Document Filtering for Long-tail Entities

Ridho Reinanda† r.reinanda@uva.nl
Edgar Meij‡ edgar.meij@acm.org
Maarten de Rijke† derijke@uva.nl

† University of Amsterdam, Amsterdam, The Netherlands
‡ Bloomberg L.P., London, United Kingdom

ABSTRACT

Filtering relevant documents with respect to entities is an essential task in the context of knowledge base construction and maintenance. It entails processing a time-ordered stream of documents that might be relevant to an entity in order to select only those that contain vital information. State-of-the-art approaches to document filtering for popular entities are entity-dependent: they rely on and are also trained on the specifics of differentiating features for each specific entity. Moreover, these approaches tend to use so-called *extrinsic* information such as Wikipedia page views and related entities which is typically only available only for popular head entities. Entity-dependent approaches based on such signals are therefore ill-suited as filtering methods for long-tail entities. In this paper we propose a document filtering method for long-tail entities that is *entity-independent* and thus also generalizes to unseen or rarely seen entities. It is based on intrinsic features, i.e., features that are derived from the documents in which the entities are mentioned. We propose a set of features that capture informativeness, entity-saliency, and timeliness. In particular, we introduce features based on entity aspect similarities, relation patterns, and temporal expressions and combine these with standard features for document filtering. Experiments following the TREC KBA 2014 setup on a publicly available dataset show that our model is able to improve the filtering performance for long-tail entities over several baselines. Results of applying the model to unseen entities are promising, indicating that the model is able to learn the general characteristics of a vital document. The overall performance across all entities—i.e., not just long-tail entities—improves upon the state-of-the-art without depending on any entity-specific training data.

Keywords

Document filtering; Long-tail entities; Semantic search

1. INTRODUCTION

A knowledge base contains information about entities, their attributes, and their relationships. Modern search engines rely on knowledge bases for query understanding, question answering, and document enrichment [3, 25, 30]. Knowledge-base construction, either based on web data or on a domain-specific collection of documents, is the cornerstone that supports a large number of downstream tasks. In this paper, we consider the task of entity-centric document filtering, which was first introduced at the TREC KBA evaluation campaign [18]. Given an entity, the task is to identify documents that are relevant and vital for enhancing a knowledge base entry of the entity given a stream of incoming documents.

To address this task, a series of *entity-dependent* and *entity-independent* approaches have been developed over the years. Entity-dependent approaches use features that rely on the specifics of the entity on which they are trained and thus do not generalize to unseen entities. Such methods include approaches that learn a set of keywords related to each entity and utilize these keywords for query expansion and document scoring [11, 24] as well as text-classification-based approaches that build a classifier with bag-of-word features for each entity [18]. Signals such as Wikipedia page views and query trends have been shown to be effective, since they usually hint at changes happening around an entity [5]; these signals are typically available for popular entities but when working with long-tail entities, challenges akin to the cold-start problem arise. In other words, features extracted from and working for popular entities may simply not be available for long-tail entities.

In this paper, we are particularly interested in filtering documents for long-tail entities. Such entities have limited or even no external knowledge base profile to begin with. Other extrinsic resources may be sparse or absent too. This makes an entity-dependent document filtering approach a poor fit for long-tail entities. Rather than learning the specifics of each entity, *entity-independent* approaches to document filtering aim to learn the characteristics of documents suitable for updating a knowledge base profile by utilizing signals from the documents, the initial profile of the entity (if present), and relationships between entities and documents [5, 31, 32]. While entity-dependent approaches might be able to capture the distributions of features for each entity better, entity-independent approaches have the distinct advantage of being applicable to unseen entities, i.e., entities not found in the training data. As an aside, entity-independent methods avoid the cost of building a model for each entity which is simply not practical for an actual production-scale knowledge base acceleration system.

Our main hypothesis is that a rich set of *intrinsic* features, based on aspects, relations, and the timeliness of the facts or events mentioned in the documents that are relevant for a given long-tail entity, is beneficial for document filtering for such entities. We consider a rich set of features based on the notion of *informativeness*, *entity-saliency*, and *timeliness*. The intuition is that a document (1) that contains a rich set of facts in a timely manner, and (2) in which the entity is prominent makes a good candidate for enriching a knowledge base profile. To capture informativeness, we rely on three

CIKM'16 , October 24 - 28, 2016, Indianapolis, IN, USA

ISBN 978-1-4503-4073-1/16/10. . . $15.00

DOI: http://dx.doi.org/10.1145/2983323.2983728

sources: generic Wikipedia section headings, open relations, and schematized relations in the document. To capture entity-saliency, we consider the prominence of an entity with respect to other entities mentioned in the document. To capture timeliness, we consider the time expressions mentioned in a document. We use these features with other basic features to train an entity-independent model for document filtering for long-tail entities.

Our main contributions can be summarized as follows: (1) We propose a competitive entity-independent model for document filtering for long-tail entities with rich feature sets designed to capture informativeness, entity-saliency, and timeliness. (2) We provide an in-depth analysis of document filtering for knowledge base acceleration for long-tail entities.

2. RELATED WORK

We review related work on document filtering in the TREC KBA track and other settings as well as related work on entity profiling.

2.1 Document filtering

The main approaches to KBA (and TREC KBA in particular) can be divided into entity-dependent and entity-independent approaches. When TREC KBA was first organized in 2012, many methods relied on *entity-dependent*, highly-supervised approaches utilizing related entities and bag of word features [18]. Here, the training data is typically used to identify keywords and/or related entities, in order to classify the documents in the test data. Later on, entity-independent models which rely less on the specifics of each entity emerge. Balog et al. [5] propose one such *entity-independent* approach. They study two multi-step classification methods for the stream filtering task. Their models start with an entity identification component based on alternate names from Wikipedia. They introduce a set of features that have commonly been used in subsequent TREC KBA campaigns. Balog and Ramampiaro [4] also compare classification and ranking approaches for this task; ranking outperforms classification on all evaluation settings and metrics on the TREC KBA 2012 dataset. Their analysis reveals that a ranking-based approach has more potential for future improvements.

Along this line of work, Bonnefoy et al. [7] introduce a weakly-supervised, entity-independent detection of the central documents in a stream. Zhou and Chang [34] study the problem of learning entity-centric document filtering based on a small number of training entities. They are particularly interested in the challenge of transferring keyword importance from training entities to entities in the test set. They propose novel meta-features to map keywords from different entities and contrast two different models: linear mapping and boost mapping.

Wang et al. [31] adopt the features introduced in [5] and introduce additional citation-based features, experimenting with different classification and ranking-based models. They achieve the best performance for vital documents filtering in KBA 2013 with a classification-based approach. Liu et al. [24] present a related entity-based approach. They pool related entities from the profile page of target entity and estimate the weight of each related entity with respect to the query entity. They then apply the weighted related entities to estimate confidence scores of streaming documents and explore various ways of weighting the related entities. Dietz and Dalton [11] also propose a query expansion-based approach on relevant entities from the KB. They do not address the novelty aspects of the task, however, and evaluate a memory-less method where predictions are not influenced by predictions on previous time intervals.

