Multi-step Classification Approaches to Cumulative Citation Recommendation

Krisztian Balog
University of Stavanger
krisztian.balog@uis.no

Naimdjon Takhirov
NTNU Trondheim
naimdjon.takhirov@idi.ntnu.no

Heri Ramampiaro
NTNU Trondheim
heri.ramampiaro@idi.ntnu.no

Kjetil Nørvåg
NTNU Trondheim
kjetil.norvag@idi.ntnu.no

ABSTRACT

Knowledge bases have become indispensable sources of information. It is therefore critical that they rely on the latest information available and get updated every time new facts surface. Knowledge base acceleration (KBA) systems seek to help humans expand knowledge bases like Wikipedia by automatically recommending edits based on incoming content streams. A core step in this process is that of identifying relevant content, i.e., filtering documents that would imply modifications to the attributes or relations of a given target entity. We propose two multi-step classification approaches for this task that consist of two and three binary classification steps, respectively. Both methods share the same initial component, which is concerned with the identification of entity mentions in documents, while subsequent steps involve identification of documents being relevant and/or central to a given entity. Using the evaluation platform of the TREC 2012 KBA track and a rich feature set developed for this particular task, we show that both approaches deliver state-of-the-art performance.

Categories and Subject Descriptors

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

Keywords

Knowledge base acceleration, cumulative citation recommendation, information filtering

1. INTRODUCTION

Knowledge bases (KB) have become indispensable sources of information. Wikipedia, specifically, has been widely utilised in various information access contexts. Some common uses include named entity recognition and disambiguation [9, 11, 23], query modeling and expansion [28, 45], question answering [1], entity linking [20, 31], and entity retrieval [5, 13]. In many of these cases, the role of Wikipedia is to serve as a "semantic backbone," a repository of entities and their relations. While undoubtedly the most popular, Wikipedia is not unique in this capacity; recent developments in the Web of Data enable the use of domain-specific knowledge [7]. Alternatively, legacy or corporate knowledge bases can also be used to provide entities [21].

Keeping up with changes and maintaining up-to-date knowledge is in everyone's best interest. It, however, requires a continuous effort to be spent by editors and content managers, and is becoming increasingly demanding as information is being produced at an ever-growing rate. *Knowledge base acceleration* (KBA) systems seek to help humans expand knowledge bases like Wikipedia by automatically recommending edits based on incoming content streams. Identifying content items (news articles, blog posts, etc.) that may imply modifications to the attributes or relations of a given target entity is one of the basic steps to be performed by any KBA system. The Text REtrieval Conference (TREC) has launched a new Knowledge Base Acceleration track (TREC KBA[1]) in 2012, which focused on this very problem, termed as *cumulative citation recommendation* (CCR): filter a time-ordered corpus for documents that are highly relevant to a predefined set of entities [16].

A particularly challenging aspect of the CCR task is to draw a distinction between documents that are *relevant* and the ones that are *central*. An informal requirement for centrality is that the document relates directly to the target entity such that one would cite it in the Wikipedia article of that entity. As a general rule of thumb, this suffices for human annotators in the majority of cases, but it is not a precise definition that can easily be captured algorithmically. Our main research questions in this paper concern the development of a classification pipeline and the design, evaluation, and analysis of appropriate features for this task.

CCR can naturally be viewed as a binary classification problem. The difficulty arises, however, that computing a potentially large set of features for every single document-entity pair is not feasible at scale—a more efficient solution is required. Our proposed approach consists of multiple binary classification steps. It starts with an entity detection step that uses known surface forms of the entity to radically reduce the number of documents to work with, without sacrificing recall. Then, we consider two possible continuations of the pipeline. According to the first method, we have a single subsequent step in which we decide whether the document is central or not; we refer to this as the *2-step approach*. Our second method splits centrality detection into two steps: first separating non-relevant from relevant and then separating relevant from central documents; we refer to this as the *3-step approach*.

We consider four types of features for this particular task: (1) *document features* estimate the "citation worthiness" of the document

OAIR'13, May 22-24, 2013, Lisbon, Portugal.

[1] http://trec-kba.org

on its own account, irrespective of the target entity; (2) *entity features* are based solely on the target entity; (3) *document-entity features* capture the relation between a particular document and the target entity as well as between the document and other entities related to the target; (4) *temporal features* aim to determine points in time where important events for the target entity happen, based on the volume of entity mentions in the stream as well as based on Wikipedia usage data.

To summarise, we make the following contributions: (1) we address the task of centrality detection in a KBA context and propose two multi-step classification approaches; (2) we develop four main types of features to allow for automatic classification of documents; (3) we perform a thorough experimental evaluation and comparison of the two strategies, followed by an analysis of proposed features, using the evaluation platform of the TREC 2012 KBA track. The resources used in this paper (entity name variants, features, relevance rankings, and evaluation results) are made publicly available at http://bit.ly/13VF54D.

The remainder of the paper is organised as follows. In the next section we briefly review prior work from related areas. In Section 3 we introduce the problem and data set we are studying. Next, in Section 4, we present two multi-step classification approaches. In Section 5, we introduce and discuss our features. This is followed by our experimental evaluation in Section 6, and by further analysis in Section 7. We formulate our conclusions in Section 8.

2. RELATED WORK

Constructing a knowledge base from the information provided in a text collection is an effort usually referred to as *knowledge base population* (KBP). The Never-Ending Language Learner (NELL)[2] [10] and Open Information Extraction (OpenIE)[3] [15] are two interesting research efforts in this direction. The main idea behind both NELL and OpenIE is to extract structured information from unstructured web pages and then build a knowledge base from the extracted data. Another related project is YAGO [19, 40] that constructs a knowledge base by aggregating information from multiple sources, such as Wikipedia, Wordnet,[4] and Geonames.[5] In response to the growing interest in the subject, the Text Analysis Conference (TAC) introduced a dedicated Knowledge Base Population track in 2009 [22]. Three key components of KBP are addressed: entity-linking, slot-filling, and cold start KBP. Out of these, entity linking bears the most relevance to our task and we shall discuss it next.

