Learning Sufficient Queries for Entity Filtering

Miles Efron, Craig Willis, Garrick Sherman

Graduate School of Library and Information Science, University of Illinois, Urbana-Champaign
{mefron, willis8, gsherma2}@illinois.edu

ABSTRACT

Entity-centric document filtering is the task of analyzing a time-ordered stream of documents and emitting those that are relevant to a specified set of entities (e.g., people, places, organizations). This task is exemplified by the TREC Knowledge Base Acceleration (KBA) track and has broad applicability in other modern IR settings. In this paper, we present a simple yet effective approach based on learning high-quality Boolean queries that can be applied deterministically during filtering. We call these Boolean statements *sufficient queries*. We argue that using deterministic queries for entity-centric filtering can reduce confounding factors seen in more familiar "score-then-threshold" filtering methods. Experiments on two standard datasets show significant improvements over state-of-the-art baseline models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Retrieval models

General Terms

Algorithms, Experimentation

Keywords

Document filtering, Entity retrieval, Boolean models

1. INTRODUCTION

Though document filtering is well-studied in the information retrieval (IR) literature, filtering tasks are seeing renewed interest today. In recent years, the TREC knowledge base acceleration (KBA), temporal summarization and microblog tracks have all run filtering tasks. These tasks differ from the earlier TREC filtering tasks, which focused on topical information needs. Contemporary filtering often concerns *entities* such as people, organizations or places. In this paper, we argue that entity-related filtering presents different profile-representation challenges than topical filtering. We propose an approach that is tailored to contemporary domains and shows strong effectiveness on two TREC collections.

Our core argument is that using training data to craft high-quality Boolean queries which are applied deterministically during filtering can be more effective than methods based on estimating a dissemination threshold on document scores. We propose shifting the prediction problem in filtering from a *document* classification task (emit/ withold) to a *feature* classification task (include/exclude this feature from our query).

The paper's main contribution is an approach to query construction for entity filtering. We present an algorithm that includes terms in a Boolean query if they improve the ability to filter training data correctly. While this is a very simple approach, it shows strong experimental effectiveness on two TREC collections.

2. ENTITY-CENTRIC FILTERING

This paper is concerned with the problem of entity-centric filtering, as exemplified by the KBA track's cumulative citation recommendation (CCR) task [6, 12]. The goal of CCR is to monitor an incoming document stream and send appropriate documents to editors of a knowledge base such as Wikipedia. In this context, each entity corresponds to a Wikipedia page, say `Phyllis_Lambert` or `Red_River_Zoo`. CCR systems route documents containing "edit-worthy" information about the entity E to the editors of the E node in the knowledge base.

While KBA is our motivating example, the need for this type of filtering arises in other domains such as social media, where users might wish to follow information about particular people, or companies might like to track discussions of their brand [12].

But is entity-centric filtering qualitatively different from earlier TREC filtering tracks or related areas of topic detection and tracking (TDT)? One important difference is the availability of a surface-form match on the entity's name[1]. In TREC KBA 2012, out of 7,806 "central" (i.e. true-positive) documents in the training corpus, zero failed to include a surface-form match on the entity name [6]. This is in contrast to topics from earlier TREC filtering tasks such as *falk-*

[1] Throughout this paper, the *surface-form* representation of an entity is simply the title of its Wikipedia page, with punctuation and disambiguation metadata removed. Several of the 2013 KBA entities were represented by Twitter accounts instead of Wikipedia pages. These entities' surface form is the `name` element associated with their Twitter account.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609517.

land petroleum exploration, where relevant documents may contain only some query terms, with no guaranteed inter-term proximity or ordering. It has been noted [7] that few CCR runs at TREC 2013 achieved an F1 score higher than one gets by running a simple `egrep` on each entity's surface-form representation over the corpus. But looking for an exact match on queries in earlier filtering collections gives very poor performance. This disparity suggests that something is indeed different between older filtering tasks and CCR.

