The University of Illinois' Graduate School of Library and Information Science at TREC 2014

Garrick Sherman, Miles Efron, Craig Willis
Graduate School of Library and Information Science
University of Illinois, Urbana-Champaign
501 E. Daniel St., Champaign, IL 61820
gsherma2, willis8, mefron@illinois.edu

October 28, 2014

1 Introduction

The University of Illinois' Graduate School of Library and Information Science (uiucGSLIS) participated in TREC's Federated Web (FedWeb) and Knowledge Base Acceleration (KBA) tracks in 2014. Specifically, we submitted runs for the FedWeb resource selection and KBA cumulative citation recommendation (CCR) tasks.

2 FedWeb Resource Selection

The Federated Web Search (FedWeb) resource selection task (RS) requires participants to rank candidate search engines, known as resources, according to the applicability of their contents to test topics. The use-case is that a user U issues an ad hoc query Q to a federated web search system S with access to a set of resources $R = \{R_1, R_2, ..., R_n\}$. Given that S cannot efficiently submit Q to each resource R_i, the system must rank the resources in order of their appropriateness to Q so that only the most applicable resources are searched.

2.1 Experimental Data

2.1.1 The FedWeb 2014 Dataset

The FedWeb 2014 Dataset contains both result snippets and full documents sampled from 149 web search engines between April and May 2014. Snippets contain document title, description, and thumbnail image when available. Our system did not make use of snippets, instead focusing on the full text documents sampled. Documents are available in their original format; our system indexed content only from textual documents. Resources are classified into "verticals," which describe the topic, media type, or genre of resources. Though classification of resources into verticals was available, our system did not make use of them.

2.1.2 Topics

A set of 75 test topics was made available, of which 50 were ultimately used for system evaluation. The 50 evaluation topics were not made known at submission time, so runs were submitted with results for all 75 topics. Topics were in the form of ad hoc keyword queries, as are typical in web environments.

2.2 Base System

As with the KBA CCR task, we used the Indri search engine and API for core indexing and retrieval in both FedWeb runs. We did not apply a stop list or stem documents.

2.3 Submitted Runs

We submitted two runs to the FedWeb RS task this year. The first, `uiucGSLISf1`, ranked each resource by its query clarity [1] score for the topic. The second, `uiucGSLISf2`, ranked resources by a CF-IDF (collection frequency-inverse document frequency) score. These techniques are described in more detail below.

2.3.1 Query Clarity

The `uiucGSLISf1` run computed the well-known query clarity score for each query-resource pair and ranked resources by decreasing clarity for each topic.

The query clarity score is a measure of a query's ambiguity with respect to a document collection and is defined as the KL-divergence between the query and collection language models:

$$S(Q,R) = \sum_{w \in V} P(w|Q) log \frac{P(w|Q)}{P(w|R)}$$

where w is a word in the vocabulary V, $P(w|Q)$ is the probability of the word in the query language model, and $P(w|R)$ is the probability of the word in the overall resource language model.

We treated each resource as a collection composed of the documents sampled for that resource. The relevance model was computed removing stopwords, with 20 feedback documents and 10 feedback terms.

The motivating intuition underlying this approach is as follows. If a query is applicable to a resource, the quality of the query language model computed from that resource should be high. High quality query language models are assumed to be those that minimize the query's ambiguity. If a resource is not appropriate for a query, it is unlikely to promote documents that cohere to a single topic and will therefore lead to a query language model that is very similar to the overall resource language model. If a resource *is* appropriate for a query, it will tend to promote relevant documents and lead to a query language model that diverges from the overall resource language model.

