Cumulative Citation Recommendation: Classification vs. Ranking

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

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

ABSTRACT

Cumulative citation recommendation refers to the task of filtering a time-ordered corpus for documents that are highly relevant to a predefined set of entities. This task has been introduced at the TREC Knowledge Base Acceleration track in 2012, where two main families of approaches emerged: classification and ranking. In this paper we perform an experimental comparison of these two strategies using supervised learning with a rich feature set. Our main finding is that ranking outperforms classification on all evaluation settings and metrics. Our analysis also reveals that a ranking-based approach has more potential for future improvements.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software

Keywords

Knowledge base acceleration, cumulative citation recommendation, information filtering

1. INTRODUCTION

Knowledge bases, such as Wikipedia, are increasingly being utilised in various information access contexts. With the exponential growth of the amount of information being produced, a continuously increasing effort is demanded from editors and content managers who are responsible for the maintenance and update of these knowledge bases. To partly address this challenge, the Text REtrieval Conference (TREC) has launched a Knowledge Base Acceleration (KBA) track in 2012 with the ultimate goal to develop systems that can aid humans expand knowledge bases by automatically recommending edits based on incoming content streams [10]. In its first year, the track focused on a single problem and introduced the *cumulative citation recommendation* (CCR) task: given a textual stream consisting of news and social media content and a target entity from a knowledge base (Wikipedia), generate a score for each document based on how pertinent it is to the input entity.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

At TREC, two main families of approaches emerged: classification and ranking. While ranking methods were more popular, no empirical evidence has been provided yet to support that ranking is to be preferred over classification for CCR. Based on participants' system descriptions, we found a single case where the choice was made deliberately after consideration. Efron et al. [8] argue that ranking provides a relevance score, as opposed to a binary decision, which "would allow a Wikipedia editor to browse documents in decreasing order of predicted relevance, while also admitting a simple thresholding if she wished to see only those documents judged to be either 'relevant,' 'central' or both." This, however, has to do with the consumption of the results. A classifier's confidence values can easily be turned into relevance scores [3–5].

Our goal with this study is to compare classification and ranking approaches for CCR. To ensure a fair assessment, we employ supervised learning for both groups of methods and utilise a rich feature set we developed in prior work [3]. For classification, we use various multi-step models from existing work that have been shown to deliver state-of-the-art performance [3]. For ranking we test pointwise, pairwise, and listwise learning-to-rank approaches. We find a pointwise ranking approach, Random Forests, to perform best and that it improves substantially on best known results so far. Our analysis also reveals that this approach has more potential for future improvements than its classification-based counterpart.

2. RELATED WORK

Knowledge base population (KBP) refers to efforts made to expand knowledge bases by reading arbitrary text provided, e.g., as Web documents, and extracting meaningful information from it [6, 9]. The Text Analysis Conference (TAC) introduced a dedicated Knowledge Base Population track in 2009 [13]. A core component of KBP that bears immediate relevance to our task is *entity linking*: given an entity and a document containing a mention of the entity, identify and link the corresponding node in the knowledge base [17, 18]. The CCR task naturally has an entity identification element to it; entity disambiguation, however, is tackled implicitly, as part of centrality detection [3, 10].

The CCR task also shares similarities with both *information filtering* [12] and *topic detection and tracking* (TDT) [1]. One key difference lies in the end user task that is being modelled. Another key difference is that CCR has no novelty requirement. Finally, CCR attempts to make fine-grained distinctions between relevant and central documents.

3. TASK AND DATA DESCRIPTION

Recognising the need for intelligent systems that can help reduce efforts associated with the maintenance of large-scale knowledge bases, TREC has launched a Knowledge Base Acceleration (KBA) track in 2012. The track has introduced the *cumulative citation*

recommendation (CCR) task: filter a time-ordered corpus for documents that are highly relevant to a predefined set of entities [10].

Data collection. A document collection, called KBA Stream Corpus 2012[1], has been developed specifically for this track. It covers the time period from October 2011 to April 2012, and is composed of three types of documents: (1) *news*, i.e., global public news wires; (2) *social*, i.e., blogs and forums; and (3) *linking*, i.e., content from URLs shortened at `bitly.com`. The raw collection amounts to 8.7TB. It also comes in a "cleansed" version, 1.2TB (270GB compressed), where body text is extracted after HTML boilerplate removal. For *social* documents the content is further separated into title, body, and anchor fields (for *news* and *linking* only the body field is available). We work with the cleansed version, but we do not make use of the provided named entity annotations. Each *stream document* (or *document* for short) is timestamped and is uniquely identified by a `stream_id`.

Topics. The topic set consists of 29 entities (27 persons and 2 organisations), referred to as *target entities* and are uniquely identified by a `urlname`. These are described by semi-structured articles in a knowledge base, specifically, Wikipedia. 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.

Training and testing data. TREC KBA provides training annotation data, i.e., assessor judgements, 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.

Annotations are provided along two dimensions: contain 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 (G)*: not relevant (e.g., spam); *neutral (N)*: not relevant (nothing can be learned about the target entity); *relevant (R)*: relates indirectly to the target entity (e.g., mentions topics or events that are likely to have an impact on the entity); *central (C)*: relates directly to the target entity (e.g., the entity is a central figure in the mentioned topics or events). 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.

Note that, in theory, a document can be relevant, even if it does not mention the target directly. In practice, however, centrality never happens without an explicit mention of the entity in the document [10]. Therefore, we are focusing only on documents with explicit mentions, i.e., the top row in Figure 1.