Wang et al. [32] propose a novel discriminative mixture model based by introducing a latent entity class layer to model the correlations between entities and latent entity classes. They achieve increased performance by inferring latent classes of entities and learning the appropriate feature weights for each latent class, as shown by experiments on the TREC KBA 2013 dataset. Later on, Wang et al. [33] introduce a latent document filtering model for cumulative citation recommendation in which they infer different latent classes and learn the appropriate feature weights for each latent class.

Gebremeskel and de Vries [21] perform an in-depth analysis of the main factors that affect the recall of document filtering on the TREC KBA 2013 corpus. They investigate the impact of choices for corpus cleansing, entity profile construction, entity type, document type, and relevance grading. They identify and characterize citation-worthy documents that do not pass the filtering stage by examining their contents and find that this can be caused by cleansing issues, incomplete name variants, or unclear assessments reasons.

In contrast with previous years, TREC KBA 2014 focused on long-tail entities and less than half of the entities in the test set that year have a Wikipedia profile [19]. Jiang and Lin [22] achieved the best performance using an entity-dependent approach which uses time range, temporal, profession, and action pattern features. Another notable approach within that year summarizes all information known about an entity so far in a low-dimensional embedding [9].

Next, we describe two tasks that are different but closely related to the document filtering setting of TREC KBA. Document filtering has been a traditional task in TREC, in the form of Topic Detection and Tracking (TDT) [1]. TDT constitutes a body of research and evaluation paradigm that address event-based organization of broadcast news. The goal of TDT is to break the text down into individual news stories, to monitor the stories for events that have not been seen before, and to gather stories into groups that each discuss a single news topic.

Dunietz and Gillick [13] introduce the entity salience task, that is given a document d, decide whether entity $e \in E_d$ is salient, i.e., a major talking point of the document. This task is similar to document filtering without the requirement of having the documents mentioning timely facts.

2.2 Entity profiling

Next, we turn to discovering entity-oriented pieces of information within a text. Fetahu et al. [16] propose a two-stage supervised approach for suggesting news articles to entity pages for a given state of Wikipedia. First, they suggest news articles to Wikipedia entities (article-entity placement), relying on a rich set of features which take into account the salience and relative authority of entities, and the novelty of news articles to entity pages. Next, they determine the exact section in the entity page for the input article (predicting the correct section for the article) guided by what they call class-based section templates. Banerjee and Mitra [6] explore the task of automatically expanding Wikipedia stubs. They introduce a model that assigns web content to a Wikipedia section and then perform abstractive summarization to generate section-specific summaries for the Wikipedia stubs.

Taneva and Weikum [29] propose an approach that automatically compiles salient information about entities in order to ease knowledge bases maintenance. They compile highly-informative, concise "gems" about entities, identifying salient pieces of text of variable granularity using a budget-constrained optimization problem, which decides which sub-pieces of an input text should be selected for the final result. Li et al. [23] propose a novel approach to automatically generate aspect-oriented summaries from multiple documents. They first introduce an event-aspect LDA model to cluster sentences into aspects and then use LexRank to rank

Table 1: Glossary of the main notation used in this paper.

Symbol	Gloss
S	Stream of documents
d	a document
e	an entity
p	a profile of an entity
a	an aspect of an entity

the sentences in each cluster, employing Integer Linear Programming for sentence selection. Song et al. [28] present a model to summarize a query's results using distinct aspects. For this they introduce the notion of "composite queries" that are used for providing additional information for a query and its aspects, comparatively mining the search results of different component queries. Balasubramanian and Cucerzan [2] propose a method to generate entity-specific topic pages as an alternative to regular search results. Cheng et al. [10] study the task of generating compact structured summaries. Reinanda et al. [27] mine entity aspects, common information needs around entities, from query logs, while Reinanda and de Rijke [26] focus on establishing temporal extents of entity relations, which can be useful for updating sections of entity profiles that are temporal in nature.

Our work is different in the following ways. First, we focus on the vital document filtering for long tail entities specifically. Next, we introduce a rich set features for identifying vital documents based on the notion *informativeness*, *entity-saliency*, and *timeliness* of the documents. Last, we apply these rich features to train an entity-independent model for vital document filtering.

3. PROBLEM DEFINITION

In this paper, we study the problem of identifying documents that contain vital information to add to a knowledge base. We formalize the task as follows. Given an entity e and a stream of documents S, we have to decide for each document $d_e \in S$ that mentions e whether it is vital for improving a knowledge base profile p_e of entity e. More formally, we have to estimate:

$$P(rel \mid d_e, e), \quad (1)$$

where rel is the relevance of document d_e with respect to entity e. A document is considered *vital* if it can enhance the current knowledge base profile of that entity, for instance by mentioning a fact about the entity within a short window of time of the actual emergence of the new fact. Note that a profile p_e is a textual description of an entity (i.e., not a structured object), such as a Wikipedia page or any other web page providing a description of the entity at a certain point in time.

4. METHOD

In this section, we describe our general approach to perform document filtering. We consider several intrinsic properties of a document that will help to detect vital documents. In particular, we consider the following dimensions:

- **Informativeness** – a document d that is rich in facts is likely to be vital.
- **Entity-saliency** – a document d in which an entity e is salient among the set of entities E occurring in d is likely to be vital.
- **Timeliness** – a document d that contains and discusses a timely event (with respect to document creation time or classification time) is likely to be vital.

We hypothesize that not all of these properties need to be satisfied for a document to be considered vital, i.e., some combination of features derived from these properties and other basic features for document filtering would apply in different cases.

Algorithm 1 Building a Wikipedia aspect model.

Input: Wikipedia entity category: c, Wikipedia articles: W
Output: Aspect model: A_c;
1: $C \leftarrow retrieveArticles(W, c)$
2: $H_C \leftarrow extractSectionHeadings(C)$
3: $aggregateSectionHeadings(H_C)$
4: **for** each $h \in H_C$ **do**
5: $\quad S_C \leftarrow retrieveSections(H_C, h)$
6: $\quad a_s \leftarrow combineSections(S_C)$
7: **end for**

4.1 Intrinsic features

Below, we describe the intrinsic features derived to capture the three dimensions described above and how these features are used to operationalize Eq. 1. The features are meant to be used in combination with others that are commonly used in document filtering and that will be described below. In the following paragraphs we describe these features; a high-level summary can be found in Table 2.

4.1.1 *Informativeness features*

Informativeness features aim to capture the richness of facts contained in a document. The intuition is that a document that contains a lot of facts, for instance in the form of relations, such as *work-for*, *spouse-of*, *born-in*, would be more likely to be vital. We operationalize informativeness in three ways, using entity page sections in a knowledge base (e.g., Wikipedia), open relations, and schematized relations as detailed below. We denote the informativeness features as F_I.

