Evaluating Stream Filtering for Entity Profile Updates for TREC 2013
(KBA Track Overview, Notebook Paper)

John R. Frank[1], Steven J. Bauer[1],Max Kleiman-Weiner[1], Daniel A. Roberts[1], Nilesh Tripuraneni[1],
Ce Zhang[2], Christopher Ré[2],
Ellen Voorhees[3], Ian Soboroff[3]
1 KBA Organizers, Massachusetts Institute of Technology, jrf@mit.edu
2 Stanford University and University of Wisconsin
3 National Institute of Standards and Technology Gaithersburg, MD ian.soboroff@nist.gov

Abstract

The Knowledge Base Acceleration (KBA) track in TREC 2013 expanded the entity-centric filtering evaluation from TREC KBA 2012. This track evaluates systems that filter a time-ordered corpus for documents and slot fills that would change an entity profile in a predefined list of entities. We doubled the size of the KBA streamcorpus to twelve thousand contiguous hours and a billion documents from blogs, news, and Web content. We quadrupled the number of entities as query topics from structured knowledge bases (KB), such as Wikipedia and Twitter. We also added a second task component: identifying entity slot values that change over the course of the stream. This Streaming Slot Filling (SSF) subtask focuses on natural language understanding and is a step toward decomposing the profile update process undertaken by humans maintaining a knowledge base. A successful KBA system must do more than resolve the meaning of entity mentions by linking documents to the KB: it must also distinguish **vitally relevant** documents and **new slot fills** that would change a target entity's profile. This combines thinking from natural language processing (NLP) and information retrieval (IR). Filtering tracks in TREC have typically used queries based on topics described by a set of keyword queries or short descriptions, and annotators have generated relevance judgments based on their personal interpretation of the topic. For TREC 2013, we selected a set of filter topics based on Wikipedia and Twitter entities: 98 people, 19 organizations, and 24 facilities. Assessors judged ~50k documents, which included ***all*** documents that mention a name from a handcrafted list of surface form names of the 141 target entities. Judgments for documents from before February 2012 were provided to TREC teams as training data, and the remaining 12 months of data was used to measure the F_1 accuracy and scaled utility of these systems. We present peak macro-averaged F_1 scores for all run submissions. High scoring systems used a variety of approaches, including simple name matching, names of related entities from the knowledge base, and various types of classifiers. Top scoring systems achieved F_1 scores in the mid-30s. We present results for an oracle baseline system that scores in the low-30s. We discuss key lessons learned at the end of the paper.

Categories & Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – Information Filtering; H.3.m [Information Storage and Retrieval]: Miscellaneous – Test Collections; I.2.7 [Natural Language Processing] Text analysis – Language parsing and understanding
General Terms: Experimentation, Measurement

Introduction

This overview paper describes the goals of the TREC Knowledge Base Acceleration (KBA) track and its relation to other evaluations, the data generated by and for the track, and the evaluation metrics, assessment analysis, and results. TREC KBA is a stream filtering task focused on entity-level events in large volumes of data. Many large knowledge bases, such as Wikipedia, are maintained

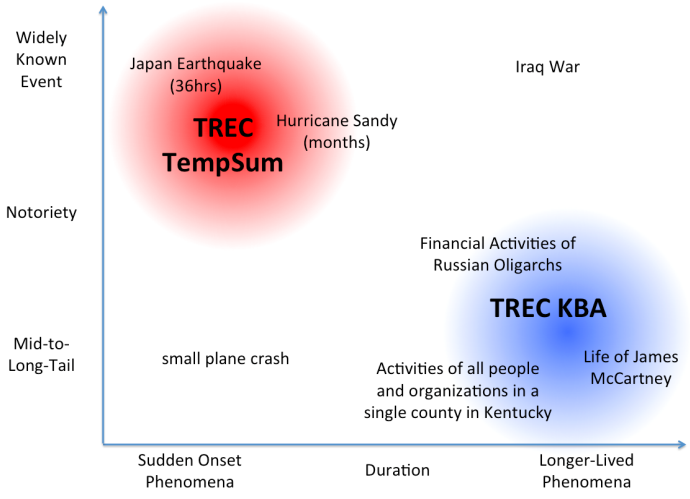
by small workforces of humans who cannot manually monitor all relevant content streams. As a result, most entity profiles lag far behind current events. KBA aims to help these scarce human resources by driving research on automatic systems for filtering streams of text for new information about **entities**.