2.3.2 CF-IDF

The `uiucGSLISf2` run used a score computed from query terms' collection and inverse document frequencies (CF-IDF) to rank resources' appropriateness for each query. These quantities are computed as expected. The collection frequency of a word is defined as

$$CF(w) = c(w, R)$$

where $c(w, R)$ is the count of the word w in the resource. The inverse document frequency is

$$IDF(w) = log\frac{d(R)}{d(w, R)}$$

where $d(R)$ is the number of documents in the resource and $d(w, R)$ is the number of documents in the resource that contain w. Putting these together, the CF-IDF of a query can be computed as

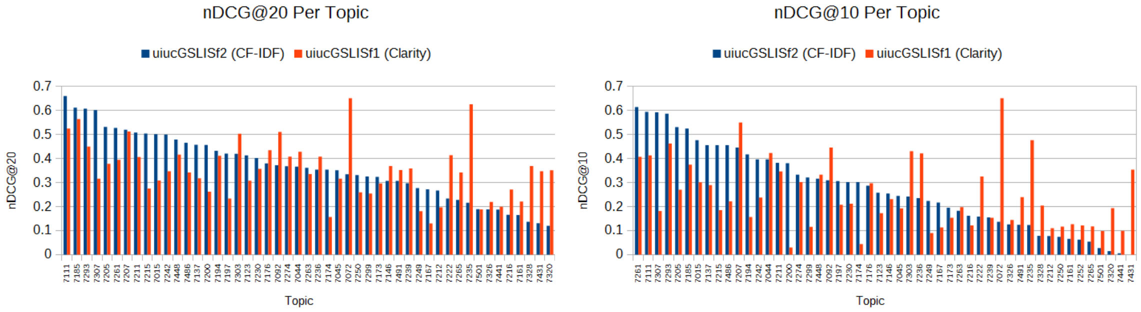
$$CF\text{-}IDF(Q) = \sum_{q \in Q} CF(q) \times IDF(q).$$

Intuitively, the CF component of this formulation is intended to capture to what degree the resource is "about" the query; that is, how often the resource includes content relating to the query. This mirrors the TF component of TF-IDF but spread across all documents in a resource. The IDF component is intended to penalize ambiguity of a query for a resource, in that the larger the proportion of documents in a resource matching a query, the less confident we can be that the documents match in a meaningful way. The decision to mix collection-level (CF) with document-level (IDF) was pragmatic: because resources are so large, most resources contain most query terms, reducing the discriminative power of a collection-level IDF factor. Use of document-level IDF allowed us to approximate a term's discriminative power on a per-resource basis.

The intuition underlying CF-IDF, then, approximates the intuition underlying the query clarity score. Both seek to reward resources containing a set of documents that are strongly on-topic in comparison with the resource as a whole. That is, both prefer resources that minimize the ambiguity of the query.

Run Name	nDCG@20	nDCG@10	nP@1	nP@5
uiucGSLISf2	0.361 (+/- 0.140)	0.274 (+/- 0.172)	0.179 (+/- 0.222)	0.213 (+/- 0.164)
uiucGSLISf1	0.348 (+/- 0.117)	0.249 (+/- 0.138)	0.101 (+/- 0.207)	0.212 (+/- 0.193)

Table 1: Results of Official GSLIS FedWeb RS Runs. Runs are shown in decreasing order of nDCG@20. Standard deviation across topics per run is in parentheses.



(a) nDCG@20

(b) nDCG@10

Figure 1: Per-topic nDCG@20 and nDCG@10 for both FedWeb RS runs. Runs are ordered by decreasing CF-IDF score.

2.4 Analysis of Results

Table 1 summarizes the results of our runs for all official metrics. It shows that the CF-IDF run outperformed the query clarity run on all measures.

Figure 1 depicts nDCG@20 and nDCG@10 per topic for both runs. As these charts suggest, the Pearson correlation between the two runs is quite low: 0.3884 for nDCG@20 and 0.3407 for nDCG@10. These results demonstrate that, despite their shared motivating intuition to promote resources that minimize query ambiguity, the CF-IDF and query clarity approaches perform quite differently when applied to the same topic.

