Distributed Non-Parametric Representations for Vital Filtering: UW at TREC KBA 2014

Ignacio Cano ICANO@CS.WASHINGTON.EDU
Sameer Singh SAMEER@CS.WASHINGTON.EDU
Carlos Guestrin GUESTRIN@CS.WASHINGTON.EDU
Computer Science and Engineering, University of Washington, Seattle, WA 98195 USA

Abstract

Identifying documents that contain timely and vital information for an entity of interest, a task known as *vital filtering*, has become increasingly important with the availability of large document collections. To efficiently filter such large text corpora in a streaming manner, we need to compactly represent previously observed entity contexts, and quickly estimate whether a new document contains novel information. Existing approaches to modeling contexts, such as bag of words, latent semantic indexing, and topic models, are limited in several respects: they are unable to handle streaming data, do not model the underlying topic of each document, suffer from lexical sparsity, and/or do not accurately estimate temporal vitalness. In this paper, we introduce a word embedding-based non-parametric representation of entities that addresses the above limitations. The word embeddings provide accurate and compact summaries of observed entity contexts, further described by topic clusters that are estimated in a non-parametric manner. Additionally, we associate a staleness measure with each entity and topic cluster, dynamically estimating their temporal relevance. This approach of using word embeddings, non-parametric clustering, and staleness provides an efficient yet appropriate representation of entity contexts for the streaming setting, enabling accurate vital filtering.

1. Introduction

To build up to date entity profiles from streaming text corpora, we need to find references to entities of interest, and study the information trends over time. Unfortunately, this is an incredibly difficult task to perform efficiently on streams, and thus, a large number of pertinent articles are seldom retrieved by automated approaches. For example, Frank et al. (2012) observe a considerable lag between the publication date of articles and the date of their citations in Wikipedia. The median time is over a year, and the distribution has a long and heavy tail. This gap can be drastically reduced if automated systems can accurately and efficiently suggest relevant documents for entities of interest to editors as soon as they are published.

The Knowledge Base Acceleration (KBA) track at TREC addresses this task. Recent submissions to KBA (Liu et al., 2013; Bouvier & Bellot, 2013; Efron et al., 2013; Zhang et al., 2013; Bellogín et al., 2013) focus on solving the aforementioned problems with supervised methods, using mainly document, document-entity, and temporal level features. They are, however, somewhat limited: they depend heavily on labeled data, do not handle lexical sparsity in contexts appropriately, and further, do not model the various semantic topics in the references.

In this submission, we introduce a semi-supervised approach suitable for streaming settings that uses word embedding clusters and temporal relevance to represent entity contexts. In particular, the word embeddings provide low-dimensional yet accurate summaries of previously observed entity contexts, and our algorithm updates the topic clusters, number of topics, and the entities and topics temporal relevance in an online fashion, observing only a single document at a time. We use this representation of unlabeled documents as features in a supervised classifier to utilize labeled data. This combination of word embeddings, non-parametric clustering, and temporal relevance (staleness) provides an efficient yet accurate representation of entity contexts that can be updated in a streaming manner, thus addressing the document filtering requirements on large streams of text. We present experimental results that demonstrate the benefits of our method and show our performance on the TREC KBA 2014 Vital Filtering task. As part of the Accelerate and Create

Proceedings of the Text Retrieval Conference (TREC): Knowledge-Base Acceleration (KBA), Gaithersburg, MD, USA, 2014. Copyright 2014 by the author(s).

task, we also describe an exploratory tool for efficient and intuitive visualization of large streams.

2. Vital Filtering Task

In this section, we formalize the problem setup and introduce our notation. We assume a set of m target entities $E = \{e_1, ..., e_m\}$. We further assume a set of n documents $D = \{d_1, ..., d_n\}$ that arrive in chronological order.

Each document is a sequence of sentences composed by collections of words, annotated with NLP tools. Further, we assume w.l.o.g. that every document in D refers to a single entity $e \in E$. Since our focus here is to distinguish vital and non-vital references, we use a naive classifier based resolution to identify the documents relevant (or referent) to each entity (details in Section 6.4), although in practice this is a challenging task and more sophisticated techniques are required (Rao et al., 2010; Singh et al., 2011).

A mention to e in a document $d_i \in D$ is identified by a string matching algorithm that searches for exact matches of canonical and surface form names of the entity e. We represent each d_i as a compound of a timestamp t_i and a bag of words $W_i = \{w_{i1}, ..., w_{ip}\}$ located in the context of (and including) mentions to the entity e. Finally, we assume an online setting, i.e. the algorithm should provide predictions for documents arriving at time t before seeing any documents arriving at time $t + 1$.

Given this setup, the vital filtering task requires classification of each document d relevant to an entity e as follows:

- *Vital* if the document contains information that, at the time it enters the stream, would cause an update to the entity e with timely, new information about the entity current state, actions or situation, e.g. "Barack Obama has been elected President".
- *Non-Vital* if the document is relevant, but contains information that is not timely, i.e. it may contain information relevant when building an initial profile of the entity e, but does not contain information that an accurate, updated profile would not have, e.g. "Barack Obama was born on August 4th, 1961".

3. Proposed Approach

Given a stream of documents D that refer to entity e, the task at hand is to predict whether each document is *vital* or *non-vital* to e. To detect whether a document contains novel information, one needs to provide an accurate and generalizable representation of historical contexts, while also capturing the temporal dynamics of the references.

To this end, we propose a three-pronged solution: (1) represent documents with low-dimensional embeddings that address sparsity and generalization (Section 3.1), (2) represent the entity context using non-parametric topic clustering (Section 3.2), and (3) estimate the novelty of the document information using a staleness measure (Section 3.3).