Entities are a special subset of general topics. Entities have strongly typed attributes and relationships, such as a name, birth date, father, hometown, employer, profession, etc. By restricting attention to entities as queries, instead of more general topics, IR systems can use structured properties of the entities gathered from knowledge bases, and can also leverage the extensive research in natural language processing (NLP) for gathering signals from texts.

NLP algorithms are often developed using test data sets that are significantly smaller than real-world data sets, e.g., on the Internet. IR often deals with large data sets; however, IR often focuses on simple statistics instead of entity-oriented signals. The streaming context is a crucial element of real world systems. Nonetheless, many research efforts ignore streaming by focusing on static data sets.

TREC KBA addresses these issues with a novel testbed for streaming, entity-centric IR research. One of KBA's contributions is the 1bn document StreamCorpus, which spans 17 months (11,948 hours) of news, blog, and Web content.

In this year's TREC, we piloted a second task, Streaming Slot Filling (SSF), and also piloted an interface with the Knowledge Base Population "Cold Start" task in the Text Analytics Conference.



Filtering at TREC 2013

Mining large streams of unstructured data is a key area of IR research. The time series nature of the data and the volume of content provide a rich landscape for characterizing end users' needs, designing retrieval systems, and quantifying performance and quality.

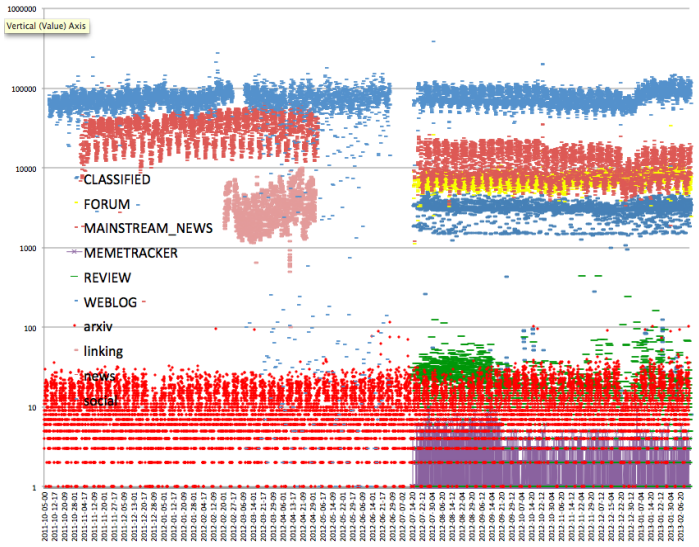
Several TREC tracks evaluate systems on stream-oriented data, including KBA, Microblog, and Temporal Summarization (TS). TS and KBA require systems to simulate decision-making as documents are processed in temporal order. This *online* decision-making and evaluation imposes different constraints on algorithms. TS focuses on large-scale event-type entities with many sub-events and filtering of unstructured nuggets. KBA focuses on updating a large semi-structured knowledge base, in which the long tailed distribution of entity mentions tends to make it difficult for humans to keep the profiles up-to-date with new attributes and relationships of active entities.

Corpus

For 2013, we expanded the KBA StreamCorpus to over a billion documents by adding new data donated by Spinn3r. The arxiv preprint service also donated their full text to the corpus.

	2012	2013
Corpus	7 months (4,973 hours) >400M documents 40% English Oc 2011 to Apr 2012.	17 months (11,948 hours) [1] >1B documents 60% English or unknown Oct 2011 to Feb 2013
Queries	27 people 2 organizations all from Wikipedia	98 people, 19 organizations, and 24 facilities. Fourteen inter-related communities of entities, such small towns like Danville, KY, and Fargo, ND, and academic communities like Turing award winners.
Assessing	70% agreement on "central"	3198 hours have >0 vitals 76% agreement on "vital" (replaced "central")
Submissions	11 teams, 40 runs	13 teams, 140 runs
Metrics	F_1, Scaled Utility	F_1, Scaled Utility [2]

Figure 2: document counts per hour for each "source" type in the KBA StreamCorpus v0_2_0



CCR Assessing

We updated the assessing guidelines for TREC 2013 by refining the highest relevance criterion to: would this document change an already up-to-date profile? This "vital" rating replaced the "central" rating of 2012. The second rating level was renamed from "relevant" to "useful," and includes documents that are citation-worthy as background information that might be used when writing an initial dossier but do not present timely or "fresh" changes to the entity. This boosted inter-assessor agreement from 70% to 76% over TREC KBA 2012 -- a 20% gain. This removed a large area of subjectivity in the notion of "citation-worthiness" by requiring assessors to imagine they have an already up-to-date profile. The assessing process continues to require several forms of subjective judgment, including deciding exactly what "already up-to-date" should mean for particular documents and entities.

