Collecting Representative Social Media Samples from a Search Engine by Adaptive Query Generation

Virgile Landeiro
Department of Computer Science
Illinois Institute of Technology
Chicago, IL USA

Aron Culotta
Department of Computer Science
Illinois Institute of Technology
Chicago, IL USA

***Abstract*—Studies in computational social science often require collecting data about users via a search engine interface: a list of keywords is provided as a query to the interface and documents matching this query are returned. The validity of a study will hence critically depend on the representativeness of the data returned by the search engine. In this paper, we develop a multi-objective approach to build queries yielding documents that are both relevant to the study and representative of the larger population of documents. We then specify measures to evaluate the relevance and the representativeness of documents retrieved by a query system. Using these measures, we experiment on three real-world datasets and show that our method outperforms baselines commonly used to solve this data collection problem.**

***Index Terms*—classification, data collection, sampling bias**

I. Introduction

Scientific studies that use online data commonly require interaction with some kind of search engine. For example, Twitter's API may be used to identify tweets matching certain keywords. The validity of the study will hence critically depend on the representativeness of the data returned by the search engine. To collect data for such a study, one must have a mechanism to identify relevant documents. One method is to first manually identify keywords that likely indicate relevance, and then query the database/API to find matching documents. However, there are at least two key threats to the validity of this methodology: 1) the keywords are unlikely to span all types of relevant documents, so we may miss a potentially significant portion of relevant data (**coverage error**); and 2) if the keywords happen to over-represent documents associated with a specific subset of the population, then the study may reach erroneous conclusions (**sampling bias**). These problems are only exacerbated when there is high class imbalance.

ASONAM '19, August 27-30, 2019, Vancouver, Canada

ACM ISBN 978-1-4503-6868-1/19/08?/$15.00
https://doi.org/10.1145/3341161.3342924

To advance progress in this area, 1) we provide several empirical measures to quantify the amount of coverage error and sampling bias in a text dataset; 2) using these measures, we compare several querying methods on 3 real-world datasets; and 3) we introduce a new querying algorithm that directly aims to reduce coverage error and sampling bias, finding that it outperforms traditional methods[1].

II. Related Work

A. Computational social science

Exploiting online social media data to build observational studies is becoming increasingly common due to the diversity and quantity of such data. Olteanu et al. [1] and Cunha et al. [2] both collected data from social media (respectively Twitter and Reddit) to build observational studies. These examples all used some strategy to control for possible confounders (matching, propensity score) and thus are more robust to possible bias caused by these variables. However, concerns have been raised that the population on social media is not representative of the larger offline population leading to biased social media studies [3, 4]. Because these studies are used to infer real world outcomes, it is crucial that the bias induced by the data collection be minimal.

B. Active learning

Several active learning algorithms relate to our proposed method. For example, the problem of class imbalance pushed researchers in active learning to develop strategies to find documents belonging to rare classes [5]. Another common problem in the active learning field is how to choose the best next instance to label. Recent strategies combined informative and representative samples in order to achieve the best post-labeling performance [6]. Although the approach proposed in this paper takes inspiration from multiple techniques developed in active learning, it does not require human interaction and therefore has different aims from active learning. Instead, we wish to develop a more scalable, low-cost method that does

[1]A longer version of this paper and the code to reproduce experiments are available at https://github.com/tapilab/asonam-2019-sample

not require additional human interaction to obtain high quality samples.

C. Convert a classifier into search query

Finally, researchers wishing to collect more relevant data studied how to convert a classifier trained on labeled data into search queries [7, 8]. The differences between these techniques and the proposed approach in this paper is that these techniques mostly use information gain to pick top terms and they do not account for representativeness. Thus, these strategies might not be appropriate to collect data for observational studies. In particular, they might under perform in the presence of class imbalance.

III. Problem Definition

In this section, we first define the general problem we wish to solve and the evaluation measures of interest. We then introduce a framework to build a query and retrieve one or more documents relevant to the query.