3.1. Document Embeddings

To identify whether an entity context in a document contains novel information, or even if it is relevant for the entity, we need a structured representation of the context. A common solution to this problem is to use vector space models, often the Bag of Words (BOW) models, where a document is represented as the bag of its words, disregarding grammar and even word order. Unfortunately, vector space models are often too sparse to represent fine-grained information in contexts, for example, straightforward BOW representations will have minimal overlap between "Barack was elected president today" and "Obama has won the election", treating the other as novel information even after having seen one of them. Further, the size of BOW representations grows over time when the vocabulary is not predefined beforehand, which is a problem in streaming settings.

In order to address these concerns, we propose to represent contexts of entities in documents using word embeddings. A word embedding is a dense, low-dimensional, and real-valued vector associated with every word in a vocabulary such that they capture useful syntactic and semantic properties of the contexts that the word appears in. The low-dimensionality of the embeddings as compared to vector space models (hundreds instead of millions) make them an elegant solution to address lexical sparsity in settings with very few labels (Turian et al., 2010), and further, they can be efficiently trained on massive corpora. Many of the syntactic patterns can be represented with simple algebraic operations. For example, the result of $v_{paris} - v_{france} + v_{germany}$ is closer to v_{berlin} than to any other word vector (Mikolov et al., 2013a;b).

We introduce a function $f : w \rightarrow v_w \in \mathbf{R^d}$ that defines the pre-computed word embedding representation of the word type w[1]. To define the embedding for a set of words W, we define $g : W \rightarrow v_W \in \mathbf{R^d}$ that computes embedding as:

$$g(W) = v_W = \frac{1}{|W|} \sum_{w \in W} f(w) \tag{1}$$

Given the document $d_i \in D$ that refers to entity e and contains the words $w_i \in W_i$, we compute its vector representation using function g as follows:

$$v_{d_i} = v_{W_i} = g(W_i) \tag{2}$$

With this, we intend to capture the context where the entity

[1]We use the 300-dimensional word embeddings trained on the Google News Corpus, available at `https://code.google.com/p/word2vec/`.

e is mentioned in a document, i.e. the topic, and represent it with a dense, low-dimensional vector.

Further, it may be useful to separately capture the context in terms of different parts of speech. Let W_{i_n} denote the set of all common nouns in W_i, W_{i_N} the set of the proper nouns, and W_{i_v} the set of all verbs in W_i, where $W_{i_n} \cup W_{i_N} \cup W_{i_v} = W_i$. We compute the embedding vectors of all the common nouns, proper nouns, and verbs that appear in the context of entity e using function g, as:

$$v_{d_{i_n}} = v_{W_{i_n}} = g(W_{i_n}) \tag{3}$$

$$v_{d_{i_N}} = v_{W_{i_N}} = g(W_{i_N}) \tag{4}$$

$$v_{d_{i_v}} = v_{W_{i_v}} = g(W_{i_v}) \tag{5}$$

Computing separate embeddings for different word types is a flexibility our method provides that may better encapsulate the underlying content of the document.

3.2. Non-parametric Clustering

Although word embeddings capture the context around a single topic quite accurately, they are unable to represent the variety of topics that an entity may be mentioned in. For example, the context around Obama during *elections* is quite different from the context for *presidential speech* or *international visit*. Using a single word embedding to represent multiple such topics may result in embeddings that conflate them, i.e. a single embedding is inaccurate for representing multiple topics.

One typical approach to tackle this problem is using topic models (Blei, 2012). Such models can be trained in an offline manner over a large corpus, followed by streaming inference for each document. However, the number of topics often needs to be decided apriori, which is quite difficult to specify for each entity of interest (non-parametric approaches to LDA are quite expensive). Further, drift over time can make the topic distributions obsolete. Finally, it is difficult to learn per entity topic distributions, especially if some of the entities have very few relevant documents.

Instead of representing the context using only a single embedding, we propose to use a number of embeddings that capture the different *topic clusters* of the entity, retaining the advantages of using embeddings while still having a precise context representation. We assume that the context in a single document belongs to a single topic, though we dynamically estimate the number of topic clusters in a non-parametric manner. As we are concerned with a streaming setting, topic clusters evolve over time, i.e. identities, members and number of clusters change over time.

We represent each topic cluster by the mean embedding vector of the documents assigned to that cluster at a certain timestamp. More precisely, the vector representation of the j-th topic cluster at timestamp t_i, c_i^j, can be computed using:

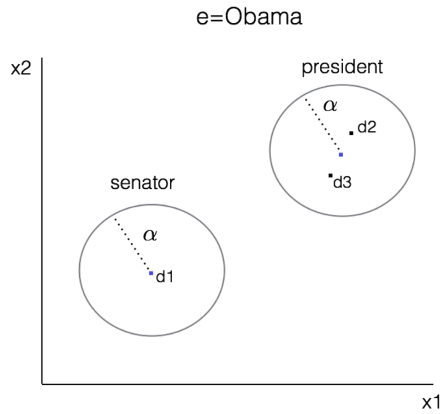


Figure 1: Example of Non-parametric Clustering

$$v_{c_i^j} = \frac{1}{|D_i^j|} \sum_{d \in D_i^j} v_d \tag{6}$$

where D_i^j is the subset of all the documents that belong to cluster j at timestamp t_i, and $\forall\ d_q \in D_i^j, t_q \leq t_i$.

The number of topic clusters for the context of entity e is unknown beforehand. Initially, we let the entity context to have zero topic clusters. We create the first topic cluster for the entity context when the first relevant document is observed. For any following relevant document d, the topic clusters are updated as follows. We first compute a distance of v_d with every existing topic cluster. If the minimum distance to any topic cluster is greater than or equal to α ($0 \leq \alpha \leq 1$), we create a new topic cluster just containing document d, otherwise we merge document d into the closest cluster to v_d, and update the cluster vector representation. Our approach is closely related to the online non-parametric clustering procedure described in Neelakantan et al. (2014).