In the remainder of this paper, we propose a framework for exploiting this difference. We develop highly accurate Boolean queries for each entity in the CCR task. This is in contrast to more familiar filtering approaches where documents are scored against a topic profile, then emitted if their score exceeds an empirically determined threshold. Our approach capitalizes on what is easy about entity-centric filtering–the near-guarantee of surface-form matches in relevant documents–while avoiding what is hard about all filtering tasks: defining a document scoring function and an associated dissemination threshold.

2.1 Vocabulary and Notation

Let E be an entity that a user would like to track using a CCR system. We define the moment at which the user specifies E to be time t_0. All documents available to the system prior to t_0 are available for training. After t_0 the CCR system runs without user feedback (test phase).

We assume that at time t_0, the user labels $m \geq 0$ documents with respect to their relevance to E. These labeled documents comprise the training set $\mathbf{T} = (T_1, T_2, \ldots, T_m)$. The system may use $\mathbf{T}$ to inform subsequent decisions.

During the test phase, documents reach the system sequentially, as a stream. When the system encounters a document D_i, it must immediately make a decision: emit the document, or do not emit. The decision to emit implies that D_i is relevant to E.

3. APPROACHES TO CCR

Score-then-Threshold: The scenario described above is identical to the setup of the earlier TREC filtering tasks [11], particularly batch filtering. A strategy developed for the earlier tasks and still used in KBA today is what we call *score-then-threshold* (STT). An STT system estimates θ_E, a profile for E. The system also relies on a scoring function $\varphi(\theta_E, D_i)$ (e.g. the KL divergence between language models, BM25, etc.) whose scores ostensibly correlate with document relevance. Finally, an STT system defines a threshold τ. If $\varphi(\theta_E, D_i) > \tau$ the system emits D_i, otherwise it does not. Typically τ is estimated by optimizing some accuracy measure such as F1 over $\mathbf{T}$. Discussions of STT approaches include [3, 10, 11].

We argue that much of the difficulty in STT-based filtering arises because systems must tackle at least three problems:

1. Profile estimation
2. Document scoring
3. Threshold estimation.

Handling these three tasks simultaneously has proven very difficult, especially with respect to the CCR task, where sophisticated approaches failed to outperform a simple surface-form match on entity names.

Sufficient Queries: Given the difficulty and weak performance of STT-based strategies, we propose a novel approach to the CCR task: filtering via *sufficient queries*:

> **Definition 1**: *Sufficient Query.* A Boolean query with enough breadth and nuance to identify documents relevant to an entity E without further analysis or estimation.

For an entity E, the "sufficient query application" method (SQA) to filtering involves defining a sufficient query Q_E, and then simply applying Q_E to all incoming documents, emitting those that evaluate to true with respect to Q_E. Because a sufficient query is expressed as a Boolean criterion, no document scoring or thresholding is necessary.

For an entity such as *Phyllis Lambert*, a sufficient query must cast a wide enough net to capture a large proportion of relevant documents. A simple match on the surface-form query $S =$ `#1(phyllis lambert)` does this[2].

But a sufficient query must also constrain the set of retrieved documents in order to reduce the number of false positives; since Phyllis Lambert is a relatively common name, a sufficient query must discriminate between the intended person (i.e. the architect from Montreal) and all other people with that name. Additionally, the query must filter documents that contain the bigram `phyllis lambert` but that do not rise to the level of relevance.

Our goal is to elaborate on the surface-form query S to improve effectiveness. Of course many changes to S might accomplish this. For simplicity, we rely on a single strategy. For entity E, we emit document D iff it:

- contains a match on S, the surface-form query for E.
- matches any of k additional features, $(f_1, f_2, \ldots, f_k)$, where $k \geq 0$.

For example:

```
#band(#1(phyllis lambert)
  #syn(architect montreal canada))                    (Q1

#band(#1(phyllis lambert)
    #syn(#1(canadian architect) #1(public art)))   (Q2
```

conform to this structure. Query Q1 requires a document to match the quoted phrase *"phyllis lambert"* and to contain at least one of the terms: *architect, montreal* or *canada*. Query Q2 has the same structure, but relies on two bigrams to refine the reach of S. More systematically, we build queries that consist of two parts:

- **Constraint Clause**: The surface-form query, S.
- **Refinement Clause**: A set of 0 or more features (unigrams, bigrams, etc.).