Wikipedia aspects. We define aspects as key pieces of information with respect to an entity. The central idea here is that a vital document contains similar language as some specific sections in Wikipedia pages; cf. [17]. We therefore aggregate text belonging to the same Wikipedia section heading from multiple Wikipedia pages in order to build a classifier. To be able to extract aspect features for a document, we first construct a bag-of-words model of aspects A_c of an entity type c from Wikipedia as detailed in Algorithm 1. Here we first retrieve Wikipedia articles of *all* entities belonging to the Wikipedia category c; our entities are filtered to be either in the *Person* or *Location* category. Next, we identify the section headings within the articles. We take the m most frequent section headings and, for each section heading, we remove stopwords and aggregate the contents belonging to the same heading by merging all terms that occur in the heading as an aggregated bag-of-words. We then represent each aggregated content section as a bag-of-words representation of aspect $a_k \in A$ and compute the cosine similarity between the candidate document d and aspect a_k to construct an aspect-based feature vector

$$A_k(d) = \cos(d, a_k), \quad (2)$$

We refer to the vector A_k as the $ASPECTSIM$ features in Table 2.

Open relation extraction. Here, we use the relation phrases available from an open information extraction system, i.e., Reverb [14]. As an open relation extraction system, Reverb does not extract a predefined set of entity relations from text, but detects any relation-like phrases. Given a text as input, it outputs unnormalized relational patterns in the form of triples of an entity, a verb/noun phrase, and another entity. As another feature, we utilize the relational

Algorithm 2 Selecting open relation phrase patterns.

Input: Open relation phrases: P, Corpus C
Output: Ranked open relations model: R;
$G \leftarrow groupPhrasesByLemma(P)$
for each $g \in G$ **do**
  for each $p \in g$ **do**
    $c_p \leftarrow getCount(C, p)$
  end for
  $c_g \leftarrow c_g + c_p$
end for
$R \leftarrow selectTopk(G, c)$

patterns generated by Reverb from the ClueWeb09 corpus [15]. Algorithm 2 details our procedure to generate a list of open relation phrases from this output. Due to the large number of patterns and limited amount of training data, it is not feasible to use all of these patterns as features. Therefore, we select popular phrases out of all available patterns. To this end, we first cluster the relation phrases based on their lemmatized form, obtaining grouped patterns G. Then, we estimate the importance of each pattern group $g \in G$ based on their aggregated count in the ClueWeb09 corpus. That is, we sum the occurrence c_p of each pattern p as the count of group g, obtaining c_g. Finally, we select the n most frequent relation phrases. We compute the feature vector by splitting a document into sentences and, for each relation phrase R compiled in the previous step, we generate a feature vector containing the counts:

$$R_k(d) = count(d, r_k), \tag{3}$$

where $count(d, r)$ returns the count of any instances of open relation pattern r in the document d. We refer to the vector R_k as the $RELOPEN$ features in Table 2.

Closed relation extraction. The last informativeness feature is based on the occurrence of a set of pre-defined relations within the text of the candidate document. We obtain all relation mentions detected in the text by a relation extraction system, the Serif tagger [8]. In our task, the corpus contains annotations of relation types based on the ACE relation schema [12]. We only consider relations involving entities that are a person, organization, or location which amounts to 15 ACE relation types. We construct a vector of the ACE relation types at the document level:

$$S_k(d) = count(d, s_k), \tag{4}$$

where $count(d, s)$ is the count of detected relations k in the document. We refer to S_k as the $RELSCHEMA$ features in Table 2.

4.1.2 Entity saliency features

The entity saliency features F_E aim to capture how prominently an entity features within a document. Although the basic features (defined in §4.2) might capture some notion of saliency, they are focused on the target entity only. We extend this by looking at mentions of other entities within the document. For example, if e is the only entity mentioned in the document then it is probably the main focus of the document.

We define a *full mention* as the complete name used to refer an entity in the document and a *partial mention* as the first or last name of the entity. We introduce the following novel features based on this notion of entity saliency. The first feature is simply the number of entities in the document:

$$DOCENTITIES(d) = |M|, \tag{5}$$

where M is the set of all entity mentions. The next feature is the number of entity mentions:

$$DOCMENTIONS(d) = \sum_{e'} n(d, m_{e'}), \tag{6}$$

that is, the total number of entity mentions as identified by the Serif tagger. The next feature is the number of sentences containing the target entity e:

$$NUMSENT(d, e) = |S_e|, \tag{7}$$

where S_e is the set of all sentences mentioning entity e.

We further define the fraction of full mentions of e with respect to all entity mentions in the document:

$$FULLFRAC(d, e) = \frac{n_{full}(d, m_e)}{\sum_{e'} n(d, m_{e'})}, \tag{8}$$

and also include the fraction of partial mentions m_e of e with respect to all entity mentions in the document:

$$MENTIONFRAC(d, e) = \frac{n_{partial}(d, m_e)}{\sum_{e'} n(d, m_{e'})}, \tag{9}$$

where $n(d, m)$ counts the number of mentions in document d again obtained by the named entity recognizer.

4.1.3 Timeliness features

Timeliness features F_T capture how timely a piece of information mentioned in the document is. We extract these features by comparing the document metadata containing the document creation time t with the time expressions mentioned in the documents:

$$TMATCH_Y(d) = count(year(t), d), \tag{10}$$

where $count(year(t), d)$ counts the occurrences of year expressions of t appear in the document.

$$TMATCH_{YM}(d) = count(yearmonth(t), d), \tag{11}$$

where $count(yearmonth(t), d)$ counts the number of times year and month expressions of t appear in the document. Finally,

$$TMATCH_{YMD}(d) = count(yearmonthday(t), d), \tag{12}$$

where $count(yearmonthday(t), d)$ counts the number of times the year, month, and date expressions of t occur in the document d.

4.2 Basic features

This section describes basic features F_B that are commonly implemented in an entity-oriented document filtering system [5, 31], as described in §2. We also propose some new basic features.

Document features. Features extracted from document d, capturing the characteristics of d independent of an entity. This includes the length, type, and language of d.

Entity features. Features based on knowledge about entity e including, for instance, the number of related entities in the entity's profile p_e. In addition, we incorporate the length of profile p_e and the type of entity profile available: *Wiki*, *Web*, or *Null*.

Document-entity features. Features extracted from an entity and document pair. This includes the occurrences of full and partial mentions of e in the document as well as the first and last position of occurring. They also include similarity between d and p_e and the number of related entities of e mentioned in the document.

Temporal features. Temporal features extracted from the occurrences of e within the stream corpus S. After aggregating entity mentions in hourly bins, we obtain the counts in the previous k hours before the creation of document d, where $k \in \{1, \ldots, 10\}$.