2.5 Conclusion and Future Directions

More investigation is needed to determine why the CF-IDF and query clarity runs differ so strongly in their performance on each topic. This will involve identification of differences between the two approaches (e.g., query expansion occurs in query clarity but not CF-IDF), common properties among topics that strongly favor one approach or the other, and commonalities among the resources that either approach might favor. Establishing the runs' respective strengths and weaknesses will help in the design of more effective techniques for the future.

3 KBA CCR

The Knowledge Base Acceleration (KBA) cumulative citation filtering task (CCR), also referred to as "vital filtering", is a stream-oriented document filtering task. The use-case is that a user U is an editor of a node T for an entity in a knowledge base. Given an incoming time-ordered stream of documents $\mathcal{D}$, the system must decide whether to recommend each document D_i to U. The "vital" criteria means that the document would "motivate a change to an already up-to-date knowledge base article." In other words, our goal is to monitor $\mathcal{D}$, signaling to U when we find a document that contains "edit-worthy" information regarding the entity T.

3.1 Experimental Data

3.1.1 The 2014 KBA Stream Corpus

The KBA "Stream Corpus" is a collection of timestamped documents published on the web between October 2011 and January 2013[1] from nine different sources including news, social medial, blogs, forums, linking services, and scholarly publications. This year we used the filtered subset[2] which consists of approximately 20 million documents. The stream corpus includes additional metadata and derived data (e.g., part-of-speech tags) that were not used by our system.

3.1.2 Target Entities

The CCR task involves monitoring the stream corpus for documents that contain vital information about any of a set of 109 target entities. Each entity is associated with a URL representing a profile in an imaginary knowledge base. This year the entities also included an external profile URL that was not used by our system. This year, systems were evaluated on a subset of 71 target entities that contained sufficient training data.

3.1.3 Training Data

Track organizers provided a distinct training time range (TTR) for each entity. Teams were permitted to analyze documents from before the end of the TTR period for training purposes. Only those entities with sufficient training data were used for evaluation.

3.1.4 Annotation Data

The CCR task in 2014 recognized two levels of relevance with respect to an entity-document pair: *useful* and *vital*. A vital rating indicates a stronger usefulness of the document than

[1] `http://trec-kba.org/kba-stream-corpus-2014.shtml`

[2] `http://s3.amazonaws.com/aws-publicdatasets/trec/kba/index.html`

a useful rating does. The official goal of the task was to maximize F1 based on these vital ratings.

3.2 Base System

For core indexing and retrieval we used the Indri search engine and API[3]. Very little preprocessing was used in our experiments: a standard stop list was used for all tasks. In general we did not stem documents, although stemming was used in one of our models (textttverb, described below).

All documents are filtered using a preliminary proximity query based on an ordered window of length 2. Documents that match the proximity query are then scored based on the criteria outlined below.

For assessing entity-document similarity, we used the negative KL-divergence between the language model θ_i for document D_i and the language model θ_E for the entity E [2]:

$$sim(D_i, E) = -\sum_{f \in E} P(f|\theta_E) \log \frac{P(f|\theta_E)}{P(f|\theta_i)}. \tag{1}$$

where f is a "feature" of the profile we defined to represent E. Usually, some feature f_j is the simply a term that is highly associated with E.

Additionally, $sim(D_i, E)$ is combined with the supplemental features listed in Section 3.4 below. This yields our final score for D_i against E:

$$score(D_i, E) = sim(D_i, E) + f_1(D_i) + f_2(D_i) \ldots \tag{2}$$

Conceptually, $f_i(D_i)$ plays the role of a prior over documents. However, use of KL divergence for measuring similarity does not lend itself to the proper introduction of a prior (unlike, say, query likelihood).

The decision to emit `true` for D_i is based on the magnitude of $score(D_i, E)$ with respect to a threshold τ. We emit `true` iff $score(D_i, E) \geq \tau$.

In all experiments, τ is set simply by scoring all training documents according to Eq. 2 and finding the cutoff that maximizes the F1 score with respect to the training annotations. When finding the optimal cutoff, F1 is calculated using only vital documents as positives.