More formally, $\forall\ c_{i-1}^j$, at time i, document d_i is added to the topic cluster that solves the following optimization problem:

$$\begin{aligned} &\arg\min_j \ \ dist(v_{d_i}, v_{c_{i-1}^j}) \\ &\text{subject to } dist(v_{d_i}, v_{c_{i-1}^j}) < \alpha \end{aligned} \tag{7}$$

where $dist(\cdot, \cdot)$ is the cosine distance defined as:

$$dist(x, y) = 1 - \cos(x, y) = 1 - \frac{x \cdot y}{||x||||y||} \tag{8}$$

The j-th topic cluster at time i is updated, and therefore composed by the subset of documents $D_i^j \subseteq D$, where $D_i^j = D_{i-1}^j \cup \{d_i\}$. Note that the cluster center is updated in constant time by incrementally maintaining the sum of the member embeddings.

Figure 1 illustrates an example of such clustering, using two-dimensions to represent the vectors. Let's assume document

d_1 appears in the stream first, and mentions the days Barack Obama was a senator. As it is the first document referring to the entity Obama, we add a new topic cluster $senator$ with vector v_{d_1}. Then, document d_2 appears in the stream, and refers to Obama as being elected President of the United States. The distance with the previous cluster $senator$ is greater than α due to semantic difference in the words, therefore the algorithm proceeds to create a new topic cluster $president$ centered at v_{d_2}. Finally, d_3 enters the stream. It talks about Obama as the current President of the U.S. The algorithm compares its distance to the previous clusters and finds that it is closest to the $president$ cluster. The distance is less than α, hence it adds d_3 to the $president$ cluster and updates the cluster center.

3.3. Staleness

We have been concerned with detecting whether a document d contains a novel context in terms of the documents seen so far. By representing the context of an entity as a set of topic clusters, each with an embedding vector, we are able to accurately summarize the entity context information. We expect that documents that are not close to existing clusters contain novel information. Unfortunately, this representation ignores the timeliness of the information, and it is quite possible that a document that is similar to existing clusters contains novel information. For example, when a document describes Obama victory in an election, it may be assigned to an existing cluster describing a previous election he won, nonetheless it actually contains new information.

A potential solution is to keep track of when the last document was assigned to a cluster, however, KBA challenge requires *all* documents that contain novel information within a time frame to be marked vital as per the timeliness of the document. Such timeliness is a subjective interpretation that can vary per entity and event. As an example let's assume that several documents talk about an event that happened to entity e. During a "short" time frame (here is where the subjective interpretation comes in) that information can be considered new. After a while, that new information transitions to a background state, so as the documents transition from being *vital* to *non-vital*.

In order to address such temporal dynamics that capture novelty and transition documents from a *vital* to a *non-vital* state, we propose a dynamic staleness measure λ_i, $0 < \lambda_i \leq 1$. This staleness measure can be used both for entities and topic clusters. Low staleness of the assigned entity/cluster represents *vital* documents, while high staleness intends to represent *non-vital* ones.

The staleness of an entity/cluster at any time t depends on the staleness and the time of the last document d_j assigned to the entity/cluster. The staleness decay rate is exponential,

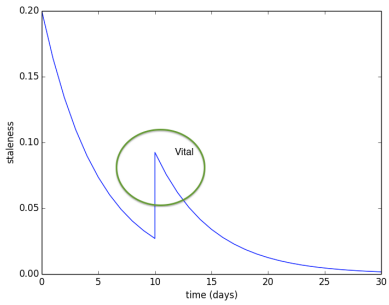


Figure 2: Staleness of Unpopular Entity

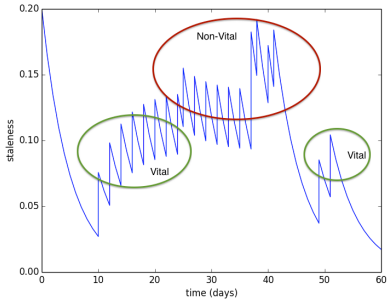


Figure 3: Staleness of Entity with Fluctuating Popularity

and is controlled by the hyperparameter γ_{dec}:

$$\lambda_t = \lambda_j \exp\left(-\gamma_{dec} \frac{t - t_j}{T}\right) \tag{9}$$

where $\gamma_{dec} \geq 0$, t_j and λ_j are the timestamp and staleness of the last document assigned to the entity/cluster, and T is a constant (used to transform the units of time).

When a new document d_i is assigned to an entity/cluster at time t_i, we can estimate the staleness of the entity/cluster at that time using the above equation, $\lambda_{t_i} = \lambda_{i-1} \exp\left(-\gamma_{dec} \frac{t_i - t_{i-1}}{T}\right)$. This staleness can be used to estimate the novelty of the information in d_i, i.e. a low λ_{t_i} suggests the document contains information that has not been observed for a while.

Thereafter, since we have just observed a relevant document for the entity/cluster, we need to increase its staleness. We use a simple interpolation to increase it:

$$\lambda_i = 1 - \gamma_{inc}(1 - \lambda_{t_i}) \tag{10}$$

where $0 \leq \gamma_{inc} \leq 1$. The staleness for the entity/cluster is now λ_i, which is used when the next document d_{i+1} is observed.

Figure 2 illustrates an example of an entity with a decreasing staleness. There are almost no documents referring to the entity. As soon as some activity is detected, i.e. a document mentioning the entity appears ($t = 10$), the staleness

increases slightly. Given the fact that there is not much information about the entity, every new document would drive an update to the entity profile, strongly suggesting vitalness.

Figure 3 represents staleness of an entity with fluctuating activity levels in the stream of documents. An important event involving the entity starts at time t=10 and continues for a substantial period, showing a growing trend in popularity. At the beginning, those documents can be considered vital, but as documents continue commenting on the same event over time, the information starts getting stale, clearly indicating non-vitalness. Near t=40 when the event is over, a steep decrease in popularity is observed. At a later time, $t = 50$, a new event occurs, strongly suggesting vitalness.