Table 2: Features for document filtering, for an entity e and/or document d. The last column indicates the value types of the features: N for numerical features and C for categorical features.

Feature	Description	Source	Type	Value
$SRC(d)$	Document source/type	[5]	basic	N
$LANG(d)$	Document language	[5]	basic	N
$REL(e)$	Number of of related entities of e	[5]	basic	N
$DOCREL(e)$	Number of of related entities of e in d	[5]	basic	N
$NUMFULL(d, e)$	Number of mentions of e in d	[5]	basic	N
$DOCREL(d, e)$	Number of of related entities of e in d	[5]	basic	N
$NUMPARTIAL(d, e)$	Number of partial mentions of e in d	[5]	basic	N
$FPOSFULL(d, e)$	First position of full mention of e in d	[5]	basic	N
$LPOSPART(d, e)$	Last position of partial mention of e in d	[5]	basic	N
$SPRPOS(d, e)$	Spread (first position $-$ last position) of mentions of e in d	[5]	basic	N
$SIM_{cos}(d, p_e)$	Text cosine similarity between d and p_e	[5]	basic	N
$SIM_{jac}(d, p_e)$	Text jaccard similarity between d and p_e	[5]	basic	N
$PREMENTION_h(d, e)$	Mention count of entity in the previous h hour before document creation time of d	[31]	basic	N
$DOCLEN_{chunk}(d)$	Length of document in number of chunks	this paper	basic	N
$DOCLEN_{sent}(d)$	Length of document in number of sentences	this paper	basic	N
$ENTITYTYPE(e)$	Type of e (PER, ORG, or FAC)	this paper	basic	C
$PROFILETYPE(e)$	Profile type: *wiki*,*web*, or *null*	this paper	basic	C
$PROFILELEN(e)$	Length of entity profile e	this paper	basic	N
$ASPECTSIM_k(d)$	Cosine similarity between d and $aspect_k$ estimated from Wikipedia	this paper	informativeness	N
$RELOPEN_k(d)$	Number of normalized open relation phrases k in d	this paper	informativeness	N
$RELSCHEMA_k(d)$	Number of relation type k in document d	this paper	informativeness	N
$NUMENTITIES(d)$	Number of unique entity mentions in the documents	this paper	entity saliency	N
$NUMMENTIONS(d)$	Number of entity mentions in the documents	this paper	entity saliency	N
$NUMSENT(d, e)$	Number of sentences in d containing entity e	this paper	entity saliency	N
$FULLFRAC(d, e)$	Number of full mentions of e in the document, normalized by number of entity mentions	this paper	entity saliency	N
$MENTIONFRAC(d, e)$	Number of full or partial mentions of e in the document, normalized by number of entity mentions	this paper	entity saliency	N
$TMATCH_Y(d)$	Number of year expressions of timestamp t in d	this paper	timeliness	N
$TMATCH_{YM}(d)$	Number of year, month expressions of timestamp t in d	this paper	timeliness	N
$TMATCH_{YMD}(d)$	Number of year, month, date expressions of timestamp t in d	this paper	timeliness	N

4.3 Machine learning model

Next, we detail our classification-based machine learning model. We formulate the task as binary classification and train a classifier to distinguish vital and non-vital documents using the concatenated vector of all features described previously: $F = F_B \cup F_I \cup F_E \cup F_T$. We train a global model M in an *entity-independent* way, utilizing all training data available for the model. Creating such a general model has the benefit that it can be readily applied to entities that do not exist in the training data.

We use gradient boosted decision trees (GBDT) [20] as our machine learning algorithm. GBDT learns an ensemble of trees with limited complexity in an additive fashion by iteratively learning models that aim to correct the residual error of previous iterations. To obtain the probabilistic output as required by Eq. 1, the gradient boosting classifier is trained as a series of weak learners in the form of regression trees. Each regression tree $t \in M$ is trained to minimize mean squared error on the logistic loss:

$$MSE = \frac{1}{n} \sum_{i}^{n} \left(y_i^2 - \left(\frac{1}{1 + e^{pred_i}} \right)^2 \right), \tag{13}$$

where y is the training label converted to either 0 or 1 for the negative and positive class, respectively, and $pred$ is the prediction score of the regression tree at data point i. The trees are trained in a residual fashion until convergence. At prediction time, each tree produces a score s_t; these are combined into a final score s, which is then converted into a probability using the logistic function:

$$P = \frac{1}{1 + e^{-s}}. \tag{14}$$

We take this output as our estimate of Eq. 1. We refer to our proposed entity-independent document filtering method as EIDF.

5. EXPERIMENTAL SETUP

In this section we detail our experimental setup including the data that we use, the relevance assessments, and the evaluation metrics. Our experiments address the following research questions: (RQ1) How does our approach, EIDF, perform for vital document filtering of long-tail entities? (RQ2) How does EIDF perform when filtering documents for entities not seen in the training data? (RQ3) How does EIDF compare to the state-of-the-art for vital document filtering in terms of overall results?

5.1 Data and annotations

The TREC KBA StreamCorpus contains 1.2B documents. Roughly half of these (579M) have been annotated with rich NLP annotations using the Serif tagger [19]. This annotated set is the official document set for TREC KBA 2014. Out of these annotated

Table 3: Distribution of entity profile types and examples.

Entity profile	Count	Examples
Wiki	14	*Jeff Mangum, Paul Brandt*
Web	19	*Anne Blair, Bill Templeton*
Null	41	*Ted Sturdevant, Mark Lindquist*

documents, a further selection is made for the Cumulative Citation Recommendation (CCR) task of KBA 2014. This results in the final *kba-2014-en-filtered* subset of 20,494,260 documents, which was filtered using surface form names and slot filling strings for the official query entities for KBA 2014. These documents are heterogeneous and originate from several Web sources: arxiv, classifieds, forums, mainstream news, memetracker, news, reviews, social, and blogs. We perform our experiments on this filtered subset.

The entities used as test topics are selected from a set of people, organizations, and facilities in specific geographical regions (Seattle, Washington, and Vancouver). The test entities consist of 86 people, 16 organizations, and 7 facilities, 74 of which are used for the vital document filtering task. Assessors judged ∼30K documents, which included most documents that mention a name from the handcrafted list of surface names of the 109 topic entities. Entities can have an initial profile in the form of *wikipedia*, *web*, or *null*, indicating that no entity profile is given as a description of the entity. In order to have enough training data for each entity, the collection was split based on per-entity cut-off points in time. Some of the provided profile pages are dated after the training time cutoff of an entity. To avoid having access to future information, we filter out entity profiles belonging to those cases. Table 3 provides a breakdown of profile types of the test entities.

