Towards Less Biased Web Search

Xitong Liu
University of Delaware
Newark, DE, USA
xtliu@udel.edu

Hui Fang
University of Delaware
Newark, DE, USA
hfang@udel.edu

Deng Cai
The State Key Lab of CAD&CG
Zhejiang University, China
dengcai@cad.zju.edu.cn

ABSTRACT

Web search engines now serve as essential assistant to help users make decisions in different aspects. Delivering correct and impartial information is a crucial functionality for search engines as any false information may lead to unwise decision and thus undesirable consequences. Unfortunately, a recent study revealed that Web search engines tend to provide biased information with most results supporting users' beliefs conveyed in queries regardless of the truth.

In this paper we propose to alleviate bias in Web search through predicting the topical polarity of documents, which is the overall tendency of one document regarding whether it supports or disapproves the belief in query. By applying the prediction to balance search results, users would receive less biased information and therefore make wiser decision. To achieve this goal, we propose a novel textual segment extraction method to distill and generate document feature representation, and leverage convolution neural network, an effective deep learning approach, to predict topical polarity of documents. We conduct extensive experiments on a set of queries with medical indents and demonstrate that our model performs empirically well on identifying topical polarity with satisfying accuracy. To our best knowledge, our work is the first on investigating the mitigation of bias in Web search and could provide directions on future research.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval
General Terms: Algorithm
Keywords: topical polarity, search bias

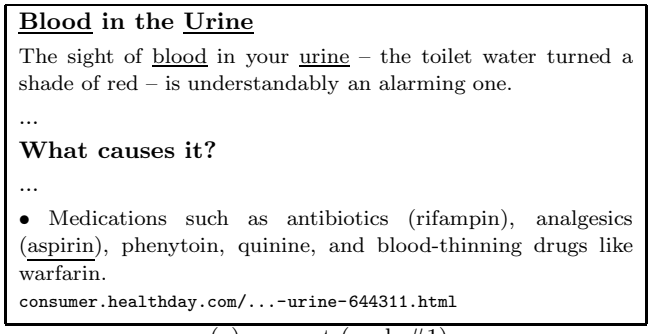
1. INTRODUCTION

The advance of Web search engines provides much easier access to huge volume of information with coverage of remarkably wide spectrum. Delivering accurate information clearly is the crucial functionality of Web search engine, as the results may have direct impacts on people's decisions and actions thereafter. However, an existing study [9] reveals that biases are observed during search, as the information users seek or search engine returns significantly deviates from the truth in two aspects: (1) Most results support the query while only a few disapprove it. (2) Results supporting the query are ranked higher than results disapproving it.

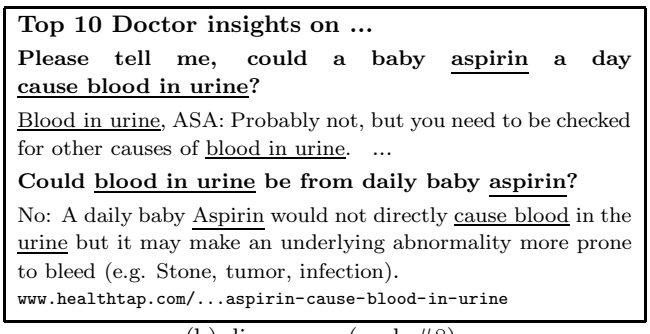
This search bias problem is particularly crucial in medical domain, as any incorrect decision after search may lead to undesirable consequences on the health condition of users. White et al. [10] found that about 3% of search queries have medical intent on Bing, reflecting the high demand of delivering accurate information to end users. Consider the query "can aspirin cause blood in urine" which is expressed in a question, the user may have symptom of "blood in urine" already and have taken "aspirin" before. She wants to confirm whether taking "aspirin" would be the cause under the preconception that "aspirin" caused her symptom. Such process is described as *confirmation bias*, a common psychological tendency which most people have in the interpretation of information. On the other side, Web search engines would return biased results which favor user's preconception. Among the top 10 results from Google, 8 documents are considered as relevant based on our manual assessment, and 6 documents support the belief while only 2 documents disapprove it. Figure 1 presents two relevant documents. The #1 ranked document in Figure 1(a) lists aspirin as one possible cause. Conversely, the #8 ranked document in Figure 1(b) shows that baby aspirin (i.e., low-dose aspirin) would probably not cause blood in urine, and it is the first document against the query. Due to the fact that user is much more likely to click top ranked results while disregarding lower ranked ones, such bias would lead her to believe that aspirin *caused* blood in urine, even though she only took low-dosage.