Tempo: The difference between an vital update and background biographical info can be subjective in several ways. One particular aspect of subjectivity involves judging whether an event that changed the entity is being mentioned way after the fact. For example, a text might explain when an entity was born -- fifty years after the fact. Such reporting is obviously biographical (therefore "useful") and not sufficiently timely to be vital. Borderline examples exist. For example, should we consider this passage timely? "Sara helped start NXIVM in 2007." (reporting date 2011)

No, that's not timely enough. That's useful, not vital. If the posture of the reporting language is itself biographical, it is even easier to discern: "In the past, Sara had helped start NXIVM -- around 2007." (Reporting date not needed to judge as useful, not vital.)

Assessors must also decide what to include in an entity's profile. For example, for a person with a Wikipedia article, a profile might include why the person is noteworthy. If the person is less noteworthy, the profile might simply describe how they spend their time. This is typical of Twitter entities.

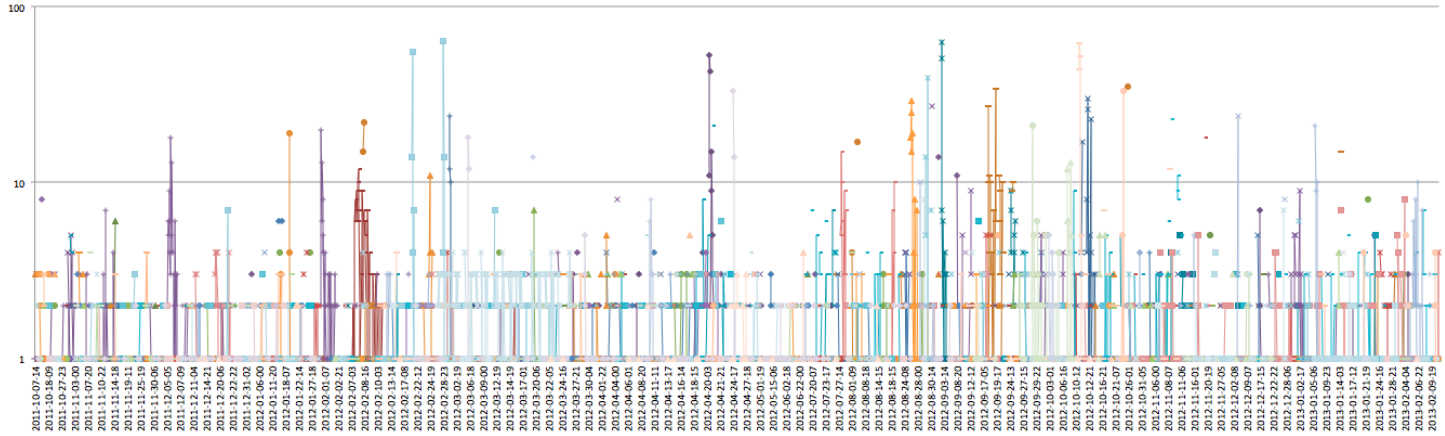


Figure 3.: KBA 2013's 141 query entities and their vital document counts per hour. Several spikes are visible. Most spikes are echos in the blogosphere that reverberate after a single event, which is often characterized by changes to a few strongly typed entity attributes or relationships, such as death or the breakdown of a corporate or spousal relationship.

CCR Metrics (see SSF Metrics below)

The metrics for CCR are F_1 and Scaled Utility(SU) and are shown in Figure 4. No system achieved an SU over 0.333, which corresponds to a run with no output. The F_1 score is the harmonic mean of the macro-averaged precision and macro-averaged recall. In this context, macro-averaging means using the confidence cutoff for which the F_1 is highest for the system under study, and summing the precision (or recall) scores from all of the query entities and dividing by the number of entities. Some of track participants have invented a time-sensitive metric [5].