Annotators assessed entity-document pairs using four class labels: *vital*, *useful*, *neutral*, and *garbage*. For a document to be annotated as *vital* means that the document contains (1) information that at the time it entered the stream would motivate an update to the entity's collection of key documents with a new slot value, or (2) timely, new information about the entity's current state, actions, or situation. Documents annotated as *useful* are possibly citable but do not contain timely information about the entity. *Neutral* documents are documents that are informative, but not citable, e.g., tertiary sources of information like Wikipedia pages. *Garbage* documents are documents that are either spam or contain no mention of the entity. The distribution of the labels is detailed in Table 4. As our model performs binary classification, we collapse the non-vital labels into one class during training.

One of our proposed features is based on generic Wikipedia sections of *Person* and *Location* entities. For this purpose, we use a Wikipedia dump from January 2012.

5.2 Experiments

We run three experiments: two main experiments aimed at assessing the performance of EIDF on long-tail entities and on unseen entities, and a side experiment in which we determine the performance on all entities.

Main experiment: Long-tail entities. Our main experiment aims to answer RQ1 and adapts the standard TREC KBA setting with one difference: we aggregate the results for different entity popularity segments. We define *long-tail entities* to be entities without a Wikipedia or Web profile in the TREC KBA ground truth data. All training entities are used to train the model and, during evaluation, a confidence score is assigned to every candidate document. All experiments are performed on the already pre-filtered documents using the canonical name of the entities as detailed above. Only

Table 4: Label distribution in the ground truth.

Label	Training	Test
Vital	1,360	4,665
Useful	5,482	20,370
Neutral	522	2,044
Garbage	3,302	1,961

documents containing at least a full match of the entity name are therefore considered as input. We focus on distinguishing vital and good documents, and use only documents belonging to these labels as our training data.

Main experiment: Unseen entities. Our second main experiment aims to assess the performance of EIDF on *unseen entities*, i.e., entities not found in the training data (RQ2). We design this experiment as follows. We randomly split the query entities into five parts and divide the training data accordingly. For every iteration we train on the training data consisting only of document-entity pairs of the corresponding entity split and test on the remaining split. We perform this procedure five times, resulting in a 5-fold cross-validation.

Side experiment: All entities. Our side experiment aims to answer RQ3 and follows the standard TREC KBA setting. All entities within the test set are considered in the evaluation (i.e., the results are not segmented) to asses the overall performance of EIDF.

5.3 Evaluation

In our experiments, we use the evaluation metrics introduced in the TREC KBA track for the vital filtering task: F_{macro}, and maximum scaled utility (SU). We also compute precision (P), recall (R), and F measure: the average of the harmonic mean of precision and recall over topics. For significance testing of the results, we use the paired t-test.

The main evaluation metric, F_{macro}, is defined as the *maximum* of the harmonic mean of averaged precision and recall computed at every possible threshold θ which separates vital and non-vital documents: $\max(\text{avg}(P), \text{avg}(R))$. The motivation behind this is evaluation setup is as follows. A filtering system will have a single confidence threshold θ for which the classification performance is maximized. Different systems might have different optimal confidence score calibrations, hence choosing the maximum scores with respect to each system's best threshold would ensure the fairest comparison. Below we explicitly distinguish between F_{macro} and F when reporting our experimental results.

SU is a linear utility measure that assigns credit to the retrieved relevant and non-relevant documents and is computed as follows:

$$SU = \frac{\max(NormU, MinU) - MinU}{1 - MinU},$$

where $MinU$ is a tunable minimum utility (set to -0.5 by default), and $NormU$ is the normalized version of utility function U which assigns two points for every relevant document retrieved and minus one point for every non-relevant document. The normalization is performed by dividing $NormU$ with the maximum utility score (i.e., 2 times the number of relevant documents). The official TREC KBA scorer sweeps over all the possible cutoff points and the reports the maximum SU. To gain additional insight, we also computed SU at the cutoff θ with the best F_{macro}: SU_{θ}.

5.4 Baselines

In our main experiments, we consider the following baseline approaches to compare the effectiveness of our approach.

Official Baseline [19]. The official baseline in TREC KBA considers matched name fractions as the confidence score.

BIT-MSRA [31]. A random forest, *entity-independent* classification approach utilizing document, entity, document-entity, and temporal features. This approach achieved the best official performance at the TREC KBA 2013 track.

In our side experiment aimed at assessing the performance of EIDF on all entities we also consider a state-of-the-art entity-dependent approach.

MSR-KMG [22]. A random forest, *entity-dependent* classification approach based on document cluster, temporal, entity title and profession features, with globally aligned confidence score. This approach achieved the best official performance in TREC KBA 2014. We take the team's best automatic run for comparison.

5.5 Parameters and settings

Recall that a document filtering system should output an estimate of $P(rel \mid d_e, e)$ (Eq. 1). The official KBA setup expects a confidence score in the $[0, 1000]$ range for each decision made regarding a document. To make the initial output of our model compatible with this setup, the probabilities are mapped to a confidence score that falls in this interval by adopting the mapping procedure introduced in [5]—we multiply the probability by 1000 and take the integer value.

Our approach involves two sets of hyperparameters. The first set deals with the machine learning algorithm of our choice. GBDT depends on two key parameters: the number of trees, k, and the maximum depth of each tree, d. The other set of parameters concerns the informativeness features. That is, the number of aspects that we used for the aspects-features, m, and the number of open relation patterns to consider, n.

We perform cross-validation on the training data to select the values of these parameters. For the GDBT parameter we consider $k = [100, 250, 500]$ and tree depth $d = [6, 7]$. For the informativeness parameters, we consider $m = [30, 40, 50]$ for the number of aspects and $n = [150, 200, 250]$ for number of the open relation patterns. We select the combination of parameters which maximize the mean F score across the validation folds, and finally set $k = 100$, $d = 6$, $m = 50$, and $n = 200$.

6. RESULTS AND DISCUSSION

In this section, we present and analyze our experimental results.

6.1 Main experiment: Long-tail entities

One of our goals in this work is to develop methods that are specifically geared towards filtering documents for long-tail entities. Therefore, we are particularly interested in comparing the performance of the methods on entities with different levels of popularity. To gain insight into our results along this dimension we segment the results by entity popularity using the type of entity profile as a proxy for popularity as defined in §5.2. We compute the best threshold for each approach, determine its per-entity performance using this cutoff, and then aggregate the performance by averaging the per-entity scores. We present these results in Table 5. Here, we answer RQ1 and compare our approach with other *entity-independent* approaches.

First, we look at the average scores in each popularity group, starting with the *Null* segment, which represents the long-tail entities in our setting. In the *Null* segment, the recall performance of different methods is considerably lower than on the other two segments, but this is complemented by the fact that precision is higher than for the *Wiki* segment. One important factor in this analysis

Table 5: Results segmented by entity popularity. Significance of EIDF result is tested against the strong baseline (BIT-MSRA). Significant improvement is denoted with ▲ ($p < 0.05$). Here the *null profiles* segment represents the long-tail entities.