The constraint and refinement are combined via a Boolean AND. In general, the queries we propose using as deterministic document filters have this form:

`#band(` S `#syn(` $f_1, f_2, \ldots, f_k$`))` (Q3

for the constraint S and refinement clause `#syn(` $f_1, f_2, \ldots, f_k$`)`. In entity retrieval S will usually accumulate many relevant documents, but it is probably too broad. The refinement clause shrinks the size of the overall retrieved set. Because members of the refinement clause are treated as an equivalence class, as their number grows the overall query will tend to revert back to the breadth of S.

4. LEARNING SUFFICIENT QUERIES

As described above, for an entity E, the CCR task defines a set of m labeled training documents $\mathbf{T}$. Let $\mathbf{F}$ be the set

[2] We express example queries using the Indri query language. All queries generated during experiments are available at `http://timer.lis.illinois.edu/sigir-2014`.

of "features" that we will consider adding to our constraint clause. In this paper we define $\mathbf{F}$ as the set of word bigrams in relevant training documents[3]. Thus, all of our estimated queries take the form of Q2 above.

When building a query for SQA-based filtering, we wish to find the best features in $\mathbf{F}$ to include in the refinement clause. For this, we use the Bayes decision rule:

$$\text{log-odds}(f, \mathbf{T}) = \log \frac{P(T|f)P(f)}{P(T|f')P(f')} \tag{1}$$

where $P(T|f)$ is the probability that a training document in $\mathbf{T}$ is correctly classified, given that we add feature f to the refinement, and $P(T|f')$ is the probability of a correct classification if we exclude f. We include all features f_i where $\text{log-odds}(f_i, \mathbf{T}) > 1$.

To estimate Eq. 1, we simply have:

$$\hat{P}(T|f) = \frac{n(T^+|f)}{m} \tag{2}$$

where $n(T^+|f)$ is the number of correctly classified training documents if we include f in the query. $P(T|f')$ is calculated analogously, replacing the numerator in Eq. 2 with the number of correct classifications when f is omitted.

The factor $P(f)$ in Eq. 1 encodes the knowledge that some features are, a priori, better topical discriminators than others. However, without such knowledge readily available, we let $P(f) = P(f')$. This yields the simple decision rule that we include a feature f if it improves classification accuracy on $\mathbf{T}$ over a query that lacks f.

To make estimation tractable, we treat each candidate feature $f_i \in \mathbf{F}$ in isolation when computing Eq. 1. Thus, $P(T|f')$ is calculated from the surface-form query, while $P(T|f)$ derives from a query requiring a surface-form match AND the presence of f. This is not globally optimal, as non-independencies among features surely exist. However, without this simplification, the search space during query building is intractably large.

5. EXPERIMENTAL EVALUATION

To test the effectiveness of the queries generated by the method described above, we performed the CCR task over two data sets: the TREC KBA collections from 2012 and 2013. Summary statistics about these collections appear in Table 1. Full details about pre-processing of the data is available in [5]. However, very little pre-processing was done: no stemming, and a stoplist applied at query time[4].

Table 1: Collection Statistics for 2012 and 2013 KBA CCR Tasks.

	KBA 2012	KBA 2013
Num. Docs	~400 M	~1 B
Num. Entities	28	141
Median Training Docs	684	45
Median Rel. Training Docs.	51	12

We compared five approaches to the CCR task, which we enumerate in Table 2. Gray rows indicate that a method uses the score-then-threshold strategy. Blue indicates a Boolean approach. The row labeled SQ-2 (sufficient queries, 2-grams) is our sufficient query approach, and as such, is our main point of interest. The SF (surface-form) run simply emits any document that contains a surface-form match on the entity. The STT-Base run is a "pure" score-then-threshold run, where every document containing a unigram from the surface-form query was scored. We include STT-Base for completeness, but its effectiveness was low. For more realistic STT benchmarks, we include Base and RM3. These runs are similar to many TREC KBA submissions. In both approaches, documents are initially filtered on S. Matching documents are then scored and thresholded. So although we label them as STT, they are in fact hybrid methods. Base and RM3 differ only in entity profile representation. Base uses unigram features that comprise S to score and threshold. RM3 represents each entity by a relevance model [9] estimated from the training documents.