CCR Oracle Baseline Systems

We considered several baseline systems to characterize the task. The official baseline assigns a "vital" rating to every document that matches a surface form name of an entity and assigning a confidence score based on the length of the observed name. See code in github [6]. This achieves scores of (macro-P = 0.190, macro-R= 0.824, F1 = 0.310, SU = 0.157) This is shown in the official score plots below as "oracle baseline." We call it an "oracle" because it used a hand-picked set of surface form names to have high recall.

Motivated by spikes visible in Figure 3, we also considered a time-clustering heuristic for a second baseline system. The heuristic uses two parameters (cluster-width and time window "cutoff"). This system assigns two documents to the same cluster if they are within one time window cutoff of each other, and assigns "vital" rating to all skinny (or fat) clusters, and "useful" for all fat (or skinny) clusters and clusters containing a single document. Clusters are defined to be skinny if the time-lag between first and last document in a cluster is less than the cluster-width, and fat if the time-lag is greater. By optimizing for F_1-score over the range (burst-width, cut-off) = (1:1:24, 1:1:24), using a skinny-burst algorithm with (burst-width=20, cutoff=4) achieves (P=0.206, R=0.736, F1=0.322 , SU=0.157) and using a fat-burst algorithm with (burst-width=1, cutoff=24) achieves

(P=0.191, R=0.602, F1=0.290, SU=0.159). Similarly optimizing for scaled utility over an identical range results in for a (burst-width=1, cut-off=24) that (P=0.140, R=0.126, F1=0.132, SU=0.169) for the skinny-burst algorithm.

Modifying the 'skinny' time-clustering heuristic by removing garbage/neutral documents (auto-assigning documents within a cluster with a ground truth rating of 0 or -1 a rating of 0), we find optimizing over the range (burst-width, cut-off) = (1:1:24, 1:1:24) that a (burst-width=24, cutoff=6) achieves (P=0.290, R= 0.735, F1=0.416, SU=0.256). So, while time-cluster-filtering of all name-match documents performs roughly as well as assigning confidence scores based on matched name string length, this modification boosts F_1 to 42% by removing garbage and neutral documents, suggesting that time-cluster-filtering of echos may be useful in CCR. By "echo," we mean a series of redundant documents all repeating a single event. Often large echos result from a single vital change. This also underscores the fact CCR does not fully capture the desires of end-users who would prefer not to weed through large bursts of redundant documents.

CCR Results

Thirteen teams submitted 141 runs. Figure 4 ranks the teams by their highest scoring system using the maximum F_1 or maximum Scaled Utility using two different retrieval objectives. The primary ranking is F_1 on the vital-only retrieval objective. This is a very hard task, as illustrated by the plateau of high ranking systems with roughly the same score as the oracle baseline.