4. Visualization for Accelerate and Create

Intuitive and effective visualization techniques can provide valuable tools in assisting editors to populate entity profiles and to perform exploratory analysis of large collections of documents. In this section, we describe the requirements of such visualization tools for streaming documents. Then, we present our visualization prototype for the Accelerate and Create task that enables users to enlarge parts of the visual space while simultaneously shrinking the context, a technique called focus-plus-context (Silic & Basic, 2010).

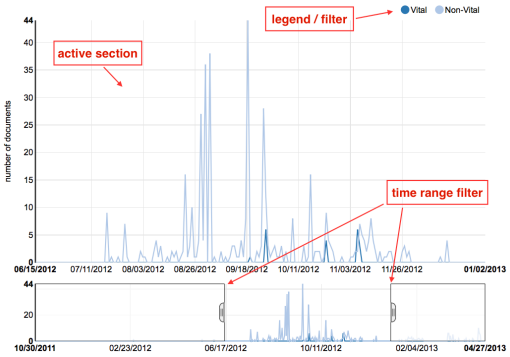
4.1. Goals and Challenges

Visual exploration of text streams is a challenging task. As text streams continuously evolve, visualization methods should allow tracing the temporal evolution of existing topics, detection of new ones, and examination of the relationships between them. Such systems should also allow users to interactively change the information they are seeking at any time. Interactivity is therefore a crucial factor in a domain where users do not know the text documents in advance (Alsakran et al., 2012).

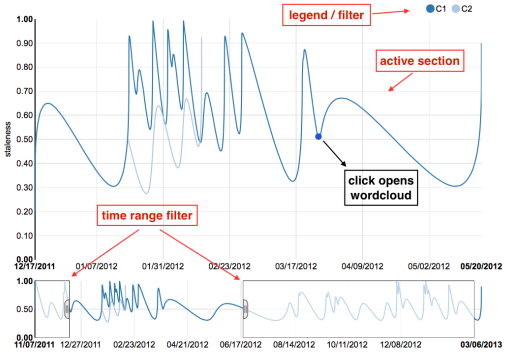
In this work we intend to provide an easy-to-use vizualization that enables users debug what is going on in the sytem. We provide different mechanisms to select data based on users interests; in particular we focus our attention on providing interactive time-series widgets.

4.2. Our Implementation

We propose a browser-based visualization prototype that enables users to switch between multiple entities of interest, select the time ranges to explore over, explore the prominence of topics over time, and understand the topics using lists of similar words. The visualization tool initiates with the user selecting an entity of interest using an autocomplete-enabled text-box. For the selected entity, our visualization consists of two views: *Document* and *Topic*.



(a) Predictions Distribution chart example



(b) Staleness chart example

(c) Topic cluster example

Figure 4: Staleness and Topic cluster evolution

The *Document* view shows the distribution of vital and non-vital documents over the complete timeline, summarizing and differentiating the time frames when the documents contain a non-vital reference to the entity, and when they contain vital information. Figure 4a illustrates the distribution of predictions of entity *Kshama Sawant* in a specific period of time. Once the interesting time frames have been identified, this view also allows the user to navigate to and read individual documents.

The *Topic* view shows the evolution of the topic clusters for the entity, illustrating the predicted proportion of topic clusters over time. This view primarily plots the staleness of a cluster over time, indicating when a cluster was started, mentioned in the documents, and fall into obsolescence. The

user can also study the topics in finer detail; clicking on any point in the timeline brings up a word cloud representation of the topic at that time. Figure 4b, for example, shows the staleness evolution for the different topic clusters of entity *Mike Kluse*. Figure 4c is the result of a user click on the point highlighted in Figure 4b. It shows the closest words to the topic cluster C1 at that time.

Both views consist of a timeline over the whole stream, allowing users to quickly navigate them over different time frames. The timeline is an active (zoomed) section that can be changed using the time range filters located below. Legends also act as filters, users have the option of observing specific clusters or predictions by selecting them in the legend. These interactive time-series controls combined with the word cloud representations allow users to explore streaming data and filter information based on their needs.

5. Related Work

Several knowledge based acceleration competitions have been done in the recent past, testifying the great progress achieved in these fields (Gross et al., 2012). Liu & Fang (2012) present one of the best performing systems in TREC KBA 2012. They created broader representations of entity profiles based on a Wikipedia snapshot and considered the anchor text of all internal Wikipedia links as related entities. In TREC KBA 2013 competition, different families of methods were proposed, including query expansion, classification, and learning to rank.

Our strategy is somewhat similar to Wang et al. (2013) in the sense that we first target a high recall system and then apply different classification methods to differentiate between *vital* and *non-vital* documents. One key difference is that we do not exploit any external resources to construct features, e.g. we do not use Wikipedia entity pages nor existing citations in the Wikipedia page of an entity.

Representing words as continuous vectors has been studied for a number of years (Hinton et al., 1987; Elman, 1990). The progress in machine learning techniques in recent years has enabled training more complex models on much larger data sets (Mikolov et al., 2013a). One popular approach to improving accuracy by exploiting large datasets is to use unsupervised methods to create word features, or to download word features that have already been produced (Turian et al., 2010). In our method, we do the latter, using already induced word embedding features in order to improve our system accuracy. To the best of our knowledge, no other technique has proposed the use of word embeddings representations for the vital filtering task.

One of the pioneering work on detecting novel documents was introduced by Zhang et al. (2002). They explicitly model relevance and redundancy as separate concepts. They propose different redundancy measures and empirically show that the cosine similarity metric is effective in identifying redundant documents; one limitation is that they just keep only the 10 most recent documents for a profile. In our method, we summarize the complete history of documents for a given entity, which allows a more accurate estimate of the query document redundancy. Gamon (2006) addresses the problem of staleness detection by building an association graph that connects sentences and sentence fragments, and uses graph-based features as indicators of lack of novelty. Though the task is somewhat similar, it is more limited in the sense that they do not need to model the transition from new to background information.