In this paper, we propose to alleviate the bias in Web search through predicting the topical polarity of documents to balance search results. The topical polarity represents the overall tendency of one document about whether it supports (e.g., Figure 1(a)) or disapproves (e.g., Figure 1(b)) the belief in query. When presented with balanced results, users would have deeper and wider perspective about the topic, seek for answer in a comprehensive approach and make wiser decision thereafter. To reach our goal, we propose to perform binary classification over retrieved documents to predict the topical polarity in a supervised learning approach. In particular, we first extract query-representative textual segments from documents, and train a learning model on convolutional neural network [5], an effective deep learning approach which can unfold useful features automatically. The ultimate document-level prediction is derived based on the aggregation of prediction from textual segments. Exper-

ICTIR'15, September 27–30, Northampton, MA, USA.
 ISBN 978-1-4503-3833-2/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2808194.2809476.



(a) support (rank #1)



(b) disapprove (rank #8)

Figure 1: Excerpts of two documents for query "*can aspirin cause blood in urine*".

imental evaluation over a set of question-like queries with medical intents demonstrates that our model could predict topical polarity with satisfying accuracy. Although we evaluate on queries in medical domain only, our model is not limited to any specific domain and can be generalized to other domains (e.g., political domain) which require less biased and more impartial results.

2. RELATED WORK

Biases have been a constant problem on Web search engine and received considerable attention from different aspects. Ieong et al. [3] investigated domain bias, a phenomenon in Web search that users' tendency to prefer a search result just because it is from a reputable domain, and found that domains can flip a user's preference about 25% of the time under a blind domain test. White [9] found that users show biases by favoring information confirming their belief when conducting search, and they are subject to the bias of search engine which usually returns more results to support the belief in query regardless of the truth. However, none of them studied how to reduce the bias in search results.

Motivated from sentiment analysis [7], opinion retrieval aims at retrieving documents with subjectively inclined opinion (positively or negatively) towards the query. Most approaches fall into two categories: (1) Lexicon-based, which builds a list of terms with known sentiment orientation [1]. (2) Classification-based, which builds a classifier from training data with opinionated labels and apply the classifier on the testing data thereafter to estimate the opinion score. Representative work include Zhang et al. [11], He et al. [2]. Different from sentiments which are people's subjective attitude (e.g., like or dislike), topical polarity is the tendency about whether one document supports or disapproves the belief in query (e.g., something can cause some symptom), and therefore requires comprehensive understanding to identify it, making the prediction more challenging.

Our research is related to search result diversification [8], which tackles ambiguity in query and redundancy in search results to achieve both high coverage and novelty at the same time. However, there is a clear distinction: they identify subtopics for each query as different aspects and rank documents to balance coverage and novelty based on them, we aim to balance result coverage on two aspects only: supporting and disapproving the belief in query.

3. TOPICAL POLARITY PREDICTION

The topical polarity of a document can be modeled as a binary flag indicating whether the document supports the belief in query, or conversely, disapproves it. Predicting topical polarity of a document is not a trivial problem. We propose to perform supervised classification to solve the problem. There are three major challenges: (1) How to generate the feature representation of a document? (2) How to unfold latent patterns in the feature representations? (3) How to determine the topical polarity based on feature representations? We would discuss our approach to tackle each of them from Section 3.1 to 3.3.

3.1 Textual Segment Extraction

Generating the feature representation for documents is the first step in many IR applications. Language model has been proved to be a simple yet effective representation in many document retrieval models. However, it could not work well in our case as it drops the sequential dependence between terms, making it impossible for comprehensive understanding of documents.