Segment	P	R	F	SU_θ
Null profiles				
Official baseline	0.279	**0.973**	0.388	0.268
BIT-MSRA	0.362	0.630	0.404	0.313
EIDF	**0.398**▲	0.645	**0.433**▲	**0.350**▲
Web profiles				
Official baseline	0.391	**1.000**	0.513	0.381
BIT-MSRA	**0.430**	0.867	**0.536**	**0.429**
EIDF	0.424	0.827	0.517	0.410
Wiki profiles				
Official baseline	0.169	**0.975**	0.275	0.044
BIT-MSRA	0.204	0.737	0.296	0.121
EIDF	**0.227**▲	0.704	**0.317**	**0.130**

is that these are most likely tail entities with very few candidate documents to consider. More importantly, our approach achieves a significant improvement in the *Null* segment, while keeping a comparable or better performance as compared to BIT-MSRA on the *Wiki*, and *Web* segments. In particular, the improvements in precision, F, and SU_θ in this segment are statistically significant.

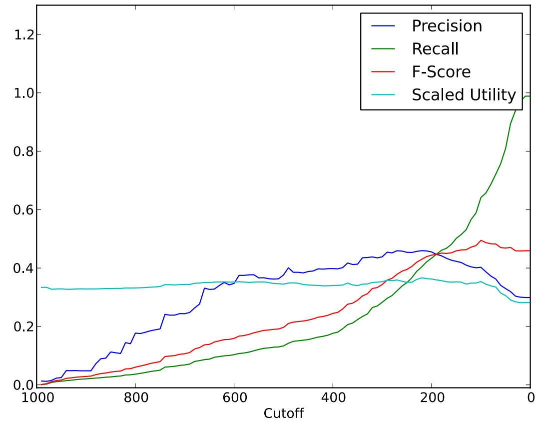
This finding is important because it confirms the effectiveness of our approach in the setting of long-tail entities. Faced with a considerably smaller pool of candidate documents in this segment, EIDF manages to detect more vital documents while simultaneously improving precision. Note that in the TREC KBA 2014 track, long-tail entities constitute a large fraction of the query entities (41 entities, i.e., 56%). The performance of EIDF and BIT-MSRA for long-tail entities across different cutoff points is shown in Fig. 1.

Filtering documents for the *Web* profile segment seems to be the easiest relative to the other segments. Recall and precision are highest compared to the other groups, which explains the higher F score. Our approach, EIDF, achieves a P score of 0.424, an F score of 0.517 and SU_θ of 0.410 in this segment. This happens to be lower than the strong baseline (BIT-MSRA), but the differences in performance in this segment are not statistically significant.

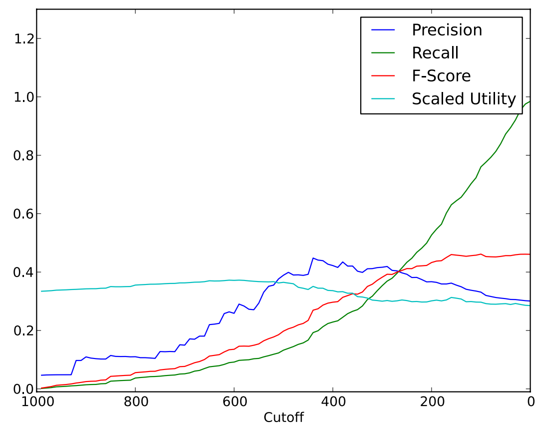
Interestingly, the performance of all methods when filtering documents of entities belonging to the *Wiki* group is the lowest. The recall is relatively high, but the F score is brought down by the lower precision. This may be due to the fact that these popular entities have a much larger pool of candidate documents, making the filtering task difficult because a system has to recover only a selective fraction of the documents. Thus, faced with a large set of candidate documents, methods tend to work towards obtaining high recall. Despite this, EIDF manages to get the best precision, obtaining a significant improvement over the strong baseline. The low SU_θ scores indicate that it is difficult to beat a system that returns no relevant documents for this segment group.

After looking at the general performance across the different segments, we compare the performance of our approach against the official TREC KBA baseline. Considerable gains are obtained in all three segments in terms of precision, F and SU_θ.

Informed by the previous insights, we also perform a follow-up experiment on training segment-conditioned models. Since feature value distributions might be different due to the popularity of an entity, we need to distinguish long-tail entities from more popular ones. One natural way of doing is to consider the existence of a



(a) EIDF



(b) BIT-MSRA

Figure 1: Performance of EIDF and BIT-MSRA for long-tail entities across different cutoff points.

knowledge base profile from Wikipedia—some entities may have a Wikipedia profile, some only an initial profile on a webpage, and some entities have no profile at all. To capture this difference in characteristics, we train three separate machine learning models: M_{wiki} for entities with a Wikipedia page, M_{web} for entities with a lesser profile in the form of a Web page, and M_{null} for entities with no profiles at all. During prediction, the appropriate model is automatically selected and applied to perform the predictions. We failed to obtain any improvements with these segment-conditioned models. This may be due to the fact that by segmenting the data, we lose important information required to train our model with rich feature sets. To fully utilize the data while recognizing the different characteristics of each segment, a learning algorithm that can handle feature interaction, as we employ with tree-based ensembles, seems like a good solution. Having one global model that can handle feature interaction seems to be a better way to handle this problem, without resorting to individual models.

In sum, our approaches achieve the best performance overall across different segments, with the biggest performance gain realized for the long-tail entities segment. Importantly, the features designed for improvement in the long-tail entities segment do not have a significant detrimental effect on the results of other segments. In addition, learning a separate model for each segment does not yield additional benefits.

Table 6: Results of cross-validation experiments with unseen entities, in terms of F_{macro} (top), P (middle), and R (bottom).

	Fold1	Fold2	Fold3	Fold4	Fold5	Overall
Official baseline	0.410	0.482	0.401	0.532	0.400	0.445
BIT-MSRA	0.405	**0.489**	0.413	0.537	0.407	0.450
EIDF	**0.458**	0.485	**0.438**	**0.539**	**0.408**	**0.465**
Official baseline	0.256	0.318	0.252	0.363	0.250	0.288
BIT-MSRA	0.258	**0.324**	0.266	0.371	0.257	0.295
EIDF	**0.328**	0.320	**0.329**	**0.373**	0.257	**0.321**
Official baseline	**1.000**	**1.000**	**0.975**	**0.993**	**1.000**	**0.994**
BIT-MSRA	0.956	0.992	0.923	0.972	0.976	0.964
EIDF	0.762	0.996	0.654	0.973	0.987	0.874

6.2 Main experiment: Unseen entities

In this section, we describe the results of our experiments on answering (RQ2). The results of our experiments with unseen entities are detailed in Table 6. Our approach performs best on almost all folds in terms of F_{macro}, gaining significant improvements compared to other approaches on Fold1 and Fold3.