The entity linking task is defined as follows: given an entity and a document containing a mention of the entity, identify and link the corresponding node in the knowledge base. This task has been studied extensively previously in both monolingual [14, 38] and cross-lingual [27, 42] contexts. Another related evaluation effort is the Link-the-Wiki track that ran at INEX with the aim of developing a standard methodology for the evaluation of link discovery [20]. Given a text document, the goal is to recommend a set of incoming and outgoing links from anchor text to the best entry point in other documents in the collection. Wikify! [31] and Wikipedia-Miner [32] take these approaches one step further by performing both the detection of entity mentions and their linking to the corresponding Wikipedia article. Meij et al. [29] propose an approach to analyse queries submitted to search engines, automatically identify concepts that are related to the queries, and then link the queries to DBpedia[6] entries. In another work, Meij et al. [30] suggest a method to automatically identify what microblog posts are about by first determining concepts that are related to them and then generating corresponding links to Wikipedia articles. The CCR task we are investigating inherently has an entity identification element to it; this, we address as a separate step. Entity disambiguation, however, is addressed implicitly, as part of the centrality detection.

Information filtering is a research area that is also closely related to our work. It refers to managing large information flows from a stream of documents with the ultimate goal to capture information about the user's interest and use it to provide an improved service to her [34]. Typical early approaches represent the user's interests by a list of "profile queries" [18] or treat information filtering as a specialised text classification task [33]. Later methods range from network-based profiling [35] to personalised delivery of microblog messages to users [39]. CCR differs from traditional filtering tasks in two main aspects. First, topics are entities, described by semistructured articles in a knowledge base. Second, the stream nature of the task implies that entities may evolve over time and a previously non-relevant document may be relevant at a later time.

Topic detection and tracking (TDT) is concerned with the development of techniques for finding and following events in broadcast news stories. Specific tasks include novelty detection (detecting new events), topic tracking (monitoring events throughout time), and topic detection (organising news stories as they arrive) [2]. The main focus in this context is on discovering and threading topically related content in data streams. From a sufficient distance, both TDT and CCR are about identifying "interesting" documents in a stream corpus. Nevertheless, TDT has a very strong focus on news events as topics [3], while CCR attempts to make fine-grained distinctions between relevant and central documents.

A recent development was the introduction of the Knowledge Base Acceleration (KBA) track at TREC 2012 [16]. Its main goal is the development of filtering and recommender systems that can aid human curators in their task of maintaining high quality and up-to-date knowledge bases. Participating systems either approached the CCR task as a ranking [6, 17, 25, 41] or as a classification [6, 8, 24] problem (using either SVM [24] or Random Forest [6, 8] classifiers). The only work that uses a multi-step method is by Bonnefoy et al. [8]. In this paper, we compare two different multi-step methods and use a richer feature set.

3. PROBLEM AND DATA DESCRIPTION

Our work is guided by the need to maintain the accuracy and high quality of knowledge bases. Time, new facts and discoveries may turn the content outdated or inaccurate. We wish to develop automated methods that allow editors and content managers to discover and process new information as it becomes available. We base our investigations on the task definition, data set, and manual annotations provided by the TREC 2012 KBA track [16].

Task. The *cumulative citation recommendation* (CCR) task is defined as follows: given a textual stream consisting of news and social media content, and an input entity from a knowledge base (Wikipedia), generate a score for each document based on how pertinent it is to the target entity.

Data collection. The data set that has been built for evaluation purposes is called the KBA Stream Corpus 2012;[7] covering the time period from October 2011 to April 2012, it is composed of three sources:

[2] http://rtw.ml.cmu.edu/rtw/.

[3] http://ai.cs.washington.edu/projects/open-information-extraction.

[4] http://wordnet.princeton.edu/

[5] http://www.geonames.org/

[6] http://dbpedia.org

[7] http://trec-kba.org/kba-stream-corpus-2012.shtml

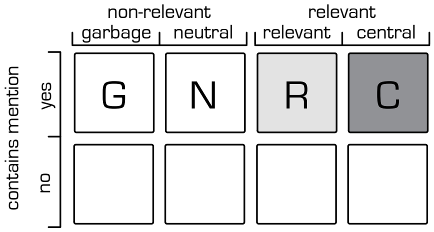


Figure 1: Document annotation matrix from the TREC 2012 KBA track. The goal of the CCR task is to identify central documents, i.e., the ones in the top right corner.

- **News**: global public news wires.
- **Social**: blogs and forums.
- **Linking**: content from URLs shortened at `bitly.com`.

Each stream item (i.e., URL) is time-stamped, uniquely identified by a `stream_id`, and has its content fetched. We will refer to these as *stream documents* (or *documents* for short) from now on. We work with the boilerplate cleansed version of the corpus, i.e., the cleansed-only version, but we do not use the provided named entity annotations. For *social* documents the content is further separated into title, body, and anchor fields (for *news* and *linking* only the body field is available).

Topics. The topic set consists of 29 entities (27 persons and 2 organisations), referred to as *target entities*. These are described by semistructured articles in a knowledge base, specifically, Wikipedia. Each of these entities is identified uniquely by a `urlname`. Target entities were chosen such that they receive a moderate number of mentions in the stream corpus: between once per day and once per week. The focus was on entities with complex link graphs of relationships with other active entities.

Annotation. Annotations are provided along two dimensions: contains mention and relevance. The annotation matrix is shown in Figure 1. Rows denote whether the document mentions the target entity explicitly (top) or not (bottom). Columns indicate the level of relevance, which is judged on a 4-point scale:

- **Garbage**: not relevant; e.g., spam.
- **Neutral**: not relevant; nothing can be learned about the target entity.
- **Relevant**: relates indirectly to the target entity, e.g., mentions topics or events that are likely to have an impact on the entity.
- **Central**: relates directly to the target entity, e.g., the entity is a central figure in the mentioned topics or events.