We propose to extract query-related textual segments as the feature representation of documents. By textual segments, we mean any term sequences from the documents, which could include title, sub-title, sentence, etc. The advantages for textual segment over language model include: (1) It retains the term sequential dependence, which is vital for document comprehension. (2) It filters out noises in documents. While language model is estimated based on all the terms in the document, we only select the textual segments which are most relevant to the query, since empirical observation reveals that the relevance of a document is mainly determined by a few textual segments. Consider the example in Figure 1, only the textual segments shown in the excerpt would suffice to determine the relevance.

Algorithm 1 ExtractTextualSegments

Input: query q, document d
Output: Textual segment list seg_list

```
PL(q) = GeneratePostingList(q)
PL(d) = GeneratePostingList(d)
JPL = PL(q) ∩ PL(d) /* The joint posting list */
PQ ← [ ] /* A priority queue of term-position pairs */
for term ∈ JPL do
    pair = MakePair(term, Pop(JPL(term)))
    Push(PQ, pair)
end for
upper = Max(JPL) /* The upper bound */
coord_list = LocateSegments(JPL, PQ, upper)
seg_list = GenerateSegments(d, coord_list)
return seg_list
```

Clearly, a representative textual segment should be relevant to the query. We propose a novel segment extraction al-

gorithm based on the snippet generation algorithm in search engine result page. The details are described in Algorithm 1, which consists of several major steps:

1. Generate the joint posting list at line 3. It covers all the terms shared by query and document.
2. Prepare a priority queue as segment window at line 7. The postings of each term serve as anchors for the window to be shifted from beginning to the end.
3. Locate all the valid segments in *LocateSegments* at line 10.
4. Generate all the segments by *GenerateSegments* function based on the coordinates of segments at line 11.

The details of function *LocateSegments* are illustrated in Algorithm 2. We have a segment window covering at least two query terms, and shift the window from the beginning of document to the end. The priority queue is employed to ensure we always have the valid anchors for the window. If the window length exceeds the length constraints, we exclude the last term to shrink the window and find segments via a recursive call at line 18.

Algorithm 2 LocateSegments

Input: joint posting list JPL, priority queue PQ, upper bound of position *upper*
Output: List of segment coordinates *coord_list*

```
1: coord_list ← [ ]
2: while True do
3:    Sort(PQ) /* Sort the priority queue by position */
4:    /* A valid segment should cover at least 2 terms */
5:    if Len(PQ) < 2 then
6:       break
7:    end if
8:    if Tail(PQ)[1] > upper then
9:       break/* We reached the upper bound and stop */
10:   end if
11:   seg_len = Tail(PQ)[1] − Head(PQ)[1]
12:   if seg_len > MIN_SEG_LEN then
13:      if seg_len < MAX_SEG_LEN then
14:         Push(coord_list, PQ) /* It is a valid segment */
15:      else
16:         /* The window is too long, we exclude the last
            term to shrink it and find segments recursively */
17:         PQ* = PQ \ Tail(PQ)
18:         ext = LocateSegments(JPL, PQ*, Tail(PQ)[1])
19:         coord_list = coord_list ∪ ext
20:      end if
21:   end if
22:   Pop(PQ) /* Shift segment window to the next term*/
23:   pos_list = JPL(Head(PQ)[0])
24:   if len(pos_list) > 0 then
25:      pair = MakePair(Head(PQ)[0], Pop(pos_list))
26:      Push(PQ, pair)
27:   end if
28: end while
29: return coord_list
```

Due to the fact that *LocateSegments* will only find segments which start and end with query terms, it would break the integrity of sentences as in most cases query terms are spread in the middle of sentences. For example, a possible segment in Figure 1(b) would be "aspirin a day cause blood in urine". The missing of head and tail terms would cause the loss of useful information, and sometimes the meaning would be totally different. To mitigate such information loss, we extend the segments to the boundaries in the original document based on HTML tags, punctuation marks to make sure the segments consists of whole sentences in function *GenerateSegments* at line 11 in Algorithm 1. In the example we mentioned before, the segment would be extended to "Please tell me, could a baby aspirin a day cause blood in urine" to retain its original meaning.