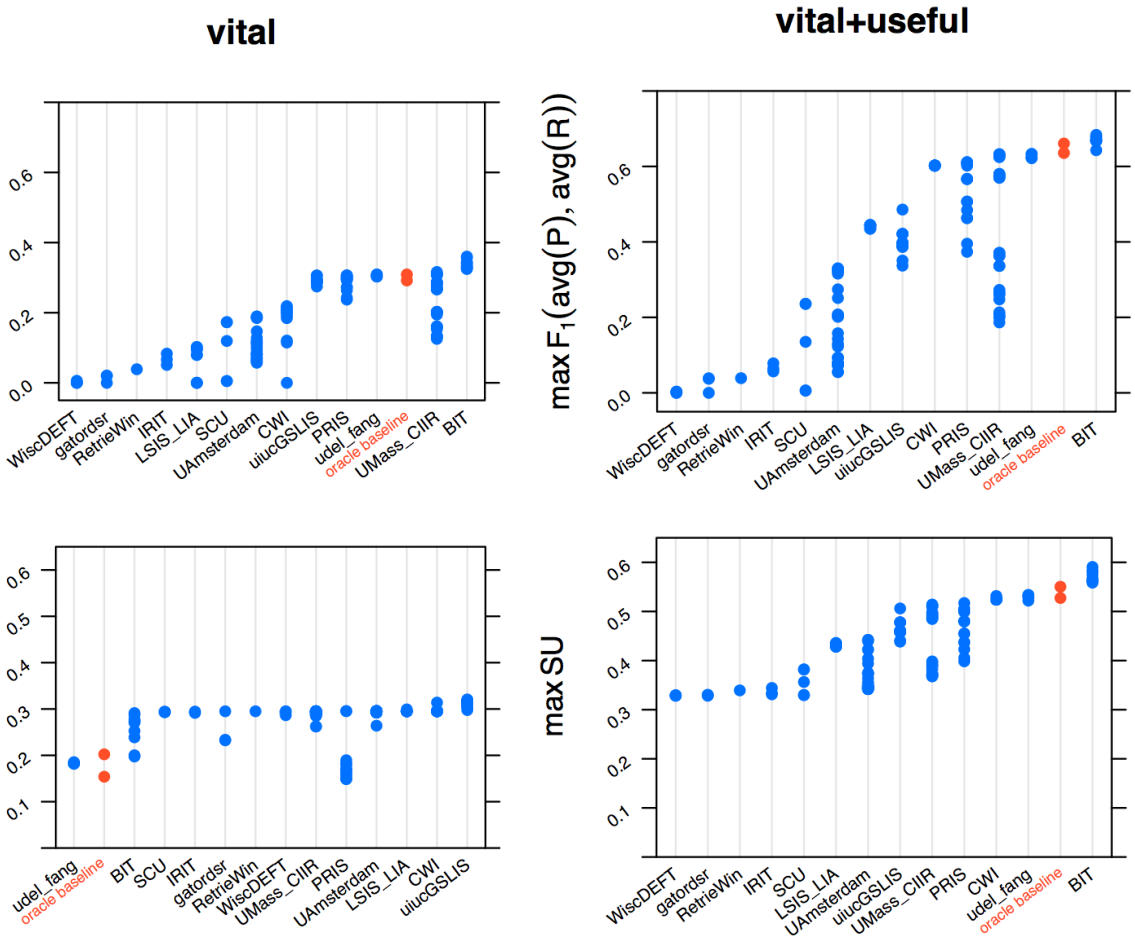


Figure 4: Official scores using "vital" and "vital+useful" as the classification objective. The red

points represent the oracle baseline system that used a hand-picked set of names to select a high-recall set of documents.

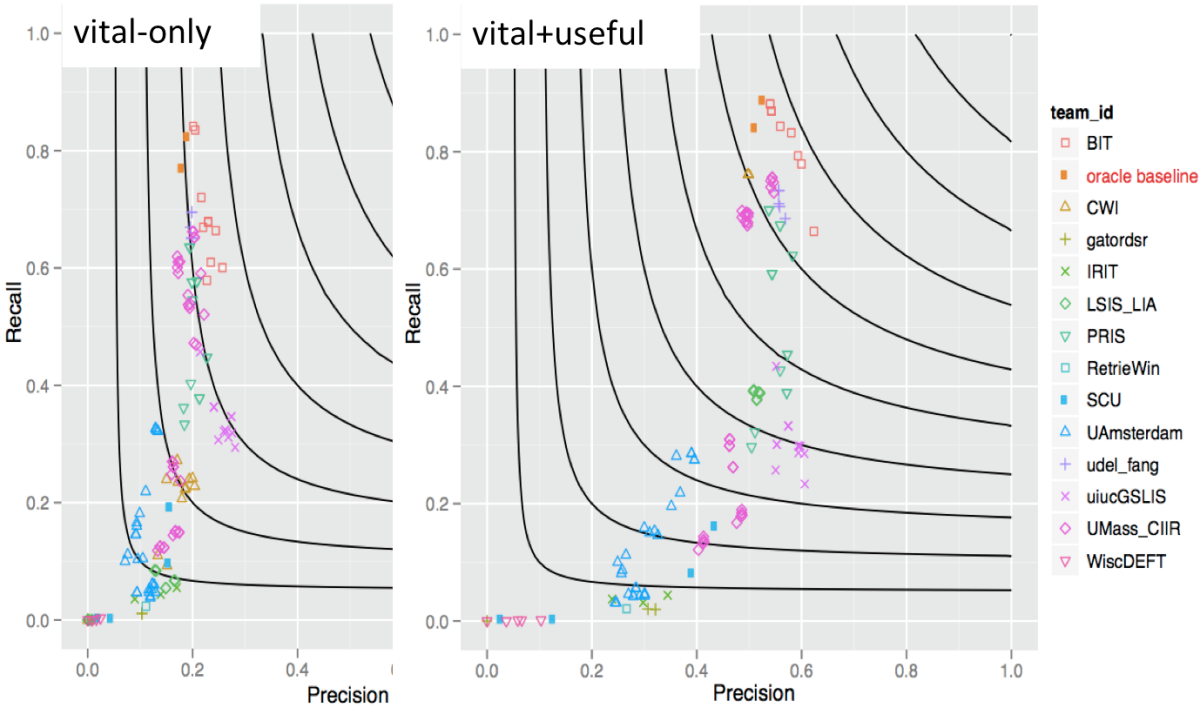


Figure 5: macro-averaged recall versus precision with curves of constant F_1 for "vital" CCR. The precision and recall values correspond to the maximum F_1 as a function of confidence cutoff.