3.3 Entity Representation

This year, we used two different representations of each target entity:

- `sf`: The basic surface form of the entity name
- `rm3`: Relevance model [3] interpolated with the surface form ($\lambda = 0.5$) using true relevance feedback for vital documents from the training period for each entity. If a document was judged by multiple assessors, the highest score was used (`any-up`)

[3] `http://lemurproject.org`

Feature	**Abbrev**	**Description**
Length	`length, len`	Probability this document is vital for this entity given the length.
Source	`source, src`	Probability this document is vital given the source.
Verb	`verb`	Probability this document is vital for this entity given the verb language model.
Burst	`burst`	Probability this document is vital for this entity given the amount of time between it and the previous document.
Previous docs	`prevdocs, pd`	Probability this document is vital for this entity given the number of documents in the last 24 hours.

Table 2: Features evaluated in CCR task

3.4 Features

Our main research goal this year was to improve CCR effectiveness by combining scores based on purely lexical information with additional features. We focused on document attributes including length and source, temporal features, and a verb-based language model. The features and a brief description are listed in Table 2.

Formally, let R_T be the set of documents labeled as *vital* in the training data. Let D_i be the ith document. Let $S(D_i)$ be the source of D_i.

3.4.1 Document length

As demonstrated last year, vital documents tend to be well-behaved with respect to length. This year we explored the effect of document length conditioned on source. We used the training data to find the length of all vital documents for each source.

If we let $L(D_i)$ be the length of document D_i, we hypothesized that $L(D) \in R_T$ would follow a log-normal distribution $\mathcal{N}_{L,S}$. Using R_T, we estimated the mean and standard deviation of $\mathcal{N}_{L,S}$ for each source S by maximum likelihood, such that

$$\ell(D_i) = \mathcal{N}_{L,S}(\log n(D_i), \hat{\mu}_s, \hat{\sigma}_s) \tag{3}$$

where $n(D_i)$ is the length of D_i, and $\hat{\mu}_s, \hat{\sigma}_s$ are the estimated parameters (mean and standard deviation) of the log-normal distribution for source S.

3.4.2 Document source prior

We expect that some document sources will have a higher prior probability of producing vital documents than others. Given each source S and set of vital documents V in the training data, we calculated $P(S_i)$ by maximum likelihood:

$$P(S) = \frac{n(V, S)}{n(V)}$$

Where $n(V, S)$ is the number of vital documents from source S and $n(V)$ is the total number of vital documents. For each document D, the prior probability of the document $P(D)$ is estimated using this value. In runs using this model, log $P(D)$ was added to the negative KL divergence score of each document.

3.4.3 Temporal features

We also explored two sources of temporal information: the previous number of documents (`prevdoc`) in a specified interval (e.g., 48 hours) and the amount of time between the current document and the previous document (`burst`).

The `prevdocs` feature models the probability that the current document is vital for the current entity given the number of documents seen about the current entity in specified interval P_d for the current source S. Each source will likely have a different "tempo" by which documents enter the stream. We used the training data to calculate the probability of a vital judgment given the number of previous documents published in the last interval for the current document source. We hypothesize that this follows a Poisson distribution and estimated the mean over the training data:

$$P(V|P_d, S_i) \sim \ Poisson(\lambda = 7)$$

Similar to the previous number of documents, the `burst` feature uses the amount of time between the current document and the most recent previous document. The intuition here is that vital documents will be published in "bursts" and that the interval between documents about an entity will be shorter when events are happening to that entity. Given the time of the current document t_i and the time of the previous document t_{i-1} and a specified interval (e.g., 24 hours), we model the distribution of differences as a normal distribution with $\hat{\mu} = 0$ and $\hat{\sigma} = 2$.