Many of the recent approaches have focused on scaling novel detection algorithms, also known as First Story Detection, in the streaming setting, by either using LSH (Petrović et al., 2010) or just employing simple heuristics (Luo et al., 2007). While their work mainly focuses on efficiency at the cost of accuracy, our work aims to achieve accurate representations without compromising efficiency.

Streaming document filtering is also related to several other fields, including but not limited to, entity linking (Ji & Grishman, 2011), text categorization (Kjersten & McNamee, 2012), news surveillance (Steinberger, 2014), and cross-document coreference (Rao et al., 2010; Singh et al., 2011).

6. KBA Vital Filtering Evaluation

In this section we describe the TREC KBA Vital filtering task, and our evaluation setup.

6.1. Data

To assess our contributions we use TREC KBA 2014 filtered stream corpus. The filtered corpus contains around 20M documents annotated with BBN's Serif NLP tools, including within-doc coreference and dependency parse trees. Further, we use the 71 target entities given by KBA organizers for the Vital Filtering task. Among the 20M documents, around 28K have truth labels. From these labeled example, 8K are treated as training instances, while the rest as test instances. We preprocess the corpus to retain only the documents that contain exact string matches to the target entities names, including canonical and surface form names.

6.2. Features

Our approach extends the classifier introduced by Wang et al. (2013). We construct a basic set of features based on the document and the entity of interest. Using our representation, we include additional features for the embedding, clustering, and staleness. A summary of the features we use is presented in Table 1.

Basic Features, F_b	
Based on document d	
$\log(len(d))$	log of the length of d
$source(d)$	discretized source of d
Based on document d and target entity e	
$n(d, e)$	# of occurrences of target entity e in d
$n(d, e^p)$	# of occurrences of partial name of e in d
$\text{fpos}(d, e)$	position of first occurrence of entity e in d
$\text{fpos}_n(d, e)$	$\text{fpos}(d, e)$ normalized by document length
$\text{fpos}(d, e^p)$	position of first partial occurrence of e in d
$\text{fpos}_n(d, e^p)$	$\text{fpos}(d, e_p)$ normalized by document length
$\text{lpos}(d, e)$	position of last occurrence of entity e in d
$\text{lpos}_n(d, e)$	$\text{lpos}(d, e)$ normalized by document length
$\text{lpos}(d, e^p)$	position of last partial occurrence of entity e in d
$\text{lpos}_n(d, e^p)$	$\text{lpos}(d, e^p)$ normalized by document length
$\text{spread}(d, e)$	$\text{lpos}(d, e) - \text{fpos}(d, e)$
$\text{spread}_n(d, e)$	$\text{spread}(d, e)$ normalized by document length
$\text{spread}(d, e^p)$	$\text{lpos}(d, e^p) - \text{fpos}(d, e^p)$
$\text{spread}_n(d, e^p)$	$\text{spread}(d, e^p)$ normalized by document length
Embedding Features, F_e	
Based on a single, combined embedding, F_e^c	
v_d	mean word embedding representation of d
$\text{zero}(v_d)$	$\mathbb{1}_{v_d=0}$, set to 1 if v_d is 0
Based on POS embeddings, F_e^p	
v_{d_n}	mean word embedding for common nouns
$\text{zero}(v_{d_n})$	$\mathbb{1}_{v_{d_n}=0}$, set to 1 if v_{d_n} is 0
v_{d_N}	mean word embedding for proper nouns
$\text{zero}(v_{d_N})$	$\mathbb{1}_{v_{d_N}=0}$, set to 1 if v_{d_N} is 0
v_{d_v}	mean word embedding for verbs
$\text{zero}(v_{d_v})$	$\mathbb{1}_{v_{d_v}=0}$, set to 1 if v_{d_v} is 0
Clustering Features, F_c	
$\min_c(v_d, v_c)$	minimum distance of v_d to topic clusters of e
$\text{avg}_c(v_d, v_c)$	average distance of v_d to topic clusters of e
Temporal Features, F_t	
$\lambda(e)$	current staleness of entity e
$\lambda(e, c)$	current staleness of topic c of target entity e

Table 1: Features for Vital Filtering classification

6.3. Relevance Classification

TREC KBA 2014 corpus contains documents that do not refer to the target entities, even though they may contain mentions to them. We therefore need to use a *non-referent* category of documents. A *non-referent* document denotes that it does not refer to a target entity or the context is so ambiguous that it is impossible to decide whether the mention refers to an entity or not. An example of the former case is "Barack Ferrazzano provides a wide range of business-oriented legal". It clearly does not refer to Barack Obama. For the latter, an example is "Barack is a great father and a better husband". The mention "Barack" may refer to any married parent named Barack, therefore, we consider it *non-referent*. The *vital* and *non-vital* classes described in section 2 fall into a *referent* (or *relevant*) category, which contains documents that refer to the target entities.

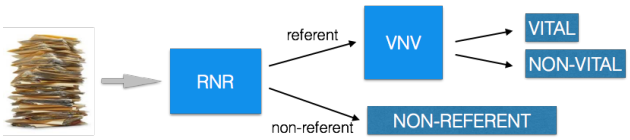


Figure 5: Classification process

Due to the fact that not all documents in the corpus refer to the target entities, we include an extra step in our classification process, as shown in Figure 5. We introduce an additional classifier, called rnr, which is trained offline and classifies documents as *referent* or *non-referent*. Consequently, in every experiment, each document goes first through the *rnr* classifier. Only the *referent* documents outputted by *rnr* are used as inputs to the *vnv* classifier, which discriminates between *vital* and *non-vital* documents, the overall focus of this work.