Streaming Slot Filling (SSF):

SSF requires systems to ***detect changes to particular slots***. The goal of SSF is to explore the hypothesis that leveraging structured properties from knowledge bases is key to automatic detection of **entity-level events**. In studying CCR, many people realized that a large fraction of "vital" documents can be explained with a sentence of the form "The entity's _____ attribute acquired this new value: ____." In fact, it is an interesting research question to identify "vital" documents that do *not* fit this pattern. Any SSF run can be scored against the CCR objective. We anticipate that eventually, SSF systems should have the highest CCR scores.

The run submission format for SSF is the same as CCR with three more columns added: slot type, equivalence class, byte range. These fields allow systems to identify passages to fill slots from a fixed inventory of entity attributes, such as CauseOfDeath and the PlaceTime of a meeting. The equivalence class column allows systems to indicate when different passages substantiate the same slot fill. We selected a fixed inventory of slots from TAC-KBP [3] and ACE [4], plus four catch-all slots for Affiliate entity, AssociateOf (person-to-person), Contact_Meet_Entity, and Contact_Meet_PlaceTime.

As a motivating example, James McCartney caused a large echo in this BBC interview, which provides an answer to the question "what organizations is this entity starting?" (slot=founderOf or similar).

Figure 6: As an example, consider James McCartney, son of Paul McCartney, who confirmed in a BBC interview that he might start a new band called “Beatles II” or “The Beatles -- The Next Generation.” The large spike of echoes that resulted all contain redundant information. A good SSF system detects the change early. Example text: “Nothing may be sacred after all: Sir Paul McCartney's son James is interested in starting a second-generation Beatles band with John Lennon's son Sean, George Harrison's son Dhani and Ringo Starr's son Zak.”

SSF Assessing

To generate the ground truth data for SSF, we had assessors identify slot fills that *changed* during the time frame of the corpus. We identified over 500 distinct slot fills that appeared as new values during the time frame of the corpus for our 141 target entities. There were 387 distinct (entity, slot) pairs, and 2152 distinct fills. Of these fills, 470 begin during the time range of the corpus and are therefore the evaluation data. The remaining 1682 did not begin during and were released as training data. 48 ended during but did not begin during the corpus time range.

SSF Metrics

The simplest way to measure an SSF system is to ask whether it identified the slot *type* that is changed by a given document. This entry point is called “DOCS” and is displayed in the left-most plot of Figure 7. Next, we measured whether a system found an appropriate passage. This step alone is complicated, see the TREC Hard track. We treated this as a diagnostic and kept only the true positives from the DOCS stage to produce the OVERLAPS plot in Figure 7. With the overlap data, it is then possible for the scorer to equate equiv_id strings from the run and truth data, thus enabling computation of an F_1 score for coreference resolution of slot values in the FILL stage. For example, one entity became the Danville Chief of Police, which some documents reported as “Danville Police Chief.”

Finally, an ideal SSF system would also find the earliest text that substantiates the change with high confidence -- this is the DATE_HOUR stage in Figure 7.