Note that a document can be relevant, even if it does not mention the target directly (through relations to other entities mentioned in the document). Relevance without an explicit mention of the target entity is, however, a rare case;[8] centrality without an explicit mention never happens. Therefore, we will only be focusing on documents with explicit mentions, i.e., the top row in Figure 1.

TREC KBA provides training annotation data, i.e., assessor judgements along the two dimensions just discussed, for corpus documents from the October to December 2011 period. Documents from the January to April 2012 period are used for testing. We follow this setup, i.e., we only use pre-2012 documents for training.

Table 1: Entity surface forms for mentions detection.

urlname	DBpedia	DBpedia-loose
Aharon Barak	Aharon Barak Aaron Barak	Barak
Lovebug Starski	Lovebug Starski Love Bug Starski DJ Luv Bug Starski Lovebug Starsky Love Bug Starsky	Starski Starsky

Topics were judged by a total of six annotators, where each topic was in most cases annotated by only one or two assessors. The inter-annotator agreement on mention vs. non-mention was 97%, while on relevance ratings it was considerably lower, around 70%. Annotator disagreements are resolved during scoring by taking the lowest rating for a given entity-document pair.

Task structure. The aim for systems performing the CCR task is to replicate the *central* judgment, that is, to propose documents that a human would want to cite in the Wikipedia article of the target entity.[9] Participating systems are required to process the corpus in hourly batches in chronological order. For each hour, systems must emit a list of documents for each target entity, where documents are assigned a confidence score in the range of $(0, 1000]$.

4. MULTI-STEP CLASSIFICATION

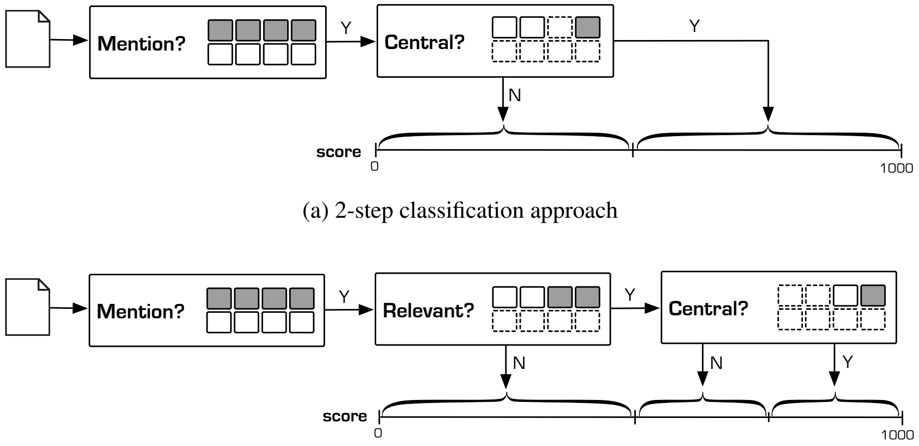
Ultimately, the question we wish to answer for each stream document is this: Is this document worth citing in the entity's Wikipedia article? In other words: Is this document central for this entity? This problem can naturally be cast as a binary classification task. One main question that arises here is the choice of features; we discuss this in Section 5. Assuming for now that we have our features (a potentially large set) specified, it becomes immediately apparent that computing them for every single document-entity pair is not feasible at scale—a more efficient solution is required. We already know from the previous section that central documents always contain an explicit mention of the target entity. Therefore, we develop a multi-step approach that starts with an entity detection step; this component is discussed in Section 4.1. We propose two alternative routes for continuing the processing pipeline. According to the first method, we only have a single subsequent step where we decide whether the document is central or not; we refer to this as the 2-step approach (Section 4.2). Our second method splits centrality detection into two steps: first separating non-relevant from relevant and then separating relevant from central documents; we refer to this as the 3-step approach (Section 4.3). Figure 2 illustrates the 2-step and 3-step approaches, respectively.

4.1 Identifying entity mentions

This component is responsible for determining whether a given document possibly contains (a mention of) a specific target entity. This can easily be seen as a binary classification task, where our main interest is in recall. On the other hand, we also wish to keep the number of false positives reasonably low (not only to save computational resources, but more importantly, to prevent propagating

[8] This happens in one in twenty cases for real citations in Wikipedia, where most of these are more properly viewed as citations for related entities that do not yet have a separate Wikipedia article and are described in a section of an existing article.

[9] For the 2013 edition of the track "relevant" and "central" have been renamed to "useful" and "vital," to better capture the two operationally different notions of citation worthiness; nevertheless, in this paper, we stick to the TREC 2012 KBA terminology.



(a) 2-step classification approach

(b) 3-step classification approach

Figure 2: Multi-step classification approaches. The two-by-four matrixes correspond to the annotation matrix in Figure 1. Squares with solid borders represent the annotations targeted in that step; no fill means the negative class, fill means the positive class.

errors throughout the entire processing pipeline). Finally, we have a strong need for efficiency here, as this step has to be performed for each document-entity pair.

Our solution is to represent each target entity as a set of its surface forms, that is, the names by which this entity is being referred to. Having this set $names(E)$ constructed, the classification works as follows:

$$mentions(D, E) = \begin{cases} 1, & \exists E_n \in names(E) : contains(D, E_n) \\ 0, & \text{otherwise,} \end{cases}$$

where $contains(haystack, needle)$ denotes a case insensitive string matching function.

Like many other participants at TREC KBA [4, 8, 12], we use DBpedia for extracting entity name variants. We consider the following three ways of constructing the set of surface forms; see Table 1 for examples.

- **urlname**: the `urlname` of the entity after basic cleaning; that is, we replace "_" with space and remove text in between brackets (e.g., `Basic_Element_(music_group)` becomes "Basic Element").
- **DBpedia**: known name variants of the entity from DBpedia. Specifically, we used DBpedia version 3.7 that is based on Wikipedia dumps generated in late July 2011[10] (which is before the start date of the stream corpus).
- **DBpedia-loose**: again, we take name variants from DBpedia, but for entities of type person we only consider their last names.