Table 3: Filtering Effectiveness Statistics.

	KBA 2012			KBA 2013		
	F1	Prec.	Recall	F1	Prec.	Recall
STT	$0.182^{\downarrow\blacktriangledown}$	$0.162^{\downarrow\blacktriangledown}$	$0.387^{\uparrow\blacktriangledown}$	$0.062^{\downarrow\blacktriangledown}$	$0.073^{\downarrow\blacktriangledown}$	$0.198^{\downarrow\blacktriangledown}$
Base	0.251	$0.274^{\downarrow\blacktriangle}$	$0.350^{\uparrow\blacktriangledown}$	$0.243^{\blacktriangledown}$	$0.243^{\downarrow}$	$0.406^{\uparrow\blacktriangledown}$
RM3	0.268	$0.321^{\blacktriangle}$	$0.273^{\blacktriangledown}$	$0.231^{\blacktriangledown}$	$0.288^{\blacktriangledown}$	$0.285^{\blacktriangledown}$
SF	0.261	$0.199^{\downarrow}$	$0.666^{\uparrow}$	$0.309^{\uparrow}$	$0.238^{\downarrow}$	$0.820^{\uparrow}$
SQ-2	$0.280^{\blacktriangle}$	$0.222^{\downarrow\blacktriangle}$	$0.596^{\uparrow\blacktriangledown}$	$0.316^{\uparrow}$	$0.252^{\blacktriangle}$	$0.737^{\uparrow\blacktriangledown}$

Statistically significant outcomes are shown in Table 3 as follows. Improvements (declines) with respect to RM3 are shown with ↑ (↓). Improvements (declines) with respect to SF are shown by ▲ (▼). Significant changes imply $p < 0.05$ using a paired, one-tailed t-test.

Though it is not surprising to see the shifting balance of precision and recall across different methods in Table 3, several noteworthy results are evident from the data.

1. With respect to F1 (the official metric of the KBA track), sufficient queries give very strong effectiveness, improving over all other methods.

2. Sufficient queries temper the decline in precision that a SF match incurs over STT approaches.

Overall, Table 3 suggests that sufficient queries perform well. Their F1 score on the 2013 data exceeds the best reported official TREC run, and the 2012 score is approximately the median among systems that use a large array of features instead of the limited text-only strategy used here.

However, a fair question is whether the benefit of SQ-2 is due to an unintended artifact; perhaps the low recall of the STT methods is due to a systematic over-estimation of the dissemination threshold, a defect that Boolean methods overcome not by better estimation but simply by virtue of being broad. Maybe instead of tuning Boolean queries, we can (using a loosely Bayesian flavor) simply lower STT's dissemination thresholds via:

$$\hat{\tau} = \lambda \tau_{train} + (1 - \lambda) min\_score \tag{3}$$

where τ_{train} is the optimal cutoff given the training data, and min_score is the lowest document score on the query among training documents, and $\lambda \in [0, 1]$. As λ decreases, the dissemination threshold drops, allowing more documents to pass through the filter. The parameter sweeps shown in

[3] The choice of bigram features was due to analysis that space constraints force us to omit.

[4] A copy of the stoplist we used is available at `http://timer.lis.illinois.edu/sigir-2014`.

Table 2: CCR Approaches used in Effectiveness Comparisons. Row colors correspond to two major types of run: score-then-threshold (STT, gray) or simple Boolean (blue).

Name	Type	Description
STT	STT	Documents scored by negative KL divergence from a query model induced from the unigram in entity name (Dirichlet smoothing, $\mu = 2500$). Threshold is cutoff that optimizes F1 over the training data.
Base	STT	Identical to STT, except documents are only analyzed if they contain a surface-form match on the entity name.
RM3	STT	Relevance model. Identical to Base, except that each entity is represented by a linear combination of the original query and a relevance model estimated from its judged relevant documents in the training data .
SF	Boolean	Simple `egrep` . The surface-form query is used as a Boolean filter.
SQ-2	Boolean	Sufficient queries estimated as in Section 4. Bigram refinement clause features.