A. General Problem

The problem setting we are interested in is as follows: suppose a researcher wishes to collect documents possessing a certain attribute; for example, Instagram comments containing hostility or tweets expressing mental distress. The researcher has access to some third-party API to search the universe of documents by keyword. The problem is to determine which queries to submit and in what order so that the researcher may obtain a large set of relevant documents that are representative of the population of all relevant documents. A key assumption is that the researcher does not have access to the entire universe of documents directly, which is most often the case for researchers working with online social media.

More formally, let $\textsc{Search}(q)$ refer to the API search function that takes as input a query q and returns a document r that matches q. After each search, assume we store the query and the returned document in the sets Q and R, respectively. Thus, after submitting N queries, all retrieved documents will be stored in R.

B. Quality measures

The quality of a querying strategy depends on the quality of the retrieved documents R. Given the full, unobservable population of documents $\mathcal{D} = \{d_1 \dots d_{|\mathcal{D}|}\}$, we define the following two measures.

1) Coverage: also known as recall or sensitivity, coverage [9] is defined as $\text{covg}_R = \frac{\sum_{(r_i,y_i)\in R} y_i}{\sum_{(d_j,y_j)\in \mathcal{D}} y_j}$ where $y \in \{0, 1\}$ indicate the relevance label for a document. $\text{covg}_R \in [0, 1]$ and higher values are better.

2) Representativeness: this measure focuses on how representative the words used in the retrieved documents are of the population of words used in all relevant documents. Let $\vec{x} = \{x_1 \dots x_k\} \in \mathbb{R}^k$ be a vector of word counts for a document over a vocabulary of size k. From the document population $\mathcal{D}$, we compute the multinomial distribution $P_{\mathcal{D}}(x_i \mid y = 1)$. We can estimate the analogous multinomial $P_R(x_i \mid y = 1)$ using only the word occurrences in the retrieved set of documents R. A natural way to measure the representativeness of the words in R with respect to D is to quantify the discrepancy between the two word distributions. We select the Hellinger distance [10] (noted H), as it is symmetric, bounded, obeys triangle inequality, and has been used previously to compare multinomials. We define our representativeness measure as the Hellinger similarity between two multinomial distributions noted $\text{repr}(P, Q) = 1 - H(P, Q)$. In this case, we define the representativeness between the two multinomial distributions described above $\text{repr}_R = \text{repr}(P_{\mathcal{D}}(x|y = 1), P_R(x|y = 1))$. Higher values of repr_R are better.

IV. Methods

Now that we have defined the problem and described the properties of good solutions, we next turn to different querying strategies. Each method implements a function $\textsc{CreateQuery}(\mathcal{U}, \mathcal{L}, R, Q)$, where $\mathcal{U} \subset \mathcal{D}$ is a large, unlabeled dataset sampled from the document population $\mathcal{D}$; $\mathcal{L} \subset \mathcal{D}$ is a small, labeled dataset containing tuples $(\vec{x}, y)$; Q is the set of previously issued queries; and R is the set of previously retrieved document.

A. Baselines

To evaluate the performance of our approach, we developed two natural baselines optimized for one of representativeness or coverage.

1) Baseline 1 - Random sampling: The first baseline only uses information from the unlabeled data $\mathcal{U}$. The goal is to retrieve documents at random with respect to the frequency of each term in the vocabulary. To do so, we estimate a multinomial $P_{\mathcal{U}}(x)$, which is simply the probability of each word x_i appearing in dataset $\mathcal{U}$. Each call to $\textsc{CreateQuery}$ samples keywords from $x_i \sim P_{\mathcal{U}}(x)$. Thus, we expect this baseline to have very poor coverage. In the experiments below, we refer to this baseline as B_1.