Averaged over all folds, our approach also achieves the best performance. The differences between the performance of different methods in the unseen entities setting is very small in terms F_{macro}. Overall, the learned model tends to be precision-oriented with some loss in recall. Compared to the results of the main experiments (Table 5), the result is lower in terms of absolute score. This might be explained as follows. First, The model is now learning on less data—roughly 80% of the full data, depending on the number of data points that contribute to the folds. Secondly, the model is now performing predictions on entities that may have very different characteristics than the ones found in the training data. The average scores in each fold also vary considerably. This can be explained by the fact that by splitting the data in terms of entities, we might end up with different numbers of training and testing data in each split. Additionally, the inherent difficulty of filtering documents within each fold will also vary based on the popularity and the size of the candidate document pools. The magnitude of the improvements obtained in each fold also tends to be smaller, because, with 80% of the data, there are fewer positive examples available to learn a rich set of features (due to the imbalance of *vital* and *non-vital* document labels).

The results of filtering documents for unseen entities are quite promising, and the fact that the learning algorithm is able to achieve a better score than a name fraction baseline indicates that it is successful in learning the characteristics of vital documents and applying it to new, unseen entities.

6.3 Side experiment: All entities

To answer (RQ3), we compare our method, EIDF, with *entity-independent* and *entity-dependent* baselines in terms of overall, non-segmented results. Table 7 shows the results for this experiment. First, looking at the absolute scores, all methods improve over the official baseline in terms of F_{macro}, SU, and P. The official baseline unsurprisingly achieves the highest recall as it simply considers all document containing exact mentions of the target entity as vital.

Our approach also outperforms the two entity-independent baselines in terms of F_{macro}; we achieve significant improvements over BIT-MSRA in terms of precision, while maintaining the same level of recall. BIT-MSRA achieves a slightly better performance than EIDF in terms of SU. However, the difference is very small and not significant.

Table 7: Overall results with official and additional metrics. Significance of EIDF result is tested against the strong baseline (BIT-MSRA). Significant improvements are denoted with ▲ ($p < 0.05$). The official TREC KBA scorer returns F_{macro}, SU, P, and R. We also compute additional metrics, F and SU_θ to gain more insight about the results. We can not compute the significance test against MSR-KMG because the run is not available. Due to the way F_{macro} is computed in TREC KBA, as a harmonic mean over recall and precision macro statistics, significance testing cannot be applied to F_{macro}.

Method	P	R	F	SU_θ	F_{macro}	SU
Entity-independent						
Official baseline	0.286	**0.980**	0.397	0.253	0.442	0.333
BIT-MSRA	0.348	0.709	0.415	0.305	0.467	0.370
EIDF	0.371▲	0.701	0.432▲	0.323▲	0.486	0.367
Entity-dependent						
MSR-KMG (automatic) [22]	**0.378**	0.744	–	–	**0.501**	**0.377**

Table 8: Feature importance analysis for the model learned in the main and side experiments on long-tail entities.

Feature	Importance
$FPOSFULL(d, e)$	0.030
$PROFILELEN(e)$	0.025
$FPOSFULL_N(d, e)$	0.022
$REL(e)$	0.021
$ASPECTSIM_{filmography}(d)$	0.019
$DOCLEN_{SENT}(d)$	0.018
$MENTIONFRAC(d, e)$	0.016
$PREMENTION_{h2}(d, e)$	0.016
$SIM_{cos}(d, p_e)$	0.015
$ASPECTSIM_{coachingcareer}(d)$	0.015
$LPOSFULL(d, e)$	0.014
$ASPECTSIM_{politicalcareer}(d)$	0.013
$LSPRFULL_N(d, e)$	0.013
$TMATCH_Y(d)$	0.012
$LPOSFULL_N(d, e)$	0.012
$SIM_{jaccard}(d, p_e)$	0.012

Compared to the best entity-dependent approach, EIDF obtains a comparable level of precision and F_{macro}. In summary, EIDF achieves the best entity-independent performance and competitive performance to the state of the art entity-dependent approach.

6.4 Feature analysis

Recall that we learn a single, entity-independent model across all entities. We zoom in on the effectiveness of each feature within this global, entity-independent model. The importance of each feature is determined by averaging its importance across the trees that comprise the ensemble model. We observe several things. First, the most important features are a combination of common features in document filtering, e.g., the first position of the entity, the spread of entity mentions, and our proposed features. One of our proposed features (profile length) is the most discriminative feature and another of our proposed saliency features, the fraction of entity mentions, is also shown to be quite important. As for the rest, the aspect-based features seem to be the most important informativeness features, with as many as three features belonging to the aspect-based group in the top most important features.

Table 9: Top Wikipedia aspect importance.

Feature	Importance
filmography	0.019
coaching-career	0.015
political-career	0.013
wrestling	0.011
references	0.011
championships-accomplishments	0.011
footnotes	0.011
achievements	0.011
selected-publications	0.010
links	0.010

Table 10: Feature types within the top-30.

Feature type	Number of features
basic	14
informativeness	13
entity saliency	2
timeliness	1

The aspect-based features might be complementary to the more common cosine and jaccard profile similarity features. In combination with the profile length feature the aspect-based features seem to be triggered when the profile similarity scores are zero, which will happen in the case of entities without a profile. Having established this, we zoom in on the most important aspect-based features as detailed in Table 9. Recall that in our experiments, we use the top-50 aspects constructed from Wikipedia. Often, including aspects-based features seems intuitive, as is the case for, e.g., *achievements*, *accomplishment*, *coaching-career*, and *political-career*, since they are things that are typically included in vital documents.

All in all, we extracted 358 features. A breakdown of feature types in the top-30 features is shown in Table 10. The informativeness features not ranked among the top in the table are not as discriminative as the Wikipedia aspects. In the case of open relation patterns, some receive a zero relative importance score. One possible explanation is that these patterns are very common and may occur in many documents, thus having very little discriminative power. In other cases, the patterns are quite rare, and they might thus only occur in a few documents.

7. CONCLUSION AND FUTURE WORK

In this paper we have addressed an information filtering task for long-tail entities on a stream of documents. In particular, we have developed and evaluated a method called EIDF for classifying vital and non-vital documents with respect to a given entity. We have done so by designing intrinsic features that capture the notions of *informativeness*, *entity saliency*, and *timeliness* of documents. We have also considered the challenges related to filtering long-tail entities and have adjusted our features accordingly. We have applied these features in combination with a set of basic document filtering features from the literature to train an *entity-independent* model that is also able to perform filtering for entities not found in the training data. Upon segmenting our results by entity popularity, as approximated by its profile type, we have found that our approach is particularly good at improving document filtering performance for long-tail entities. When looking at the overall results of experiments conducted on the TREC KBA 2014 test collection we have found that our approach is able to achieve competitive performance compared to state-of-the-art automatic *entity-dependent* approaches. On filtering documents for unseen entities, we have found that our approach achieves a lower absolute performance overall than on seen entities, as is to be expected, but still improves over a strong name matching and classification baseline. A feature analysis revealed two things. First, entity popularity, proxied using the profile length feature is important. Second, informativeness features, and in particular aspect-based features derived from Wikipedia, are important for this task.