After the segments are extracted, we need to choose the most relevant ones as feature representation. Since segments are essentially short documents, existing document retrieval models could be employed to rank them. We choose query likelihood, an effective and robust retrieval model:

$$p(q|s) = \prod_{w \in q} p(w|\theta_s)^{n(w,q)}, \tag{1}$$

where $n(w, q)$ denotes the number of w in q, θ_s is language model of segment s. To reach better performance, Dirichlet smoothing is applied for θ_s. The smoothing parameter μ is set to 250 based on preliminary results. Top k segments will be chosen for training and testing later on, denoted as $S(d)$.

3.2 Learning to Predict Topical Polarity

Recent advances in deep learning have produced remarkable results on pattern recognition in different areas. One advantage of deep learning is the capability of unfolding useful features from data automatically, which fits our task well. Convolutional Neural Network (CNN) [5], which is a type of feed-forward artificial neural network, has been extensively applied in pattern recognition on image data. There have been some pioneer efforts on adapting CNN on textual data with promising results. Kim [4] proposed an effective CNN framework for sentence classification. Word vectors [6] are leveraged to transform sentences to two dimensional matrices similar to image data with fixed width and variable height. The intrinsic characteristic of convolution make it independent of absolute position of terms and capable of capturing latent semantic relations as patterns.

We simply adopt the framework by Kim and use the 300-dimensional word vectors trained from Google News corpus. The shape of filters include 3×300, 4×300 and 5×300, and for each shape we have 100 filters. More technical details can be found in Kim's paper [4].

3.3 Aggregative Prediction

As multiple textual segments are extracted as feature representation of one document, different segments from one document may not share the same polarity, we formalize the problem as regression by predicting how strongly it supports or disapproves the query. To simplify the training process, we assume that all the textual segments share the same topical polarity of the document. We feed CNN with top k segments for document d, aggregate the predictions for all textual segments $s \in S(d)$ and apply majority vote to perform prediction. Formally, we have:

$$p(T = i|q, d) = \sum_{s \in S(d)} w(s|q) \cdot \mathbb{1}(l(s|q) = i), \tag{2}$$

where $T \in \{-1, 1\}$ is a binary variable for topical polarity (1 for support and -1 for disapproval), $l(s|q)$ is the prediction for s from CNN, and $\mathbb{1}(l(s|q) = i)$ is an indicator function to select segments with same prediction as i. $w(s|q)$ is the weight function for s with regard to q. We use the query likelihood in Equation (1) to approximate it: $w(s|q) = p(q|s)$.

Table 1: Performance Comparison by Precision

Method	$k = 5$		$k = 10$	
	Macro-Avg	Micro-Avg	Macro-Avg	Micro-Avg
Sent	0.6806	0.6834	0.6772	0.6812
Seg	0.7134	0.7132	0.7115*	0.7132
Seg-Ext	0.7190*	0.7239	0.7207*	0.7260
Seg-Exp	**0.7376***	**0.7441**	**0.7429**$^*_\dagger$	**0.7484**

$*$ and † denote improvements over **Sent** and **Seg** are statistically significant based on two-tailed paired t-test with $p < 0.05$.

4. EXPERIMENTS

There are 50 queries in our data set. 7 queries are from White's paper [9] as they are representative queries with bias. The other 43 queries are selected from topics of medical forums (e.g., drugs.com) with intensive debates. All queries are question-like inquiries (e.g., "Can I take tylenol during pregnancy") covering general usage and side-effects in medical domain. For each query, we retrieved top 20 documents from Bing and Google, and removed duplicated documents. We manually labelled the relevance of documents. For relevant documents, we further labelled the topic polarity (i.e., support or disapprove). For documents with controversial debates, we label it based on the dominant argument. Only relevant documents are used for evaluation. The average number of relevant documents per query is 18.34 and the average number of documents which disapprove the query is 7.40. The data set is available for download.[1]