2) Baseline 2 - Most predictive words: A second strategy, designed to maximize coverage, is to somehow create a list of keywords that are likely to indicate a relevant document. This baseline only uses information from the labeled data $\mathcal{L}$, using standard text classification methods to identify words that are highly correlated with the relevant class. In the experiments below, we consider two variants: $B_2(1\%)$ selects the top 1% of words, while $B_2(10\%)$ selects the top 10%. We expect this approach to greatly increase coverage but have low representativeness.

B. Proposed Approach: Multi-objective query construction

The baselines described above present two extreme strategies. Our proposed approach aims to improve over the baselines by combining ideas from each baseline into a single objective. Additionally, our proposed approach updates its model after each query to refine its strategy.

At the core of our approach is a function $f(x)$ that scores the importance of including term x in the query under construction. To construct a one word query, we select the word

with maximum value of $f(x)$. To construct a multi-word query $q = \{x_1 \dots x_n\}$, we add one word at a time to the query, at each iteration updating components of $f(x)$ to re-rank the remaining words in the vocabulary. The function $f(x)$ is composed of three functional components: two for representativeness and one for coverage. We then combine these criteria using the geometric mean to get a unique score per word. Below, we will describe each of these components in turn, then explain how we combine them into a single objective.

1) Targeting high coverage with f_c: this component focuses on selecting words that will increase covg_R. To do so, we let $y = 1$ be the relevant class, q the query created, and (r, y_r) the document retrieved and its class. Thus, coverage will increase if $y_r = 1$. However, because the document r returned by SEARCH is not known at query creation time, we use q as a surrogate for r. Therefore, we define $f_{c,i}$ for every word x_i as $f_{c,i} = p_{\mathcal{L}}(y = 1|x_i)$ where $p_{\mathcal{L}}(y = 1|x_i)$ is a classifier trained on $\mathcal{L}$.

2) Targeting high marginal representativeness with $f_{r(\vec{x})}$: this component puts more weight on words that will raise the marginal representativeness. Using the representativeness definition from Section III-B2, we define the marginal representativeness as: $\text{repr}_{\vec{x}}(Q) = \text{repr}\left(p_{\mathcal{L}\cup\mathcal{U}}(\vec{x}), p_{R\cup Q}(\vec{x})\right)$ where $p_{\mathcal{L}\cup\mathcal{U}}(x_i)$ is the probability of word x_i to appear in either the labeled or unlabeled datasets, and $p_{R\cup Q}(x_i)$ is the probability of word x_i to appear in one of the queries created by CREATE-QUERY or one of the documents retrieved by SEARCH. Then, given the current state of sets Q and R, we can estimate that $\text{repr}_{\vec{x}}$ would increase by $\delta_i\text{repr}_{\vec{x}} = repr_{\vec{x}}(Q \cup q_i) - repr_{\vec{x}}(Q)$ if we were to add query q_i containing only the word x_i to Q. $\delta_i\text{repr}_{\vec{x}}$ will be the largest for words that are least accurately represented by $R\cup Q$ compared to $\mathcal{L}\cup\mathcal{U}$. Therefore, for every word x_i in the vocabulary, we define the second component $f_{r(\vec{x}),i} = \delta_i\text{repr}_{\vec{x}}$.

3) Targeting high class conditional representativeness with $f_{r(\vec{x}|y)}$: this third factor focuses on high class conditional representativeness. It adopts a similar strategy as $f_{r(\vec{x})}$ to increase representativeness individually for each class. For brevity, we do not include the details of this component in this version but they are covered in the longer version of this paper.

4) Combining the three components to select the best next keyword: we take advantage of the geometric mean attributes to simply combine these three factors in one final importance weight $w_i = \sqrt[3]{f_{c,i} \times f_{r(\vec{x}),i} \times f_{r(\vec{x}|y),i}}$ for each word. Finally, once w_i is computed for every word in the vocabulary, we use a greedy strategy by adding word x_i with the maximum associated weight w_i to the current query. We can then update the two factors accounting for representativeness before restarting the process to select the next word.