We use randomized tree ensembles classifiers (Geurts et al., 2006) for both *rnr* and *vnv*, each composed of 100 weak learners. The maximum depth of each tree in the ensembles is 150. All our experiments use the same *rnr* model trained with the basic features listed in section 6.2. The different methods differ in the features used in the *vnv* classifier.

6.4. Methods

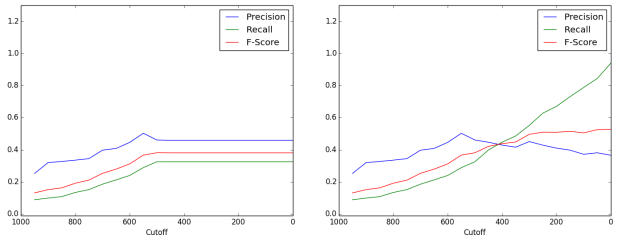
We evaluate the following approaches in our experiments. To compare against existing baselines, we use just the F_b features (**Baseline**); Wang et al. (2013) and Bellogín et al. (2013) propose a similar technique, although they train their models with a few more features. We also include an additional baseline that uses multi-task learning (Caruana, 1993) to learn separate parameters for each entity, called **Baseline, Multi-task**. In order to evaluate the effect of adding word embeddings, we introduce two extensions to the baselines that use the embedding features: **Embedding, Single** that uses a single embedding for every document (F_e^c features), and **Embedding, POS** that maintains different embeddings for common nouns, proper nouns and verbs (F_e^p features); see Section 3.1 for details. We separately evaluate the utility of temporal modeling via staleness by introducing the **Staleness only** method that includes the F_t features. Similarly, we propose the method that uses only clustering (F_c features), but not the temporal ones, called **Clustering only**. Finally, the approach that combines all contributions, **Combined**, includes all the F_b, F_e^p, F_t, and F_c features.

7. Results and Discussion

Table 2 shows the submitted and revised precision, recall and F1 results of the methods explained in 6.4, computed using KBA scorer tool, using the 2014-07-11 truth data. The models that include the F_c features use $\alpha = 0.8$, whereas

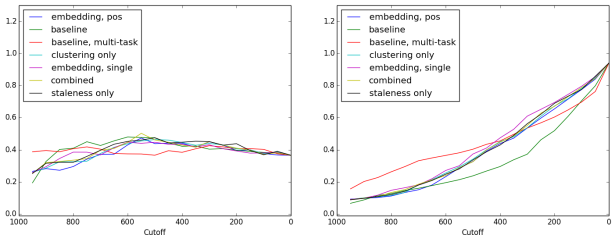
Model	Features	Vital only, *micro*						Vital only, *macro*					
		P		R		F1		P		R		F1	
Baseline	F_b	*53.9*	26.6	*26.5*	63.4	*35.5*	37.5	*47.5*	36.6	*23.8*	94.0	*31.7*	52.7
Baseline, Multi-task	F_b	*60.7*	60.7	*41.4*	41.4	*49.2*	49.2	*36.7*	36.6	*40.5*	94.0	*38.5*	52.7
Embedding, Single	$F_b + F_e^c$	*54.7*	54.7	*52.1*	51.8	*53.4*	53.2	*44.9*	38.3	*37.6*	85.5	*40.9*	52.9
Embedding, POS	$F_b + F_e^p$	*53.9*	49.9	*46.4*	53.1	*49.8*	51.4	*44.0*	36.6	*32.9*	94.0	*37.6*	52.7
Staleness only	$F_b + F_e^p + F_t$	*57.3*	57.3	*48.3*	48.3	*52.4*	52.4	*47.5*	39.1	*33.8*	85.8	*39.5*	53.7
Clustering only	$F_b + F_e^p + F_c$	*57.0*	57.0	*49.0*	48.9	*52.7*	52.6	*46.4*	38.7	*34.2*	85.0	*39.4*	53.2
Combined	$F_b + F_e^p + F_c + F_t$	*56.2*	56.2	*48.1*	48.1	*51.8*	51.8	*46.1*	36.6	*32.6*	94.0	*38.2*	52.7

Table 2: Vital Filtering performance using the *submitted* and revised runs for TREC KBA 2014

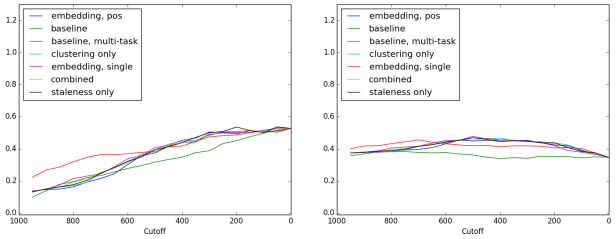


(a) Example Submission Curve (b) Example Revised Curve

Figure 6: Examples of P-R-F1 over confidence cutoffs



(a) Precision (b) Recall



(c) F1 (d) Scaled Utility

Figure 7: Macro P-R-F1-SU over confidence cutoffs

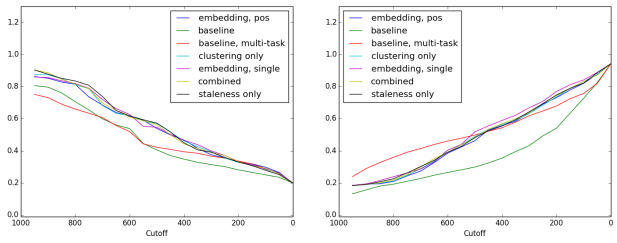
the models that include the F_t features use $\gamma_{dec} = 1$ and $\gamma_{inc} = 0.1$.