Note that we do not perform entity disambiguation explicitly; this is handled implicitly (as part of the centrality detection) in the subsequent step(s).

4.2 2-step approach

Under this approach (Figure 2(a)) we attempt to classify the document as central or not in a single step using the features discussed in Section 5. The feature set and the actual classifier used are treated as a black box at this point. We use document-entity pairs labeled as garbage (G) or neutral (N) as negative examples and central (C) ones as positive examples. We do not use instances labeled as relevant (R) here at all, as it would soften the distinction between the two classes that we are trying to separate.

According to the CCR task setup, each document-entity pair needs to have a score assigned in the $(0, 1000]$ range. We map the negative (non-central) predictions to the $(0, 500]$ range and the positive (central) predictions to the $(500, 1000]$ range. The exact position within this range is determined based on the confidence score assigned by the classifier; the higher the confidence in the negative class the closer the score to 0, the higher the confidence in the positive class the closer the score to 1000.

4.3 3-step approach

This approach consists of two steps: first, we try to separate relevant and central documents from the garbage and neutral ones (GN vs. RC). Second, we attempt to further distinguish between relevant and central (R vs. C). See Figure 2(b) for an illustration. Both steps are approached as a binary classification task; we use the same set of features for both steps, but the model is trained differently. For the first step, G and N are negative examples while R and C are positive examples. For the second step, R becomes the negative class and C alone remains the positive class.

Final document scores are also determined in two steps. Documents that are classified as negative in the first step are mapped to the $(0, 500]$ range, inversely proportional to the classifier's confidence (that is, the more confidence in the negative class the closer the score to 0). In the second step, documents classified as negative are mapped to $(500, 750]$ and documents classified as positive are mapped to $(750, 1000]$, based on the classifier's confidence values.

5. FEATURES

This section presents the features that we develop for the CCR task. Unlike in other filtering problems, the core issue here is not topicality; this requires features beyond the term space. We consider four types of features: (1) *document features* (Section 5.1) estimate the "citation worthiness" of the document on its own account, irrespective of the target entity; (2) *entity features* (Section 5.2) are based solely on the target entity; (3) *document-entity features* (Section 5.3) capture the relation between a particular document and the target entity as well as between the document and other entities related to the target; (4) *temporal features* (Section 5.4) aim to determine points in time where important events for the tar-

[10] http://wiki.dbpedia.org/Downloads37

get entity happen, based on the volume of entity mentions in the stream as well as based on Wikipedia usage data. Table 2 lists the features used. Some of these feature functions are instantiated with different parameter values; for each document-entity pair, we compute 68 feature values in total.

5.1 Document features

We use surface level features that are based solely on the characteristics of the document (D) and are independent of the target entity: the length of various document fields (D_f), such as body, title, and anchor text ($\mathrm{LEN}(D_f)$), and the source type ($\mathrm{SRC}(D)$). Further, we perform language detection to determine whether the document is written in English ($\mathrm{LANG}(D)$).

Admittedly, these are simple ideas that are not expected to deliver a solid performance on their own. Nevertheless, they are meant to represent an important group of features; one could consider more advanced document attributes here for inclusion, for example, related to credibility [43] or readability [36].

5.2 Entity features

We consider the number of entities that are known to be related to the target entity (i.e., already recorded as related in the knowledge base), $\mathrm{REL}(E)$. For convenience, we collected related entities from DBpedia (note that this could have been obtained directly from Wikipedia as well). We took all triples in which the target stood as the subject and considered all object entities as related.

While listed under the temporal block in Table 2, there are two additional features that may be considered here, as entity features: $\mathrm{WPV}(E)$ and $\mathrm{SV}(E)$. These attempt to measure the extent to which a target entity is likely to attract citations (cf. Section 5.4). As these functions have the target entity as their only parameter, they can also be seen as entity priors.

5.3 Document-entity features

The top seven feature functions in the third block of Table 2 are selected to characterise the occurrences of the target entity in the document: the number of occurrences in different document fields ($\mathrm{N}(D_f, E)$), the first and last positions in the document body ($\mathrm{FPOS}(D, E)$ and $\mathrm{LPOS}(D, E)$), and the "spread" of the entity's mentions across the document body ($\mathrm{SPR}(D, E)$). The latter three have a normalised variant too ($\mathrm{FPOS}_\mathrm{n}(D, E)$, $\mathrm{LPOS}_\mathrm{n}(D, E)$, and $\mathrm{SPR}_\mathrm{n}(D, E)$), where absolute values are divided by the length of the document, thus scaling it to $[0..1]$. We compute these feature values using both strict and loose name matching (18 values in total). For strict matches we use the entity name variants from DBpedia; for loose matched we use only the last names of persons. These correspond to the DBpedia and DBpedia-loose settings, respectively, in Section 4.1.

The next feature, $\mathrm{REL}(D_f, E)$ is about other entities, known to be related to the target, and counts their mentions in various document fields (body, title, and anchor text).

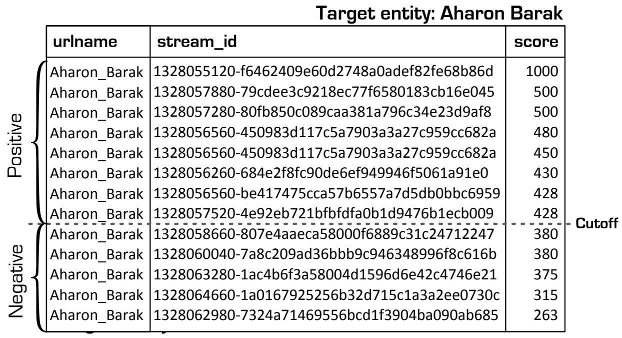
The last three features measure the textual similarity between the stream document and the target entity's article in the knowledge base, that is, the entity's Wikipedia page. We use Jaccard similarity, cosine similarity with TF-IDF term weighting, and the Kullback-Leibler divergence between language models built from the document and from the entity's Wikipedia page (using Dirichlet smoothing with the smoothing parameter μ set to the average document length in the collection).