V. DATA

In this section, we describe the three datasets collected to run the experiments of this paper. We encode each document as a vector of word occurrences using unigram features only.

a) Online harassment [11]: This dataset contains 1,134 Instagram posts and 30,987 comments that were manually labeled by Amazon Mechanical Turk workers as being hostile comments (positive class) or not (negative class). Around 4,000 comments are labeled as positive (13.2%); for the purpose of our experiments, we consider each comment as an individual document, ignoring the thread structure.

b) Twitter smoking cessation: This is a newly collected dataset on smoking cessation, containing 12K smoking-related tweets. A tweet was manually labeled as positive if the author displays some intent to quit smoking; otherwise it is labeled as negative. Approximately 2,000 tweets (18.3%) are labeled as positive, making this dataset the one with the highest proportion of positive documents.

c) 20 newsgroups: The 20 newsgroups dataset is commonly used to evaluate natural language processing methods. It consists of approximately 20,000 documents covering 20 different topics from computer graphics to atheism. For the purpose of our experiments, we binarize the topics and create a new annotation such that each document is annotated with a positive label if it belongs to the `misc.forsale` topic (5.2%) and with a negative label otherwise.

To create the three disjoints datasets required by our query method, we shuffle each dataset and then split it in shares of 10, 30, and 60% of the original dataset. These subsets are then assigned to $\mathcal{L}$, $\mathcal{U}$, $\mathcal{D}$, respectively.

VI. EXPERIMENTS AND RESULTS

In Section III-B, we proposed to evaluate the quality of the documents R retrieved by a query method using a coverage measure covg_R and a representativeness measure repr_R. In this section, we use these two measures to compare the performance of our proposed approach in Section IV-B with two baselines defined in Section IV-A. For the retrieval model, we use the Okapi BM25 scoring function. In our experiment, we consider the four CREATEQUERY methods described in Section IV. We then run COLLECTDATA for each query method and over each dataset with $N = 20,000$ queries and we retrieve one document per query. Figure 1 shows the evolution of coverage in the first column and representativeness in the second column for each dataset.

On the first two datasets, the proposed approach successfully outperforms the best baseline after any number of queries. After 20,000 queries, MOQuery provides 10% more coverage and around 5% more representativeness than B1 on the online harassment dataset. Similarly , it outperforms B1 by a few points on the smoking cessation dataset. However, coverage and representativeness on the 20 newsgroup dataset are both dominated by B2. . Figure 1e indicates that a large portion of the positive instances in the 20 newsgroups dataset can be described using only the 10% most predictive words in the vocabulary. However, when this is not true (e.g. other datasets), this method dramatically under performs compared to B1 and MOQuery. This is expected as B2 does not account at all for representativeness and therefore is limited when exploring the keyword space. Thus, when the class concept is very simple,