According to the official results, our submissions achieved the 2^{nd} best precision in the competition, but performed poorly in the overall macro F1 (8^{th} position). Revisiting our submission files, we found that we misinterpreted the concept of confidence. Figure 6a shows that we only make vital predictions with confidence greater than or equal to 500, i.e. the right part of the curve is just constant. We should have also predicted vital with low confidence, i.e. flip our high confidence non-vital predictions to be vital with low confidence. That minor change boosts our recall (in most cases), while the precision slightly suffers, as shown in Figure 6b, leaving our system in the 2^{nd} overall position.

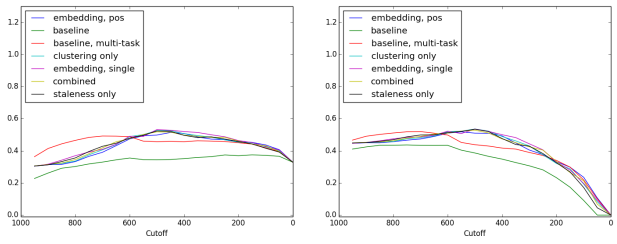
The baseline provided by TREC KBA organizers (not to be confused with our *Baseline* model) assigns a vital rating to every document that matches a surface form name of an entity, assigning a confidence score based on the number of matches of tokens in the name. The values reported by the organizers are: macro-P=0.316, macro-R=0.520, macro-F1=0.393, SU=0.3334 (Frank et al., 2014).

Baseline performs as expected, i.e. has lower F1 than the other models. On the other hand, *Baseline, Multi-task* performs far better than *Baseline*, which suggests the contexts vary sufficiently across entities to benefit from separate parameterization. Our proposed models further improve upon the *Baseline, Multi-task*. Further, using a single embedding (*Embedding, Single*) outperforms multiple embeddings representations (*Embedding, POS*), indicating word embeddings implicitly capture the various parts of speech in their representation. The proposed staleness and non-parametric clustering (*Staleness only*, *Clustering only*, *Combined*) perform slightly worse than the simple *Embedding, Single* method on the submission results. However, in the revised versions, *Staleness only* scores the best F1, followed closely by *Clustering only*, which further demonstrates the importance of F_t and F_c features.

Figures 7 and 8 complement the revised results in Table 2 for different confidence cutoffs. Figures 7a and 7b show that the macro recall has a substantial increase in the lower confidence half of the plot, while retaining most of the precision; this results in a boost to F1 scores in the revised macro scenario. For micro-averaging on the other hand, the precisions and recalls (shown in Figures 8a and 8b) follow opposite trends, causing the micro revised F1s to be almost



(a) Precision (b) Recall



(c) F1 (d) Scaled Utility

Figure 8: Micro P-R-F1-SU over confidence cutoffs

the same as the micro submission ones.

Figures 9 and 10 further illustrate the precision-recall for the different methods. For macro averaging, all methods perform almost the same. The micro metrics in Figure 10 are much more interesting. At lower recall, the micro precisions of the different models meet our expectations: the more complex methods, including non-parametric clustering and staleness, outperform our baselines. Nevertheless, at higher recall, *Embedding, Single* takes the lead.

8. Conclusion & Future Work

Filtering streaming documents in order to fill gaps plays a crucial role in the maintenance and timely updates of knowledge bases. With the exponential increase of information on the web, it becomes critical to detect relevant documents and incorporate their information in a timely manner. In this paper we introduced a semi-supervised learning model for document filtering tasks. We proposed a word embeddings based non-parametric representation of documents that groups entity references into topic clusters, and is suitable for streaming data. Further, we present a notion of staleness for entities and topics that dynamically estimates the temporal relevance of the entity contexts. The combination of these three core contributions (distributed word embedding representations, non-parametric clustering, and staleness) results in an accurate representation of entity contexts, while simultaneously addressing the filtering requirements of large corpora of streaming text documents.

A number of avenues exist for further work. A possible line of future research would be exploring hierarchical clustering

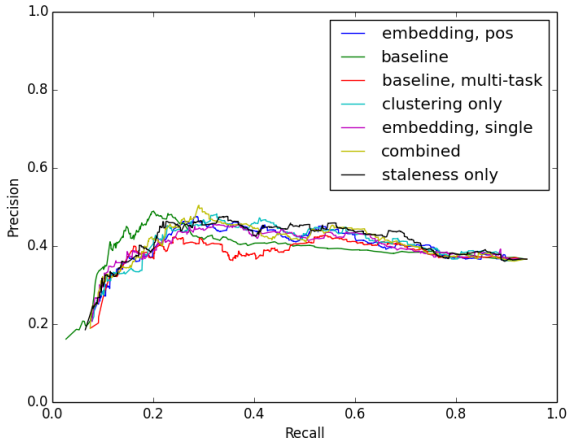


Figure 9: Macro Precision-Recall

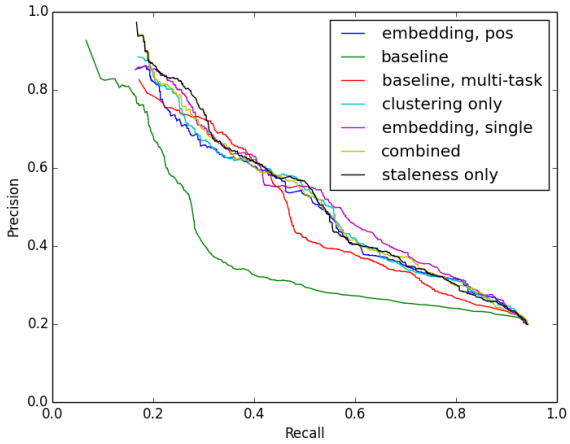


Figure 10: Micro Precision-Recall

algorithms to better represent topic clusters. It would also be interesting to assess the effect of learning the hyperparameters of the model instead of just manual tuning them for the specific datasets. Utilizing external resources such as Wikipedia entity pages to construct more features (Liu & Fang, 2012) will likely further improve the accuracy of our method. It would also be worthwhile to assess the effects of using different pre-trained word embeddings.