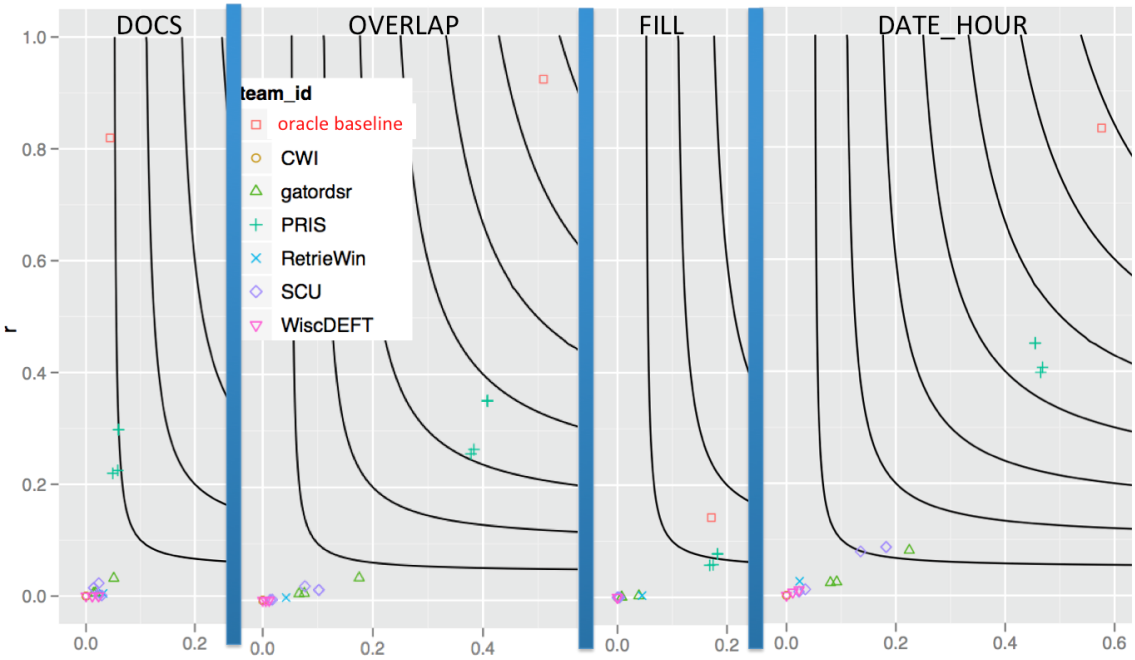


Figure 7: macro-averaged recall versus precision with curves of constant F_1 for the four stages of SSF scoring. Only the first level (DOCS) is a valid comparison across systems, because the

subsequent stages only consider an individual system's true positives from the previous stage. The high-recall of the oracle baseline is accomplished by asserting all possible slot types for every document that matches the surface form name of an entity. See the implementation of the baseline system in the github repository for the "toy_kba_algorithm.py"

The scores indicate that systems aggressively cut data from the stream, so much so that they achieve very little recall. This approach makes sense from the perspective that NLP technologies are often refined on much smaller and cleaner datasets, but in this case poor filtering renders the rest of the extraction pipeline useless.

The precision that they do achieve at that recall level is so low as to not have much influence on the overall F score. This is the core hardness of the task: systems have to identify what kind of change event is being reported in the document. The oracle baseline run picks the longest sentence that mentions the target entity, and proposes all possible event types for that entity type. This achieves a recall in excess of 0.8 and a precision of 0.044 (near the median.) The best team in the evaluation, PRIS, has a recall in the range 0.22–0.26, but with precision values just around the median of the other systems, 0.05–0.07.

To emphasize that improving recall is the critical next step, we can step back to the CCR task – finding vital *documents* – and compare the SSF systems to CCR systems at that core task. Figure 8 shows that the SSF runs are clustered in the lower left corner, with lower F than the CCR systems. We can hypothesize that if SSF systems start with a good CCR system as the first-pass filter, recall could be improved and SSF systems would show better overall performance.

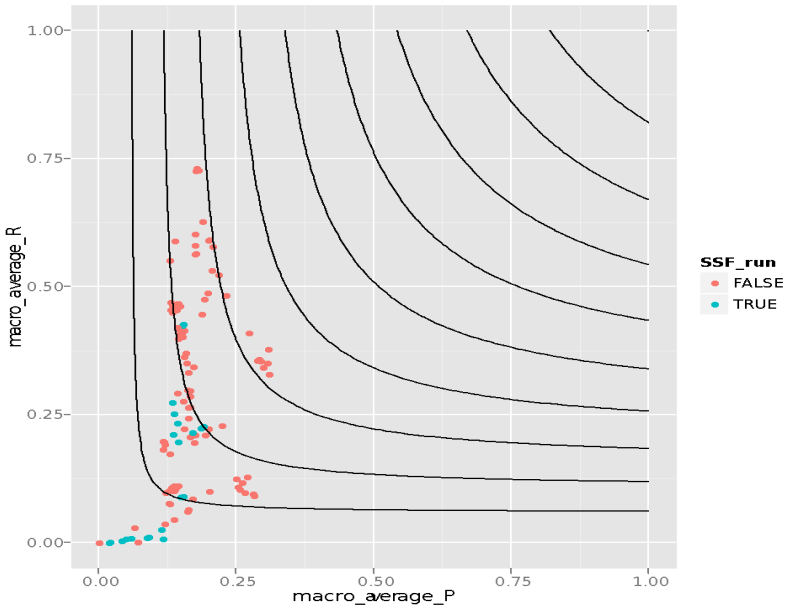


Figure 8: SSF system performance at the core CCR task, compared to CCR-specific systems.