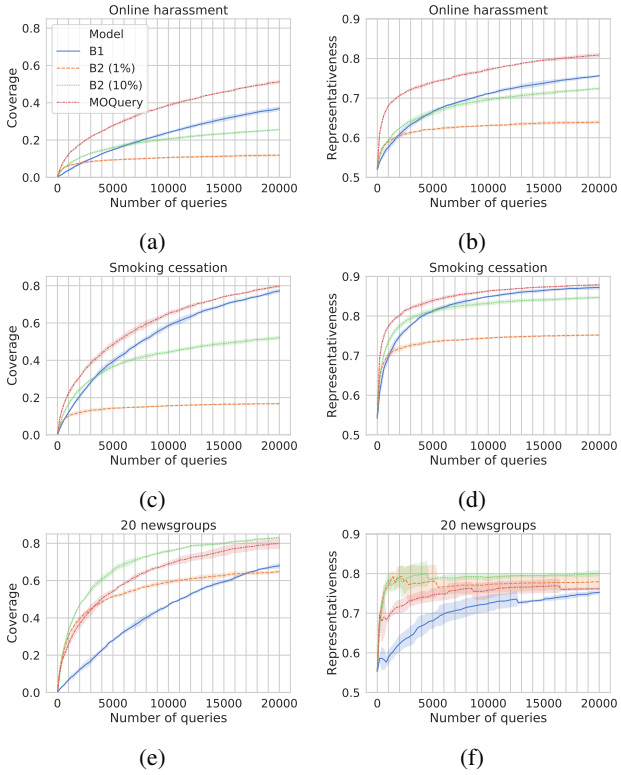

Fig. 1: Comparison of the coverage (a, c, and e) and representativeness (b, d, and f) of the four query generation methods.

the B2 baseline performs well; when the class concept is more complex, MOQuery can greatly improve both coverage and representativeness.

Finally, we note that B1 achieves relatively higher representativeness and coverage on the smoking cessation dataset than on the other datasets. This is due to the small size of this dataset (12K instances) and the relatively large ratio of positive instances (approx. 18%). This behavior indicates that in case of a less imbalanced dataset, B1 might be sufficient to collect enough instances of the rare class.

Overall, the proposed approach tops the baselines in relevance and coverage measures for all but one dataset in which its performance is only slightly lower than B2.

VII. Conclusion

In this paper, we investigated the overall question: how should we query a search engine in order to gather a large set R of relevant documents given user-specified attributes? We then proposed a measure of **coverage** and a measure of **representativeness** to evaluate the quality of the set of documents R retrieved by a query method. We introduced two baseline methods: one targeting coverage and the other targeting representativeness. Then we proposed a multi-objective approach combining three components to build a query that improves both coverage and representativeness. Finally, we compared this approach to the baselines on three real-world datasets displaying class imbalance, and we showed that the proposed approach outperforms the baselines on both coverage and representativeness.

Acknowledgments

This research was funded in part by the National Science Foundation under awards #IIS-1526674 and #IIS-1618244.

References

[1] A. Olteanu, O. Varol, and E. Kıcıman, "Distilling the outcomes of personal experiences: A propensity-scored analysis of social media," in *Proc. of The 20th ACM Conference on Computer-Supported Cooperative Work and Social Computing*, 2017.

[2] T. Cunha, I. Weber, and G. Pappa, "A warm welcome matters!: The link between social feedback and weight loss in/r/loseit," in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 1063–1072.

[3] F. Diaz, M. Gamon, J. M. Hofman, E. Kıcıman, and D. Rothschild, "Online and social media data as an imperfect continuous panel survey," *PloS one*, vol. 11, no. 1, p. e0145406, 2016.

[4] A. Olteanu, C. Castillo, F. Diaz, and E. Kiciman, "Social data: Biases, methodological pitfalls, and ethical boundaries," SSRN Pre-print, 2016.

[5] S. Li, S. Ju, G. Zhou, and X. Li, "Active learning for imbalanced sentiment classification," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 139–148.

[6] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," in *Advances in neural information processing systems*, 2010, pp. 892–900.

[7] S. Zelikovitz and M. Kogan, "Using web searches on important words to create background sets for lsi classification." in *FLAIRS Conference*, vol. 1, 2006, pp. 298–603.

[8] A. Anagnostopoulos, A. Z. Broder, and K. Punera, "Effective and efficient classification on a search-engine model," in *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 208–217.

[9] P. P. Biemer and L. E. Lyberg, *Introduction to survey quality*. John Wiley & Sons, 2003, vol. 335.

[10] M. Nikulin, "Hellinger distance. hazewinkel, michiel, encyclopedia of mathematics," *Springer, Berlin. doi*, vol. 10, pp. 1 361 684–1 361 686, 2001.

[11] P. Liu, J. Guberman, L. Hemphill, and A. Culotta, "Forecasting the presence and intensity of hostility on instagram using linguistic and social features," *Ann Arbor*, vol. 1001, p. 48109, 2018.