3.4.4 Verb language model

Because the vital filtering task indirectly rests on changes that target entities undergo over time, we hypothesized that documents whose vocabulary includes a high proportion of words indicative of change might have a higher prior probability of relevance than documents that, for instance, simply enumerate many peoples' names or that include a target entity's name in an off-hand way. To operationalize this, we focused on verbs, particularly verbs that often convey changes in the state of affairs. We thus built a language model V where we estimate

$$P(w|V) = \frac{n(w, H)}{n(H)}$$

Where H are the verbs in the headlines of the Aquaint corpus. We extracted all article headlines from the Aquaint corpus (approximately 1 million headlines) and compared each token with the list of verbs included in the current WordNet distribution. The Porter stemmer was used for all work with verbs. This yielded 4786 verbs from the Aquaint headlines. The quantity $n(w, H)$ is simply the number of headlines in which the word w occurs. We then applied the verb model to generate a document prior for each item in the streamcorpus. Given a document D, we define:

$$P(D) = z \cdot logit\left[\sum P(w|D) \cdot P(w|V)\right]$$

where z is a normalizing constant. In runs using the verb model, log $P(D)$ was added to the negative KL divergence score of each document.

3.5 Submitted Runs

Simple combinations of features ($2^6 = 64$) were tested on the KBA 2013 test collection using both the surface form (sf) and relevance model (rm3) entity models. The 6 best feature combinations for each entity model were used for the KBA 2014 submissions.

This year we submitted 14 runs. Results of these runs are shown in Table 3. We submitted two broad classes of runs:

- sf: Runs where the entity model is simply the surface form entity name.
- rm3: Runs where the entity model is the RM3 over the high-vital training data.

The names of the runs shown in the table are intended to be self-explanatory given the abbreviations listed in Table 2.

3.6 Analysis of Results

Table 3 summarizes the results of our runs. In general, no systematic differences were found.

Significance was determined using a paired t-test with $\alpha = 0.05$ on the precision, recall, and scaled utility measures. Of the two baselines, unsurprisingly the basic surface form entity representation was significantly better than the true-RM3 model in terms of recall. The pdsrc_rm3 model was significantly better than the baseline_rm3 model in both precision and scaled utility. The srclen_rm3 and verbsource_rm3 models were significantly better than the baseline_rm3 mode in both recall and scaled utility. However, none of these models were significantly different from the simple surface form baseline.

Figure 2 presents a comparison of the length, sourceless, and verb source runs to the respective sf and rm3 baselines by F1. The baseline is represented by the solid black line with queries ordered on the x-axis by the descending value of the baseline measure (e.g., precision). Each run is plotted in red over the baseline.

Run Name	Precision	Recall	F1	SU
sourcelen_sf	0.3637	0.5746	0.4455	0.3526
verbsource_sf	0.3679	0.5447	0.4392	0.3257
length_sf	0.3833	0.4885	0.4296	0.3786
baseline_sf	0.3607	0.5037	0.4204	0.3499
pdverb_sf	0.3533	0.4466	0.3945	0.3518
pdsrc_sf	0.3321	0.4727	0.3901	0.3382
verbsource_rm3	0.3773	0.4029	0.3897	0.3505
sourcelen_rm3	0.3526	0.4288	0.3870	0.3496
pdverb_rm3	0.3623	0.3876	0.3745	0.3620
prevdocs_sf	0.3261	0.4179	0.3663	0.3570
length_rm3	0.3914	0.3411	0.3645	0.3872
prevdocs_rm3	0.3384	0.3834	0.3595	0.3573
pdsrc_rm3	0.2962	0.4183	0.3468	0.3063
baseline_rm3	0.3998	0.2798	0.3292	0.4160

Table 3: Results of Official GSLIS KBA CCR Runs. Vital Only. Runs are shown in decreasing order of F1.

3.7 Conclusion and Future Direction

Our investigation of document length and source priors as well as the temporal features and verb-based language model demonstrated little or no effect on vital filtering performance when compared to the simple surface-form baseline. Figure 2 suggests significant improvements for some entities (and degradation for others). Future work will focus on the analysis of individual entities to improve our understanding of why these methods work or fail in these cases.