We use the same setting for CNN throughout the experiments, and focus on the comparison of different textual segment extraction methods, as the quality of segments are crucial to the ultimate prediction accuracy. To evaluate the effectiveness of our segment extraction method in Section 3.1, we use top k sentences from a document as feature representation. Sentence candidates are generated based on HTML tags and punctuation marks, and ranked by query likelihood as in Equation (1) with the same Dirichlet smoothing as for text segments. We denote this method as **Sent**. Based on Algorithm 1, we implement three variations:

- No segment extension is applied in *GenerateSegments* function at line 11, and denote it as **Seg**.
- Segment boundary extension is applied to make sure all segments consist of whole sentences without marginal loss. It is denoted as **Seg-Ext**.
- For medicines with synonyms in knowledge base, we add them to the query to perform expansion for segment ranking. This is common for medicines with different band and chemical names. For example, "Tylenol" is the band name for "Acetaminophen". Same segment boundary extension is applied as above, and we denote it as **Seg-Exp**.

We conduct 10-fold cross-validation over the data and report both the macro and micro-average precision over the 50 queries. Results are reported with k set to 5 and 10. MIN_SEG_LEN is set to 10 and MAX_SEG_LEN is set to 100 in Algorithm 2 to limit segment length in $(10, 100)$.

The performance of all the methods are summarized in Table 1. We observe that our segment based extraction methods deliver significant better performance over **Sent**, implying that segments could provide higher quality feature representation than sentences. In-depth analyses show that relevant information may spread across multiple sentences, and our methods could extract the useful segments covering most relevant information, while **Sent** could only extract partial relevant information.

Furthermore, the superior performance of **Seg-Ext** over **Seg** reveals that the extension over segment boundaries does help as it could retain the useful information in its original context by mitigating the marginal loss. Besides, **Seg-Exp** could deliver further improvements over **Seg-Ext**, demonstrating that simple query expansion based on knowledge base could contribute more useful segments. We expect that more advanced query expansion method would bring further improvements. Note that the best prediction accuracy could be reached at 0.75, showing that CNN could effectively learn latent patterns from textual segments.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a novel model to perform topical polarity prediction on documents. It consists of textual segment extraction to generate feature representations for documents, an existing Convolutional Neural Network framework to unfold latent patterns from textual segments, and aggregative prediction of document based on segment predictions. Experimental evaluations on a real world data set demonstrate that our model could extract useful textual segments and reach promising prediction accuracy. The open availability of data set would help future research work on alleviation of Web search bias.

There are many directions for future work. We would like to study how to leverage topical polarity predictions to balance search results and mitigate search bias. Moreover, we would like to extend convolutional neural networks to better fit topical polarity prediction. Applying our model on Web-scale data and evaluate the impacts on Web search would also be interesting to explore.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number IIS-1423002 and the National Basic Research Program of China (973 Program) under Grant 20113CB336500. We thank the anonymous reviewers for their useful comments.

6. REFERENCES

[1] B. He, C. Macdonald, J. He, and I. Ounis. An Effective Statistical Approach to Blog Post Opinion Retrieval. In *CIKM*, pages 1063–1072, 2008.

[2] B. He, C. Macdonald, and I. Ounis. Ranking Opinionated Blog Posts using OpinionFinder. In *SIGIR*, pages 727–728, 2008.

[3] S. Ieong, N. Mishra, E. Sadikov, and L. Zhang. Domain Bias in Web Search. In *WSDM*, pages 413–422, 2012.

[4] Y. Kim. Convolutional Neural Networks for Sentence Classification. In *EMNLP*, pages 1746–1751, 2014.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *NIPS*, pages 3111–3119, 2013.

[7] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[8] R. L. T. Santos, C. Macdonald, and I. Ounis. Search Result Diversification. *Foundations and Trends in Information Retrieval*, 9(1):1–90, 2015.

[9] R. White. Beliefs and Biases in Web Search. In *SIGIR*, pages 3–12, 2013.

[10] R. W. White and E. Horvitz. Studies of the Onset and Persistence of Medical Concerns in Search Logs. In *SIGIR*, pages 265–274, 2012.

[11] W. Zhang, C. Yu, and W. Meng. Opinion Retrieval from Blogs. In *CIKM*, pages 831–840, 2007.

[1] `http://infolab.ece.udel.edu/~xliu/data/bias/`