5.4 Temporal features

Temporal features are meant to capture if something is happening around the target entity at a given point in time. We present

Table 2: Features grouped by type. Source can be stream (S), knowledge base (KB), or usage data (U). Value can be numerical (N), categorical (C), or boolean (B).

Feature		**Src.**	**Val.**
Document features			
$\mathrm{LEN}(D_f)$	Length (i.e., term count) of document field f	S	N
$\mathrm{SRC}(D)$	Document source (news, social, or linking)	S	C
$\mathrm{LANG}(D)$	Whether the document's language is English	S	B
Entity features			
$\mathrm{REL}(E)$	Number of related entities	KB	N
Document-entity features			
$\mathrm{N}(D_f, E)$	No. of occurrences of the target entity in document field f	S	N
$\mathrm{FPOS}(D, E)$	Term position of the first occurrence of E in document body	S	N
$\mathrm{FPOS}_\mathrm{n}(D, E)$	$\mathrm{FPOS}(D, E)$ normalised by the document length	S	N
$\mathrm{LPOS}(D, E)$	Term position of the last occurrence of E in document body	S	N
$\mathrm{LPOS}_\mathrm{n}(D, E)$	$\mathrm{LPOS}(D, E)$ normalised by the document length	S	N
$\mathrm{SPR}(D, E)$	Spread, i.e., distance between first and last occurrences	S	N
$\mathrm{SPR}_\mathrm{n}(D, E)$	$\mathrm{SPR}(D, E)$ normalised by the document length	S	N
$\mathrm{REL}(D_f, E)$	Number of different related entities mentioned in document field f	S	N
$\mathrm{SIM}_\mathrm{jac}(D, E)$	Jaccard similarity between the document and the entity's Wikipedia page	S,KB	N
$\mathrm{SIM}_\mathrm{cos}(D, E)$	Cosine similarity between the document and the entity's Wikipedia page	S,KB	N
$\mathrm{SIM}_\mathrm{kl}(D, E)$	KL-divergence between the document and the entity's Wikipedia page	S,KB	N
Temporal features			
$\mathrm{WPV}(E)$	Average hourly Wikipedia page views (over the training period)	U	N
$\mathrm{WPV}(E, h)$	Wikipedia page views volume in the past h hours	U	N
$\Delta\mathrm{WPV}(E, h)$	Change in Wikipedia page views volume in the past h hours	U	N
$\mathrm{WPB}(E, h)$	Burst in Wikipedia page views in the past h hours	U	B
$\mathrm{SV}(E)$	Average hourly stream volume (over the training period)	S	N
$\mathrm{SV}(E, h)$	Stream volume in the past h hours	S	N
$\Delta\mathrm{SV}(E, h)$	Change in stream volume in the past h hours	S	N
$\mathrm{SB}(E, h)$	Burst in stream volume in the past h hours	S	B



Target entity: Aharon Barak

urlname	stream_id	score
Aharon_Barak	1328055120-f6462409e60d2748a0adef82fe68b86d	1000
Aharon_Barak	1328057880-79cdee3c9218ec77f6580183cb16e045	500
Aharon_Barak	1328057280-80fb850c089caa381a796c34e23d9af8	500
Aharon_Barak	1328056560-450983d117c5a7903a3a27c959cc682a	480
Aharon_Barak	1328056560-450983d117c5a7903a3a27c959cc682a	450
Aharon_Barak	1328056260-684e2f8fc90de6ef949946f5061a91e0	430
Aharon_Barak	1328056560-be417475cca57b6557a7d5db0bbc6959	428
Aharon_Barak	1328057520-4e92eb721bfbfdfa0b1d9476b1ecb009	428
Aharon_Barak	1328058660-807e4aaeca58000f6889c31c24712247	380
Aharon_Barak	1328060040-7a8c209ad36bbb9c946348996f8c616b	380
Aharon_Barak	1328063280-1ac4b6f3a58004d1596d6e42c4746e21	375
Aharon_Barak	1328064660-1a0167925256b32d715c1a3a2ee0730c	315
Aharon_Barak	1328062980-7324a71469556bcd1f3904ba090ab685	263

Figure 3: Illustration of a possible list of scores generated for a given entity at an given time slot. The cutoff value here is 400**.**

features based on two sources. First, we use Wikipedia page view statistics, as a form of "social" signal; this data is publicly available and is organised into hourly batches.[11] We use the average hourly page views ($\mathrm{WPV}(E)$) as a general measure of the popularity of the entity. Further, we use the page views volume in the past h hours, both as an absolute value ($\mathrm{WPV}(E, h)$) and relative to the normal volume, observed up until h hours before the document's appearance ($\Delta\mathrm{WPV}(E, h)$); if the increase is more than twice the normal volume, we consider it to be a burst ($\mathrm{WPB}(E, h)$).

Second, we use the volume of documents in the stream that mention the target entity (using name variants from DBpedia for entity detection). As with the Wikipedia page views, we compute absolute and relative volumes ($\mathrm{SV}(E)$, $\mathrm{SV}(E, h)$, and $\Delta\mathrm{SV}(E, h)$), and detect bursts ($\mathrm{SB}(E, h)$).

We took great care to ensure that all the data taken into account was generated before the creation of the particular stream document that is being looked at (specifically, before the hour in which the document appears in the stream). All timestamps were normalised to Zulu time. For the time intervals we considered values $h = \{1, 2, 3, 6, 12, 24\}$; that amounts to 38 temporal features in total.

6. EXPERIMENTAL EVALUATION

This section presents the experimental evaluation of our multi-step classification approaches on the CCR task.

