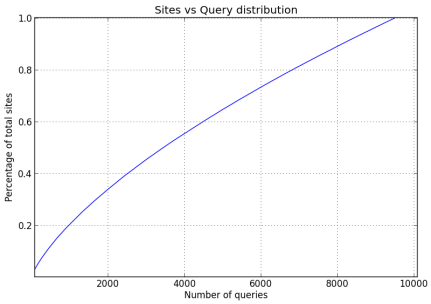

Figure 1: *Site distribution over queries*

For textual features, we assume that the different words on a site are independent of each other. Starting with an empty set of features, we calculate the Information Gain of the remaining features and add the feature with the highest value. We continue this until the information gain of the next feature is less than some threshold T. To summarize, we select all features $< f_1, f_2, ..., f_i >$ where $IG(f_i) > T$. For textual features we set $T = 0.001$.

The independence assumption is harder for Wikipedia categories given the hierarchical nature of Wikipedia. We modify the algorithm above to account for this. After we include each Wikipedia category feature f_i into the set, we remove all features f_j not in the set that are related to f_i (i.e. a parent or child of f_i). We used $T = 0.05$ for categories. Using this algorithm we get 35 category features.

VII. Building the Model and Search Engine

We used a logistic regression classifier from the open source python scilab toolkit [1] and ranked sites by predicted probability estimates.

The Google Custom Search Engine infrastructure allows us to create a search engine over a set of sites. Results in this search engine will be restricted to certain predefined URL patterns or sites. The CSE infrastucture handles the web crawling, indexing, and maintenance of the search engine. After ranking site's by their predicted probability estimates, we select the top N sites (N = 100, 200, 300) and feed these to the search engine. Medtree, the search engine created from this process, can be found at medtree.stanford.edu.

IV. Experimental Results

We started with a set of 221 medical queries from doctors at Stanford Medical School and expanded this set to to 13328 queries. We retrieved the top ten results for each of these queries from Google, producing 95043 results from 15514 sites. Figure 1 shows the distribution of sites over queries (See Section III.II Need for Automation).

The top 186 of 15514 the sites were manually examined and classified into two groups: those that were appropriate for medical professionals and those that were not. 134 of the sites were either irrelevant, targeted at patients or were from non-authoritative sources (such as user generated sites), leaving us

	Google better	Medtree better	Same
Doctor 1	0	13	2
Doctor 2	0	14	1
Doctor 3	0	14	1
Doctor 4	0	14	1

Table 1: *Evaluation of Medical CSE by 4 Doctors*

Baseline	N= 100	N= 200	N= 300
28%	86%	85.5%	84.3%

Table 2: *Percentage of good sites in top N sites*

with 52 good sites; randomly selecting sites (from those returned by Google for this context) would thus result in only 28% being appropriate for the medical search context. The curation was done by doctors at Stanford Medical School.

We randomly selected 100 sites which appeared in at least five results for labelling by doctors at Stanford Hospital. 72 of these the sites were classified as inappropriate and 28 were classified as appropriate. If our model adopted a randomized approach (randomly ranking sites) than approximately 28% of the top N (N = 100, 200, 300) would be appropriate. This establishes the baseline for our model at 28%.

We evaluated our algorithm for selecting sites to include in the contextual CSE using two different approaches. First, doctors at Stanford Hospital evaluated the algorithm by counting the number of appropriate sites in the top N (N = 100, 200 and 300) ranked sites from the model. We compare these results against the baseline (fraction of good sites in the results that a normal Google search would give us for these queries), which is 28%. Table 2 shows the percentage of good sites as a function of N. As can be seen, our algorithm performs significantly better than the basline. Secondly, in order to evaluate our algorithms in a more practical setting, we built a CSE with 513 sites. Of these 513, 28 belonged to the set of manually labelled appropriate sites 485 were sites that our model had assigned a probability of 85% or more of being appropriate. We performed a side by side evaluation on 15 medical queries with four doctors from Stanford Medical School. As seen in Table 1, the results from the CSE are overwhelmingly preferred.

Though recall is a reliable and commonly used metric in information retrieval and classification, determining the significance of a recall score is difficult for our algorithm. Recall is defined as the fraction of the set of relevant documents that are succesfully returned for a given query. Our focus however, is not on constructing a more effective algorithm for querying a corpus of documents. Rather, we attempted to identify a corpus of documents appropriate for the user's context from the web. In such a case, it is difficult to quantify or identify the documents that our algorithm missed. This is an area we are interested in continuing to examine.