SSF experimented with four "catch all" slots: Contact_Meet_Entity which is a superset including any event in which one or more entities (of any type) are present at the target facility. For example, the bold text is a Contact_Meet_Entity for the hyperlinked target_id "**The Senior Wellness Coalition of Fargo-Moorhead** will host a wellness seminar from 1 to 3 p.m. March 28 at the Hjemkomst Center, 202 1st Ave." Contact_Meet_PlaceTime is a catch-all slot for Persons. It is a superset including any event in which the target entity is present at a particular place at a particular time. Affiliate is a catch-all slot for relations between entities, and AssociateOf is specifically for person-person relations.

As expected, the SSF slots have a long tail in both dimensions: entities and slots. The long tail means that systems with very specifically written extraction routines tied to specific slot types could be miss slots with large numbers of instances.

Assessors attempted to focused their attention on slot fills that were likely to change. Nonetheless, about three-quarters of the couple thousand fills either changed before the corpus started or were hard to ascertain. The quarter that clearly began during the corpus are counted in Table 2. The assessors noted that they could often find reporting about a change to an entity for several months after the change, and identifying the earliest document for the change often required sleuthing through all of the mentioning texts. Future assessor tools could improve this.

num eval fills	num entities	slot type
232	50	Contact_Meet_PlaceTime
96	41	Affiliate
70	10	Contact_Meet_Entity
27	10	AssociateOf
19	12	AwardsWon
10	8	Titles
6	3	TopMembers
4	2	FoundedBy
2	2	DateOfDeath
2	1	EmployeeOf
1	1	SignificantOther
1	1	CauseOfDeath

Table 2

TAC KBP Interface

Text Analytics Conference's Knowledge Base Population (TAC KBP) Cold Start evaluation requires systems to auto-populate an entire knowledge base for every entity mentioned in a corpus of up to 100,000 documents. We are investigating how to interconnect KBA and KBP systems. As an initial small step in this direction, we helped the KBP organizers select a couple of the small town "groups" of KBA entities as the basis for the Cold Start input corpus. We identified Web site domains that frequently mentioned the KBA entities from these towns, and then generated an excerpt from the TREC StreamCorpus containing all documents from those domains. Along with these documents, we provided ratings from KBA systems as metadata input to the KBP Cold Start systems, so KBP participants can investigate whether this provides useful signal.

Conclusions and Future Directions

While KBA CCR remains a very challenging task, refining the definition of "vital" improved assessor agreement (20% reduction in disagreement), and some teams found methods that improve on simple baselines. SSF provided an initial step toward decomposing the filtering task into lower-level elements that model an human assessor's intuitive notion of "vital." Going forward, we plan to more tightly couple revised versions of CCR and SSF with the goal of getting closer to users' natural goals for accelerating the creation of knowledge bases.

Acknowledgements: Special thanks to Boyan Onyshkevych and Alan Goldschen for ideas and helpful discussions. JRF, MKW, DAR thank the Fannie and John Hertz Foundation for support. Ian Soboroff and Ellen Voorhees at TREC have been instrumental in designing and organizing KBA. We also wish to thank MIT, University of Wisconsin, the Open Science Grid, Amazon, Spinn3r, and the arXiv for their generous support of TREC KBA.

1: http://s3.amazonaws.com/aws-publicdatasets/trec/kba/index.html
http://s3.amazonaws.com/aws-publicdatasets/trec/kba/kba-streamcorpus-2013-v0_2_0-chunk-path-to-stream-id-time.txt.xz
2: https://github.com/trec-kba/kba-scorer/
3: http://www.nist.gov/tac/2012/KBP/task_guidelines/TAC_KBP_Slots_V2.4.pdf

4: http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines_v5.4.3.pdf

5: http://ciir.cs.umass.edu/~dietz/streameval/taia2013-cameraready.pdf
6: https://github.com/trec-kba/kba-tools/blob/f3a763daced7507025b437d690fa5b1e89fc1661/toy-system/toy_kba_algorithm.py#L140