6.1 Evaluation methodology

We follow the evaluation methodology of the TREC 2012 KBA track. Recall that for each target entity, documents (that mention the entity) are assigned a confidence score in the range of $(0, 1000]$ with respect to how likely it is for a human to cite that document. Scoring is done by sweeping a confidence *cutoff* from 0 to 1000 in steps of 50; documents with a score above this threshold are treated as positive instances (i.e., identified as relevant or "citation worthy"). To illustrate, Figure 3 shows how the cutoff value is used in deciding the set of relevant (or positive) documents in a stream, for a given target entity. The scoring tool then computes precision, recall, and F-score (F1) for each entity and for each cutoff value, with respect to the assessors' judgements (see Section 3). We also report on scale utility (SU), a metric from general information filtering that is used to evaluate the ability for a system to accept relevant and reject non-relevant documents from a document stream [37].

Table 3: Results of entity mention identification: total number of document-entity pairs found (#D-E), recall (R), and the ratio of false positives (FP).

Entity identification	Training period			Testing period		
	#D-E	R	FP	#D-E	R	FP
urlname	23,160	.862	.540	41,248	.842	.559
DBpedia	37,905	.974	.682	70,411	.974	.701
DBpedia-loose	5,550,788	.991	.998	12,514,137	.994	.998

6.2 Identifying entity mentions

We start our evaluation with the first component of our classification pipeline: identifying entity mentions. This step is common to both the 2-step and 3-step approaches and is responsible for selecting documents for downstream processing (see Figure 2). Our objectives for this filtering component are to maintain high recall and, at the same, to keep false positives at a low rate.

Table 3 presents the results for the three different entity identification methods introduced in Section 4.1. We find that using the `urlname` alone for detecting entity mentions performs surprisingly well, achieving a recall of 86.2% and 84.2% on the training and testing periods, respectively. Adding known name variants from DBpedia pushes recall to 97.4% (on both splits); this also brings along an over 25% increase in false positive rate. Considering loose matches (i.e., only the last names of persons) results in nearly perfect recall; however, 99.8% of these matches do not refer to the target entity. We conclude that using DBpedia variants with strict matches, i.e., the middle row in Table 3, provides a balanced setting. It provides sufficiently high recall, without having to work with orders of magnitude more data in the subsequent classification step(s) than it is necessary. In the remainder of the paper we will be using the DBpedia identification method.

6.3 2-step vs. 3-step classification

Before we present results on the end-to-end CCR task, we briefly discuss the different experimental settings we use. Following the official TREC KBA evaluation, we consider two relevance levels: (i) only central documents are accepted as positive (denoted as **C**) and (ii) both relevant and central documents are treated as positive (denoted as **R+C**). Further, we present two alternative ways of determining confidence cutoffs: (i) using a single cutoff value that maximises F1/SU across all entities (this reflects the idea that a single value would need to be set for the whole system) and (ii) setting the cutoff values on a per-entity basis so that F1/SU is maximised for each individual entity (this is an "oracle" setting to show the full potential of a given method). We compute macro-averaged scores in all cases. We use the features introduced in Section 5 and employ two decision tree classifiers: J48 and Random Forest (RF). Implementations are based on the Weka machine learning toolkit [44] and use default parameter settings.[12] Note that it is not our interest to squeeze out every bit of performance by tweaking the classifiers' parameters. Instead, our goal is to compare the 2-step and 3-step approaches, where we treat classifiers as black boxes. Table 4 displays the results.

Comparing the two classifiers (rows 1 vs. 2 and 3 vs. 4), we find that RF outperforms J48 in all but one case (single cutoff, **C**, F1). The difference between the two, however, is subtle (below 5%), with the exception of SU scores in the central (**C**) setting (for both types of cutoff computations), where it is above 15%.

[11] http://dumps.wikimedia.org/other/pagecounts-raw/

[12] We also experimented with Naive Bayes and SVM, but the performance of those were far below that of decision trees.

Table 4: CCR results using (i) a single cutoff value for all entities (columns 2–5) and (ii) using the best cutoff value for each entity (columns 6–10). Best scores are typeset boldface.

Method	Single cutoff				Per-entity cutoff			
	C		R+C		C		R+C	
	F1	SU	F1	SU	F1	SU	F1	SU
2-step J48	**.360**	.263	.649	.630	.394	.292	.708	.710
2-step RF	.352	.342	.668	.657	.412	.427	**.715**	**.736**
3-step J48	.335	.300	.685	**.673**	.379	.328	.703	.697
3-step RF	.351	.347	**.691**	**.673**	.395	.423	.710	.721
HLTCOE [24]	.359	**.402**	.492	.555	**.416**	**.481**	.508	.576
UDel [26]	.355	.331	.597	.591	.365	.419	.597	.613

When it comes to the 2-step vs. 3-step approaches (rows 1–2 vs. 3–4), the winner is not immediately apparent. For the more stable classifier, RF, all score differences are below 5%. With J48 the results are mixed; for example, for **C**, the 2-step approach has higher F1 scores, but the 3-step approach performs better in terms of SU. Both the 2-step and 3-step approaches gain approximately equal benefits when moving from single to per-entity cutoff (columns 2–5 vs. 6–9), but the improvement is more substantial for **C** than for **R+C**. Overall, our preferred choice is the 2-step approach (specifically, with the RF classifier) as it performs at the same level while being considerably simpler than the 3-step strategy.

For reference, we also included the two best performing official runs from TREC 2012. Our F1 scores for **C** are at same level as those of the TREC best; SU scores are between those of HLTCOE and UDel. When relevant documents are also accepted (**R+C**), our methods perform noticeably better than the best TREC approaches.

7. FEATURE ANALYSIS

In this section we perform an analysis of our features with the help of an information gain-based feature selection algorithm [46]. Table 5 reports the highest and lowest ranked features for three different classification settings: (1) non-relevant (G and N) vs. relevant (R and C), (2) non-relevant (G and N) vs. central (C), and (3) relevant (R) vs. central (C).