Both these evaluations clearly demonstrate the efficacy of our algorithm for identifying sites appropriate for medical professions and that Medtree is substantially preferred to vanilla keyword based search engines for this context.

jco.ascopubs.org	gsksource.com
aac.asm.org	ccforum.com
ahajournals.org	rjc.asm.org
oxfordjournals.org	gut.bmj.com
gsksource.com	hepatitis.va.gov
plosone.org	haematologica.org
hivinsite.ucsf.edu	onclive.com

Table 3: *Top scoring medical sites*

V. Discusion

We analyze the behavior of our model by examing its failures. Of the inappropriate sites in the top 100, 9 belonged to pharmaceutical companies. These pharmaceutical sites are highly ranked by our algorithm because they share the vocabulary found in good medical sites. The doctors conducting evaluations thought they were inappropriate for inclusion into the custom search engine because of their commercial nature. Of the remaning losses, 2 were sites in a foreign language (Spanish and Portuguese) and inaccessible to English speaking doctors. The reason these sites appeared is that many of the technical medical terms are the same in multiple languages. The feature selection algorithm strongly preferred medical terms, leading to differences in the non-medical terms getting overshadowed. The only two patient targeted sites in the top 100 are kidney.org and druginfonet.com.

It is important to note that the initial training data did not have any pharmaceutical sites or sites in a foreign language. In the future, we'd like the algorithm to adjust itself when this kind of phenomenon occurs. Ideally, the algorithm could utilize some form of user feedback (interpreting clicks from the user as feedback) to continually tune the model and eliminate new classes of inappropriate sites.

The work described in this paper shows that many of the inappropriate results returned because of the inability of keyword based search engines to identify user contexts can be avoided through context specific search engines. More importantly, our work outlines an accurate and minimal labor approach to automating the creation of these tools. To our knowledge, this is the first work on building learnable models for generating level-appropriate, on topic results for such contexts.

VI. Acknowledgements

We would like to thank Prof. Lam from Stanford for guiding our research and for feedback on several iterations of the paper. We would like to thank Vineet Gupta, Carolyn Au, Mudita Jain and Vivek Raghunathan for feedback on the paper. We would also like to thank Ramanathan Guha, Prof. Ladabaum and Prof. Musen for their mentorship through the creation of Medtree.Stanford.edu.

References

[1] S. L. Campbell, J.-P. Chancelier, and R. Nikoukhah. *Modeling and simulation in SCILAB*. Springer, 2010.

[2] M. Cariaso and G. Lennon. Snpedia: a wiki supporting personal genome annotation, interpretation and analysis. *Nucleic acids research*, 40(D1):D1308–D1312, 2012.

[3] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using odp metadata to personalize search. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2005.

[4] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

[5] G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.

[6] A. Genkin, D. D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007.

[7] Google. Google Scholar. Available at: http://scholar.google.com.

[8] N. Guha. Course specific search engines: A study in incorporating context into search. In *Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval*, ESAIR '13, pages 33–36, New York, NY, USA, 2013. ACM.

[9] N. Guha and M. Wytock. Course-specific search engines: semi-automated methods for identifying high quality topic-specific corpora. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 1247–1252. International World Wide Web Conferences Steering Committee, 2013.

[10] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *Knowledge and Data Engineering, IEEE Transactions on*, 15e(4):784–796, 2003.

[11] D. Karolchik, R. Baertsch, M. Diekhans, T. S. Furey, A. Hinrichs, Y. Lu, K. M. Roskin, M. Schwartz, C. W. Sugnet, D. J. Thomas, et al. The ucsc genome browser database. *Nucleic acids research*, 31(1):51–54, 2003.

[12] R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar. Searching with context. In *Proceedings of the 15th international conference on World Wide Web*, pages 477–486. ACM, 2006.

[13] S. Lawrence. Context in web search. *IEEE Data Eng. Bull.*, 23(3):25–32, 2000.

[14] D. D. Lewis. Feature selection and feature extraction for text categorization. In *Proceedings of the workshop on Speech and Natural Language*, pages 212–217. Association for Computational Linguistics, 1992.

[15] A. Mathes. Folksonomies-cooperative classification and communication through shared metadata. *Computer Mediated Communication*, 47(10), 2004.

[16] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch. Personalized search on the world wide web. In *The Adaptive Web*, pages 195–230. Springer, 2007.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[18] M. F. Porter et al. An algorithm for suffix stripping, 1980.

[19] M. Rogati and Y. Yang. High-performing feature selection for text classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661. ACM, 2002.

[20] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[21] M. Speretta and S. Gauch. Personalized search based on user search histories. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 622–628. IEEE, 2005.

[22] US National Library of Medicine, National Institutes of Health. Pubmed, 2013. Available at: http://www.ncbi.nlm.nih.gov/pubmed.

[23] R. Wetzker, C. Zimmermann, and C. Bauckhage. Analyzing social bookmarking systems: A del. icio. us cookbook. In *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pages 26–30, 2008.

[24] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.