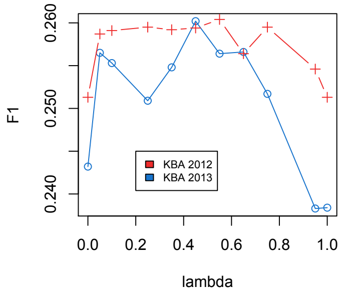

Figure 1: Parameter Sweeps of Mixture Model Coefficient for Lowering Emit Thresholds in Base Runs.

Figure 1 suggest that lowering the STT threshold does help, but not to the extent that we see with sufficient queries.

We hypothesize that the advantage here is due to the flexibility of SQ-2. The breadth of SQ-2 queries varies widely from entity-to-entity. The mean number of bigrams added by SQ-2 on the 2012 data was 88.3 (s.d. 22.6), with a mean of 46.345 (s.d. 46.2) in 2013. Occasionally, SQ-2 will add 0 features to a model. Figure 1 suggests that this flexibility is more effective than a wholesale decision to emit more documents.

6. CONCLUSION

The analysis presented here supports our core hypothesis: well-crafted Boolean queries can be effective filters for entity-based tasks such as CCR. A *sufficient query* is a Boolean query that is broad enough to allow the whole range of relevant documents to evaluate to true, while offering enough constraint to maintain reasonable precision. The method for building sufficient queries that we proposed in Section 4 yielded models that were highly effective in an experimental setting. This approach obviates the need to estimate dissemination thresholds common in other filtering approaches and shifts the prediction problem from a *document* classification task to a *feature* classification task, which we argue is more tractable in the presence of adequate training data.

Our approach gives an expanded query optimized over training data. Thus it is a form of relevance feedback. Feedback and expansion techniques in document filtering have a long history [1, 11]. But our work is closer in spirit to supervised methods that generate powerful query features from noisy data (e.g. [4, 8, 2]). We see our contribution as a natural extension of this research area.

In future work we plan to improve our ability to estimate sufficient queries. Several important directions include: adapting queries during the course of a filter's execution, integrating sufficient query-based scores into state-of-the-art machine learning approaches, and extending the applicability of sufficient queries to tasks other than entity-based filtering.

7. ACKNOWLEDGEMENTS

This work was supported in part by the US National Science Foundation under Grant No. 1217279 Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] J. Allan. Incremental relevance feedback for information filtering. In *Proc. of SIGIR'96*, pages 270–278, 1996.

[2] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proc. of SIGIR '08*, pages 491–498, 2008.

[3] J. Callan. Learning while filtering documents. In *Proc. of SIGIR '98*, pages 224–231, 1998.

[4] G. Cao et al. Selecting good expansion terms for pseudo-relevance feedback. In *Proc. SIGIR '08*, pages 243–250, 2008.

[5] M. Efron et al. The Univ. of Illinois' Grad. School of Library and Information Science at TREC 2013. In *The 22nd Text REtrieval Conference*, 2013.

[6] J. R. Frank et al. Building an Entity-Centric Stream Filtering Test Collection. In *TREC 2012*, 2012.

[7] J. R. Frank et al. Evaluating stream filtering for entity profile updates for trec 2013. In *TREC-2013*, Forthcoming.

[8] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *Proc. of SIGIR '09*, pages 564–571, 2009.

[9] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proc. of SIGIR '01*, pages 120–127, 2001.

[10] S. E. Robertson. Threshold setting and performance optimization in adaptive filtering. *Inf. Retr.*, 5(2-3):239–256, Apr. 2002.

[11] S. E. Robertson and I. Soboroff. The trec 2002 filtering track report. In *TREC 2002*, 2002.

[12] M. Zhou and K. Chang. Entity-centric document filtering: boosting feature mapping through meta-features. In *Proc. of CIKM 2013*, pages 119–128, 2013.