The first question we wish to answer is the following: Which (types of) features work best? Entity features perform very well in all settings; here, we mean not only the number of related entities ($\mathrm{REL}(E)$), but also stream volume ($\mathrm{SV}(E)$) and Wikipedia page views ($\mathrm{WPV}(E)$). It is somewhat surprising that the strongest features all work as a kind of prior and do not consider document content at all. The next best type of features is entity-document. Specifically, variants on the similarity between the document's content and the entity's Wikipedia page ($\mathrm{SIM}(D, E)$) and the spread and number of entity mentions in the document's body ($\mathrm{SPR}(D, E)$, $\mathrm{N}(D_f, E)$) are found to be particularly useful. There is no temporal feature (apart from the general volume related ones discussed already) that would stand out as universally beneficial. The lowest ranked features are entity and related entity mentions in the title and anchor fields. This is not unreasonable, given that these fields exist only for social documents (cf. Section 3).

Our second question is concerned with the differences between the three classification settings in Table 5. Do we find the same features to work best everywhere? In general, this seems to be the case; the really effective features (entity priors, entity-document similarity, spread, etc.) are among the top ones for all three settings. We find fewer volume-change-related features among the top ones (none in the top 10 actually) in R vs. C than for GN vs. RC or GN vs. C. We also find that document language ($\mathrm{LANG}(D)$) is the second-worst feature for R vs. C, while it is moderately useful in the other two settings.

8. CONCLUSIONS

In this paper we have addressed the cumulative citation recommendation (CCR) task for knowledge base acceleration (KBA). This task aims at identifying *central* documents from a content stream that would imply modifications to the attributes or relations of the given target entity in a given knowledge base (Wikipedia). We have introduced two multi-step classification approaches for this task that consist of two and three binary classification steps, respectively; hence, they have been termed 2-step and 3-step approaches. Both methods share the first component, which is concerned with the identification of entity mentions in documents based on various (known) surface forms of the entity. Subsequent steps use a total of 68 features that fall into four main categories: document, entity, document-entity, and temporal. We have performed a thorough experimental evaluation and comparison of our approaches using two decision tree classifiers (J48 and Random Forest), two relevance levels, and two alternative ways of determining confidence cutoffs. Both approaches performed very similarly, which makes 2-step approach the preferred choice given its relative simplicity. Further, we have shown that our approaches achieve very competitive performance compared to systems participating in the TREC 2012 KBA track and that they represent the current state-of-the-art. Finally, our feature analysis revealed that the most useful features are the ones related to the entity's connectedness and popularity, followed by features that capture the similarity between the document and the entity's Wikipedia article.

While relevant documents can be identified very effectively with our current features, separating between relevant and central documents remains to be challenging. In future work we are planning to extend our set of features with additional ones that are better in capturing this fine distinction.

9. ACKNOWLEDGMENTS

We thank John R. Frank for being extremely helpful and responsive with all our TREC KBA related questions and for providing valuable final feedback on the paper.

References

[1] D. Ahn, V. Jijkoun, G. Mishne, K. Müller, M. de Rijke, S. Schlobach, M. Voorhees, and L. Buckland. Using Wikipedia at the TREC QA track. In *TREC '04*, 2005.

[2] J. Allan. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, 2002.

[3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR '98*, pages 37–45, 1998.

[4] S. Araujo, G. Gebremeskel, J. He, C. Bosscarino, and A. de Vries. CWI at TREC 2012, KBA track and session track. In *TREC'12*, 2013.

[5] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2011 entity track. In *TREC '11*, 2012.

[6] R. Berendsen, E. Meij, D. Odijk, M. de Rijke, and W. Weerkamp. The University of Amsterdam at TREC 2012. In *TREC'12*, 2013.

[7] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[8] L. Bonnefoy, V. Bouvier, and P. Bellot. LSIS/LIA at TREC 2012 knowledge base acceleration. In *TREC'12*, 2013.

[9] R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL '06*, pages 9–16, 2006.

Table 5: Best and worst features based on Information Gain.

GN vs. RC			GN vs. C			R vs. C		
Feature	**Params**	**IG**	**Feature**	**Params**	**IG**	**Feature**	**Params**	**IG**
$\mathrm{REL}(E)$		0.133	$\mathrm{SIM}_{\mathrm{cos}}(D,E)$		0.246	$\mathrm{SV}(E)$		0.117
$\mathrm{WPV}(E)$		0.132	$\mathrm{REL}(E)$		0.209	$\mathrm{REL}(E)$		0.116
$\mathrm{SIM}_{\mathrm{cos}}(D,E)$		0.119	$\mathrm{WPV}(E)$		0.206	$\mathrm{SPR}_{\mathrm{n}}(D,E)$	loose	0.113
$\mathrm{SV}(E)$		0.105	$\mathrm{SPR}_{\mathrm{n}}(D,E)$	loose	0.200	$\mathrm{WPV}(E)$		0.105
$\mathrm{SPR}_{\mathrm{n}}(D,E)$	loose	0.087	$\mathrm{SIM}_{\mathrm{jac}}(D,E)$		0.180	$\mathrm{SIM}_{\mathrm{cos}}(D,E)$		0.098
$\mathrm{N}(D_f,E)$	loose, body	0.081	$\mathrm{SV}(E)$		0.158	$\mathrm{SIM}_{\mathrm{jac}}(D,E)$		0.090
$\mathrm{SPR}(D,E)$	loose	0.080	$\mathrm{SPR}(D,E)$	loose	0.153	$\mathrm{FPOS}(D,E)$	loose	0.087
$\mathrm{SIM}_{\mathrm{jac}}(D,E)$		0.076	$\mathrm{N}(D_f,E)$	loose, body	0.152	$\mathrm{SIM}_{\mathrm{kl}}(D,E)$		0.082
$\Delta\mathrm{SV}(E,h)$	$h=2$	0.067	$\Delta\mathrm{SV}(E,h)$	$h=3$	0.140	$\mathrm{N}(D_f,E)$	loose, body	0.081
$\Delta\mathrm{WPV}(E,h)$	$h=2$	0.066	$\mathrm{FPOS}(D,E)$	loose	0.134	$\mathrm{FPOS}(D,E)$		0.081
...			...			...		
$\mathrm{WPB}(E,h)$	$h=2$	0.003	$\mathrm{N}(D_f,E)$	title	0.004	$\mathrm{WPB}(E,h)$	$h=2$	0.002
$\mathrm{REL}(D_f,E)$	loose, anchor	0.001	$\mathrm{REL}(D_f,E)$	anchor	0	$\mathrm{WPB}(E,h)$	$h=3$	0.001
$\mathrm{N}(D_f,E)$	anchor	0.001	$\mathrm{N}(D_f,E)$	loose, anchor	0	$\mathrm{WPB}(E,h)$	$h=1$	0.001
$\mathrm{REL}(D_f,E)$	anchor	0.001	$\mathrm{REL}(D_f,E)$	title	0	$\mathrm{LANG}(D)$		0.001
$\mathrm{N}(D_f,E)$	title	0	$\mathrm{N}(D_f,E)$	anchor	0	$\mathrm{REL}(D_f,E)$	anchor	0