Acknowledgments

This work was supported in part by the Argentine Ministry of Science, Technology and Productive Innovation with the program BEC.AR, and in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

Alsakran, Jamal, Chen, Yang, Luo, Dongning, Zhao, Ye, Jing, Yang, Dou, Wenwen, and Liu, Shixia. Real-Time Visualization of Streaming Text with a Force-Based Dynamic System. *IEEE Computer Graphics and Applications*, pp. 34–45, 2012.

Bellogín, Alejandro, Gebremeskel, Gebrekirstos G., He, Jiyin, Lin, Jimmy, Said, Alan, Samar, Thaer, de Vries, Arjen P., and Vuurens, Jeroen B. P. CWI and TU Delft at TREC 2013: Contextual Suggestion, Federated Web Search, KBA, and Web Tracks. In *Proceedings of the Twenty-Second Text REtrieval Conference (TREC)*, 2013.

Blei, David. Probabilistic topic models. *Communications of the ACM*, pp. 7784, 2012.

Bouvier, Vincent and Bellot, Patrice. Filtering Entity Centric Documents using Numerics and Temporals features within RF Classifier. In *Proceedings of the Twenty-Second Text REtrieval Conference (TREC 2013)*, 2013.

Caruana, Richard. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 41–48, 1993.

Efron, Miles, Willis, Craig, Organisciak, Peter, Balsamo, Brian, and Lucic, Ana. The University of Illinois Graduate School of Library and Information Science at TREC 2013. In *Proceedings of the Text REtrieval Conference (TREC)*, 2013.

Elman, Jeffrey L. Finding structure in time. *COGNITIVE SCIENCE*, pp. 179–211, 1990.

Frank, John R., Kleiman-Weiner, Max, Roberts, Daniel A., Niu, Feng, Zhang, Ce, and Ré, Christopher. Building an Entity-Centric Stream Filtering Test Collection for TREC 2012. In *Proceedings of the Twenty-First Text REtrieval Conference (TREC 2012)*, 2012.

Frank, John R., Kleiman-Weiner, Max, Roberts, Daniel A., Voorhees, Ellen, and Soboroff, Ian. Evaluating Stream Filtering for Entity Profile Updates in TREC 2012, 2013, and 2014. In *KBA Track Overview, Notebook Paper*, 2014.

Gamon, Michael. Graph-Based text Representation for Novelty Detection. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, 2006.

Geurts, Pierre, Ernst, Damien, and Wehenkel, Louis. Extremely Randomized Trees. *Machine Learning*, pp. 3–42, 2006.

Gross, Oskar, Doucet, Antoine, and Toivonen, Hannu. Term Association Analysis for Named Entity Filtering. In *Proceedings of the Twenty-First Text REtrieval Conference (TREC 2012)*, 2012.

Hinton, G. E., McClelland, J. L., and Rumelhart, D. E. Distributed Representations. In *Parallel Distributed Processing: Volume 1: Foundations*, pp. 77–109. 1987.

Ji, Heng and Grishman, Ralph. Knowledge Base Population: Successful Approaches and Challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pp. 1148–1158, 2011.

Kjersten, Brian and McNamee, Paul. THE HLTCOE APPROACH TO THE TREC 2012 KBA TRACK. In *Proceedings of the Twenty-First Text REtrieval Conference (TREC)*, 2012.

Liu, Xitong and Fang, Hui. Entity Profile based Approach in Automatic Knowledge Finding. In *Proceedings of the Twenty-First Text REtrieval Conference (TREC)*, 2012.

Liu, Xitong, Darko, Jerry, and Fang, Hui. A Related Entity based Approach for Knowledge Base Acceleration. In *Proceedings of the Text REtrieval Conference (TREC 2013)*, 2013.

Luo, Gang, Tang, Chunqiang, and Yu, Philip S. Resource-adaptive real-time new event detection. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07. ACM, 2007.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeff. Efficient estimation of word representations in vector space. *CoRR*, 2013a.

Mikolov, Tomas, tau Yih, Wen, and Zweig, Geoffrey. Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*, pp. 746–751, 2013b.

Neelakantan, Arvind, Shankar, Jeevan, Passos, Alexandre, and McCallum, Andrew. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. *EMNLP*, 2014.

Petrović, Saša, Osborne, Miles, and Lavrenko, Victor. Streaming first story detection with application to twitter. In *Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

Rao, Delip, McNamee, Paul, and Dredze, Mark. Streaming Cross Document Entity Coreference Resolution. In *COLING (Posters)*, pp. 1050–1058, 2010.

Silic, Artur and Basic, Bojana Dalbelo. Visualization of Text Streams: A Survey. *Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 31–43, 2010.

Singh, Sameer, Subramanya, Amarnag, Pereira, Fernando, and McCallum, Andrew. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Association for Computational Linguistics (ACL)*, 2011.

Steinberger, Ralf. A survey of methods to ease the development of highly multilingual text mining applications. *CoRR*, 2014.

Turian, Joseph, Ratinov, Lev, and Bengio, Yoshua. Word representations: A simple and general method for semisupervised learning. In *ACL*, pp. 384–394, 2010.

Wang, Jingang, Song, Dandan, Liao, Lejian, and Lin, Chin-Yew. BIT and MSRA at TREC KBA CCR track 2013. In *Proceedings of the Text REtrieval Conference (TREC)*, 2013.

Zhang, Chunyun, Xu, Weiran, Liu, Ruifang, Zhang, Weitai, Zhang, Dai, Ji, Janshu, and Yang, Jing. PRIS at TREC2013 Knowledge Base Acceleration Track. In *Proceedings of the Twenty-Second Text REtrieval Conference (TREC)*, 2013.

Zhang, Yi, Callan, Jamie, and Minka, Thomas. Novelty and redundancy detection in adaptive filtering. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pp. 81–88. ACM, 2002.