After submitting our official runs, we noticed that the document sources had an unusual distribution in the 2014 stream corpus. Figure 3 displays the temporal distribution of documents by source. As can be seen in from the figure, sources or source labels were apparently changed midway during the stream collection process. This means that the "social" and "News" sources occur only in the first half of the collection and the "WEBLOG" and "MAINSTREAM_NEWS" sources appear in the second half. This likely has a significant effect on the use of a source prior, since little or no training data will be available for some entities for some sources. Normalizing the source labels would likely have a positive effect.

References

[1] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 299–306, New York, NY, USA, 2002. ACM.

[2] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '01*, pages 111–119, 2001.

[3] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '01*, pages 120–127, 2001.

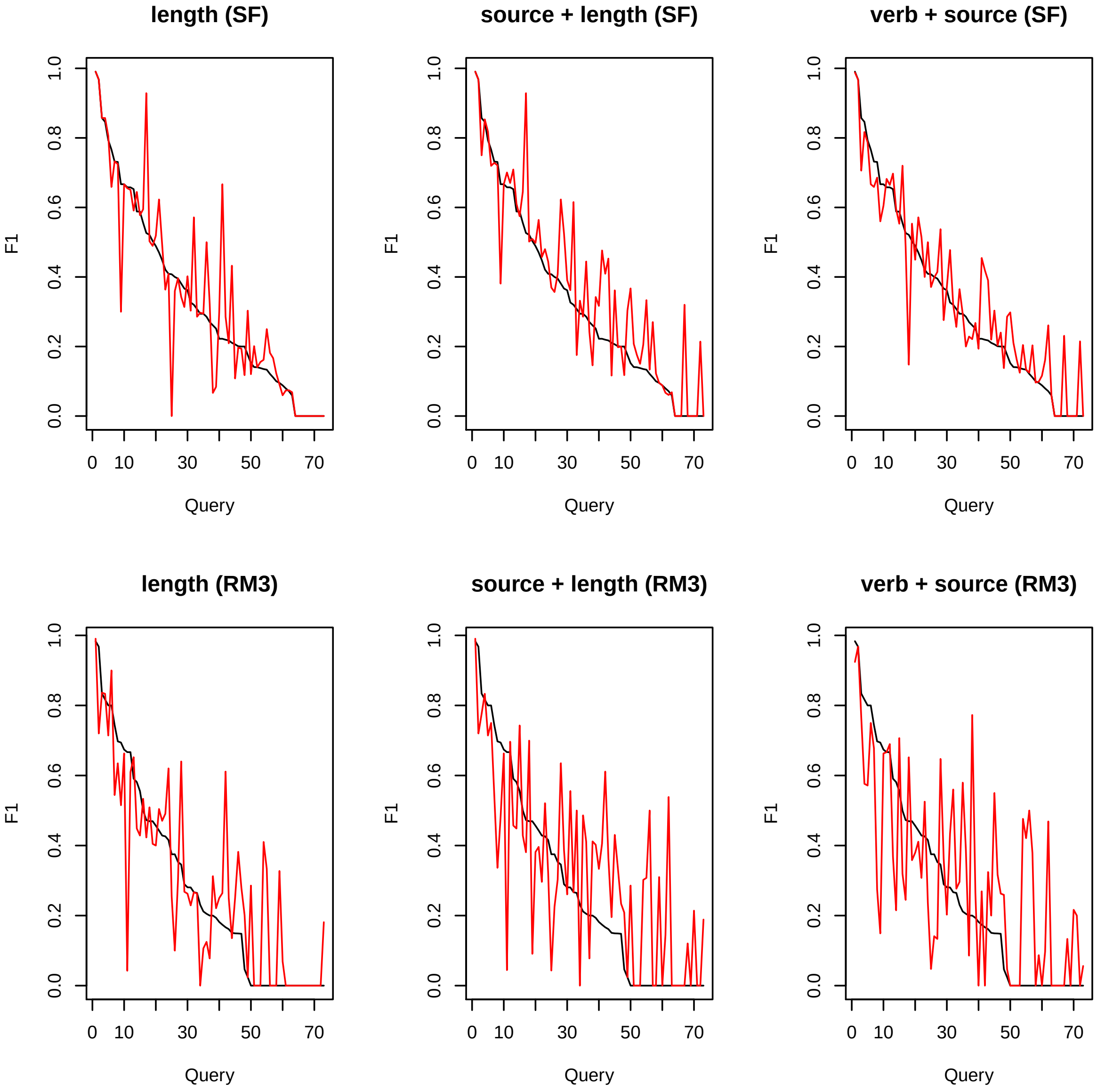


Figure 2: Comparison of length, source+length, and verb+source runs (red) to baseline (black) by F1

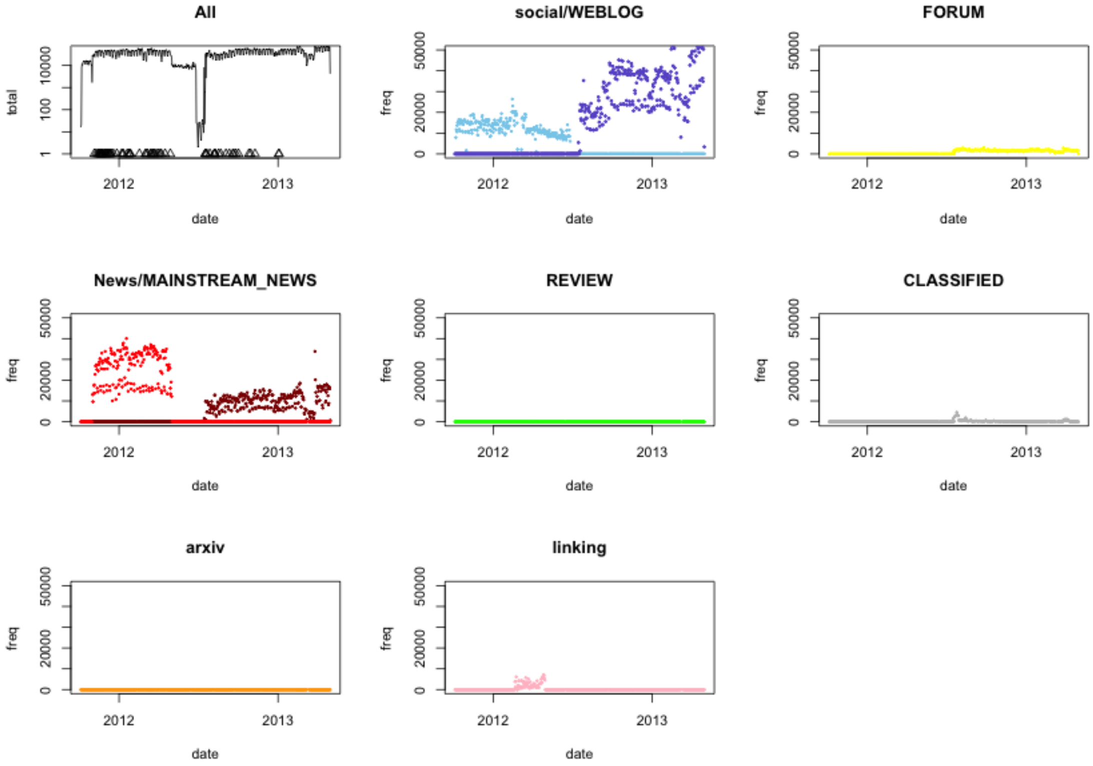


Figure 3: Temporal distribution of results in KBA 2014 filtered corpus by source