[10] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI '10*, pages 1306–1313, 2010.

[11] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP-CoNLL '07*, pages 708–716, 2007.

[12] J. Dalton and L. Dietz. Bi-directional linkability from Wikipedia to documents and back again: UMass at TREC 2012 knowledge base acceleration track. In *TREC'12*, 2013.

[13] G. Demartini, T. Iofciu, and A. P. De Vries. Overview of the INEX 2009 entity ranking track. In *INEX '09*, 2010.

[14] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *COLING '10*, pages 277–285, 2010.

[15] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, 2008.

[16] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff. Building an entity-centric stream filtering test collection for TREC 2012. In *TREC'12*, 2013.

[17] O. Gross, A. Doucet, and H. Toivonen. Term association analysis for named entity filtering. In *TREC'12*, 2013.

[18] U. Hanani, B. Shapira, and P. Shoval. Information filtering: Overview of issues, research and systems. *User Model. User-Adapt. Interact.*, 11(3):203–259, 2001.

[19] J. Hoffart, F. Suchanek, K. Berberich, and G. Weikum. YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[20] D. W. Huang, Y. Xu, A. Trotman, and S. Geva. Overview of INEX 2007 link the wiki track. In *INEX '07*, 2008.

[21] B. Huurnink, L. Hollink, W. van den Heuvel, and M. de Rijke. Search behavior of media professionals at an audiovisual archive: A transaction log analysis. *J. Am. Soc. Inf. Sci. Technol.*, 61(6):1180–1197, 2010.

[22] H. Ji and R. Grishman. Knowledge base population: successful approaches and challenges. In *ACL HLT '11*, pages 1148–1158, 2011.

[23] J. Kazama and K. Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *EMNLP-CoNLL '07*, pages 698–707, 2007.

[24] B. Kjersten and P. McNamee. The HLTCOE approach to the TREC 2012 KBA track. In *TREC'12*, 2013.

[25] Y. Li, Z. Wang, B. Yu, Y. Zhang, R. Luo, W. Xu, G. Chen, and J. Guo. PRIS at TREC2012 KBA track. In *TREC'12*, 2013.

[26] X. Liu and H. Fang. Entity profile based approach in automatic knowledge finding. In *TREC'12*, 2013.

[27] J. Mayfield, D. Lawrie, P. McNamee, and D. W. Oard. Building a cross-language entity linking collection in twenty-one languages. In *CLEF '11*, 2011.

[28] E. Meij and M. de Rijke. Supervised query modeling using Wikipedia. In *SIGIR '10*, pages 875–876, 2010.

[29] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Mapping queries to the linking open data cloud: A case study using DBpedia. *Web Semantics*, 9(4):418–433, 2011.

[30] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM '12*, pages 563–572, 2012.

[31] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07*, pages 233–242, 2007.

[32] D. N. Milne and I. H. Witten. Learning to link with Wikipedia. In *CIKM '08*, pages 509–518, 2008.

[33] D. Mladenic. Using text learning to help web browsing. In *SIGCHI '01*, pages 893–897, 2001.

[34] N. Nanas, A. Roeck, and M. Vavalis. What happened to content-based information filtering? In *ICTIR '09*, pages 249–256, 2009.

[35] N. Nanas, M. Vavalis, and A. N. D. Roeck. A network-based model for high-dimensional information filtering. In *SIGIR '10*, pages 202–209, 2010.

[36] T. Polajnar, R. Glassey, and L. Azzopardi. Detection of news feeds items appropriate for children. In *ECIR '12*, pages 63–72, 2012.

[37] S. E. Robertson and I. Soboroff. The TREC 2002 filtering track report. In *TREC '02*, 2003.

[38] W. Shen, J. Wang, P. Luo, and M. Wang. LINDEN: linking named entities with knowledge base via semantic knowledge. In *WWW '12*, pages 449–458, 2012.

[39] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *SIGIR '10*, pages 841–842, 2010.

[40] F. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge. In *WWW '07*, pages 697–706, 2007.

[41] C. Tompkins, Z. Witter, and S. G. Small. SAWUS Siena's automatic Wikipedia update system. In *TREC'12*, 2013.

[42] Z. Wang, J. Li, Z. Wang, and J. Tang. Cross-lingual knowledge linking across wiki knowledge bases. In *WWW '12*, pages 459–468, 2012.

[43] W. Weerkamp and M. de Rijke. Credibility improves topical blog post retrieval. In *ACL '08*, pages 923–931, 2008.

[44] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., 2005.

[45] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on Wikipedia. In *SIGIR '09*, pages 59–66, 2009.

[46] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97*, pages 412–420, 1997.