In summary, our results confirm the effectiveness of our entity-independent document filtering approach for knowledge base acceleration for long-tail entities, with (1) its ability to improve filtering performance specifically on the segment of tail entities, and (2) its relatively good performance on classifying documents for unseen entities, i.e., those not found in the training data.

As to future work, we are interested in exploring several directions. First, it would be interesting to explore the effect of combining the proposed features with other machine learning algorithms. Our preliminary experiment in this direction with applying logistic regression as the underlying learning algorithm indicates that we can obtain similar improvements. Next, we aim to apply more semantic approaches such as entity linking to detect entities and concepts mentioned in the context of a target entity. Last, we want to apply incremental learning so as to obtain a document filtering model that is able to learn from its previous decisions.

Acknowledgments. This research was supported by Ahold, Amsterdam Data Science, Blendle, the Bloomberg Research Grant program, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the ESF Research Network Program ELIAS, the Royal Dutch Academy of Sciences (KNAW) under the Elite Network Shifts project, the Microsoft Research Ph.D. program, the Netherlands eScience Center under project number 027.012.105, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, 612.066.-930, CI-14-25, SH-322-15, 652.002.001, 612.001.551, 652.001.003, the Yahoo Faculty Research and Engagement Program, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] J. Allan. Introduction to topic detection and tracking. In *Topic Detection and Tracking*, pages 1–16. Kluwer Academic Publishers, 2002.

[2] N. Balasubramanian and S. Cucerzan. Automatic generation of topic pages using query-based aspect models. In *CIKM '09'*, pages 2049–2052. ACM, 2009.

[3] N. Balasubramanian and S. Cucerzan. Topic pages: An alternative to the ten blue links. In *ICSC '10*. IEEE, 2010.

[4] K. Balog and H. Ramampiaro. Cumulative citation recommendation: Classification vs. ranking. In *SIGIR '13'*, pages 941–944. ACM, 2013.

[5] K. Balog, H. Ramampiaro, N. Takhirov, and K. Nørvåg. Multi-step classification approaches to cumulative citation recommendation. In *OAIR '13*, pages 121–128. Le Centre De Hautes Etudes Internationales D'Informatique Documentaire, 2013.

[6] S. Banerjee and P. Mitra. WikiKreator: Improving Wikipedia Stubs Automatically. In *ACL '15*, pages 867–877, 2015.

[7] L. Bonnefoy, V. Bouvier, and P. Bellot. A weakly-supervised detection of entity central documents in a stream. In *SIGIR '13*. ACM, 2013.

[8] E. Boschee, R. Weischedel, and A. Zamanian. Automatic information extraction. *ICIA '05*, pages 2–4, 2005.

[9] I. Cano, I. Cs, W. Edu, and G. Cs. Distributed non-parametric representations for vital filtering: UW at TREC KBA 2014. In *TREC 2014*. NIST, 2014.

[10] G. Cheng, D. Xu, and Y. Qu. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *WWW '15*, pages 184–194. International World Wide Web Conferences Steering Committee, 2015.

[11] L. Dietz and J. Dalton. UMass at TREC 2013 Knowledge Base Acceleration track. In *TREC 2013*. NIST, 2013.

[12] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel. The automatic content extraction (ACE) program-tasks, data, and evaluation. In *LREC*, 2004.

[13] J. Dunietz and D. Gillick. A new entity salience task with millions of training examples. In *EACL '14*. ACL, 2014.

[14] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction: The second generation. *IJCAI 2011*, pages 3–10, 2011.

[15] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *EMNLP '11*, 2011.

[16] B. Fetahu, K. Markert, and A. Anand. Automated news suggestions for populating wikipedia entity pages. In *CIKM '15*, pages 323–332. ACM, 2015.

[17] S. Fissaha Adafre and M. de Rijke. Discovering missing links in wikipedia. In *Proceedings of the Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-2005)*. ACM, August 2005.

[18] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, Z. Ce, R. Christopher, and I. Soboroff. Building an entity-centric stream filtering test collection for TREC 2012. In *TREC 2012*. NIST, 2012.

[19] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, E. Voorhees, and I. Soboroff. TREC KBA Overview. In *TREC 2014*. NIST, 2014.

[20] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[21] G. G. Gebremeskel and A. P. de Vries. Entity-centric stream filtering and ranking: Filtering and unfilterable documents. In *TREC 2014*, 2014.

[22] J. Jiang and C.-Y. Lin. MSR KMG at TREC 2014 KBA Track Vital Filtering Task. In *TREC*, 2014.

[23] P. Li, Y. Wang, W. Gao, and J. Jiang. Generating aspect-oriented multi-document summarization with event-aspect model. In *EMNLP '11*, pages 1137–1146. ACL, 2011.

[24] X. Liu, J. Darko, and H. Fang. A related entity based approach for knowledge base acceleration. In *TREC 2013*. NIST, 2013.

[25] P. Pantel and A. Fuxman. Jigs and lures: Associating web queries with structured entities. In *HTL '11*, 2011.

[26] R. Reinanda and M. de Rijke. Prior-informed distant supervision for temporal evidence classification. In *Coling '14*, pages 996–1006. ACL, August 2014.

[27] R. Reinanda, E. Meij, and M. de Rijke. Mining, ranking and recommending entity aspects. In *SIGIR '15*, pages 263–272. ACM, August 2015.

[28] W. Song, Q. Yu, Z. Xu, T. Liu, S. Li, and J.-R. Wen. Multi-aspect query summarization by composite query. In *SIGIR '12*, pages 325–334. ACM, 2012.

[29] B. Taneva and G. Weikum. Gem-based entity-knowledge maintenance. In *CIKM '13*, pages 149–158. ACM, 2013.

[30] N. Voskarides, E. Meij, M. Tsagkias, M. de Rijke, and W. Weerkamp. Learning to explain entity relationships in knowledge graphs. In *ACL-IJCNLP 2015*, 2015.

[31] J. Wang, D. Song, C. Lin, and L. Liao. BIT and MSRA at TREC KBA CCR Track 2013. In *TREC 2013*. NIST, 2013.

[32] J. Wang, D. Song, Q. Wang, Z. Zhang, L. Si, L. Liao, and C.-Y. Lin. An entity class-dependent discriminative mixture model for cumulative citation recommendation. In *SIGIR '15*, pages 635–644. ACM, 2015.

[33] J. Wang, D. Song, Z. Zhang, L. Liao, L. Si, and C.-y. Lin. LDTM: A latent document type model for cumulative citation recommendation. In *EMNLP '15*, pages 561–566. ACL, 2015.

[34] M. Zhou and K. C.-C. Chang. Entity-centric document filtering: Boosting feature mapping through meta-features. In *CIKM '13*, pages 119–128. ACM, 2013.