Evaluation Methodology. Systems performing the CCR task are required to process the documents in chronological order (in hourly batches) and assign a score in the range of $(0, 1000]$ for each document that is deemed citation-worthy for a given target entity. Evaluation against the ground truth (i.e., manual annotations) is performed as follows. A *cutoff* value is varied from 0 to 1000 (in steps of 50) and documents with a score above the cutoff are considered being identified as relevant (positive cases) by the system. Consequently, documents below the cutoff are treated as irrelevant (belong to the negative class). Precision, recall, and F-score (F1) are computed as a function of the relevance cutoff. In addition, as with general information filtering, the notion of *Scale Utility (SU)* is used to evaluate the ability for a system to accept relevant and reject non-relevant documents from a document stream [19].

Following the official TREC KBA evaluation methodology, we consider two experimental settings: (i) treating only central documents as positives and non-centrals as negatives (denoted as **C**) and (ii) accepting both relevant and central documents as correct (denoted as **R+C**). Further, as suggested in [3], we present two alternative ways of determining confidence cutoffs: (i) using a single cutoff value that maximises F1/SU across all entities and (ii) setting the cutoff values on a per-entity basis so that F1/SU is maximised for each individual entity. All scores are macro averages.

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

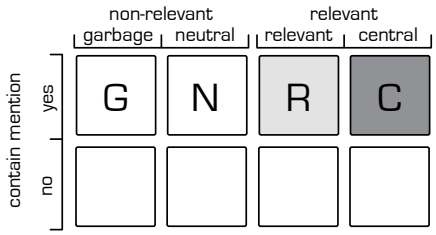

Figure 1: Document annotation matrix from the TREC 2012 KBA track. The goal of the CCR task is to identify central documents, i.e., the ones in the top right corner.

4. APPROACHES

In this section we discuss two main approaches to solving the CCR task: classification and ranking. Note that at this point our main focus in on the overall strategy; we are unconcerned about the particular choice of classification/ranking algorithm and of the features used. For both classification and ranking we employ supervised learning and use the same feature set (presented in Section 5).

Identifying entity mentions. Common to both approaches is an initial step that provides a filter to identify whether or not a document contains a mention of the target entity. We base this detection on strict string matching, using known name variants of the entity extracted from DBpedia, an approach that is shown to achieve high recall while keeping a low false positive rate [3]. Note that central documents always contain an explicit mention of the entity (cf. Section 3). Therefore, we expect this initial filtering to have a precision-enhancing effect, potentially, at the expense of recall. This filtering step also allows us to avoid having to compute a potentially large set of features for all possible document-entity pairs.

4.1 Classification

In [3] two multi-step classification approaches are presented. For both, the first step is concerned with the identification of entity mentions in documents (which may be viewed as a binary classification task). It is followed by one or two subsequent binary classification steps (making it two or three in total, respectively).

Under the *2-step approach* documents are classified as central or not in a single step. Document-entity pairs labeled as garbage (G) or neutral (N) are used as negative examples and central (C) ones are used as positive examples. Note that instances labeled as relevant (R) are ignored (so as not to "soften the distinction between the two classes that we are trying to separate" [3]). Negative predictions are mapped to the $(0, 500]$ range and the positive (central) predictions to the $(500, 1000]$ range, using the classifier's confidence estimations.

The *3-step approach* first attempts to separate garbage or neutral documents from the relevant or central ones (GN vs. RC). A second classifier is applied to documents that fall into the latter category to distinguish them further into relevant and central classes (R vs. C). The same set of features is used for both steps, but the model is trained on the respective classes. 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. In the second step, documents classified as negative are mapped

to the $(500, 750]$ range, while documents classified as positive are mapped to the $(750, 1000]$ interval. As before, the actual score is determined based on the classifier's confidence estimates.

TREC 2012 KBA track. In [14] CCR is approached as a classical text classification task; a linear SVM model in trained using unigram and entity name features. Both [4] and [5] apply a Random Forest classifier on top of documents that have been filtered to contain mention of the target entity. In [4] a single classification step is used; this corresponds to our 2-step approach. In [5] documents are first classified into G+N vs. R+C classes, then, documents assigned to the latter class are further separated into R and C, using a second classifier; this is equivalent to our 3-step approach.

4.2 Ranking

We can also approach CCR as a learning-to-rank (LTR) problem: estimate a numerical score for document-entity pairs. The target score directly corresponds to the target class: 0 for garbage and neutral, 1 for relevant, and 2 for central documents. When training the models, we can directly target the different evaluation scenarios. In the "Central" setting we exclude documents being labeled relevant from the training set. For the second setting, "Relevant + Central," we include relevant documents too, for training. Unlike with classification, we do not need a different approach for the two cases, this can be left to the learning algorithm. Also, mapping the estimated target scores to $(0, 1000]$ is straightforward.

In principle, any LTR algorithm may be used. However, since our target scores can only take a handful of possible values, we would expect pointwise and pairwise methods to work best. Listwise algorithms might have difficulties as target scores are not continuous functions w.r.t. to the ranking model's parameters.

TREC 2012 KBA track. A variety of scoring methods has been tried, including language models [2], Markov Random Fields [7], standard Lucene scoring [20], Jaccard similarity [15], and custom ranking functions based on entity co-occurrences [11, 16]. Using merely the name of the target entity as a query is a very sparse representation; therefore, query expansion is often employed to enrich the "entity profile" query with other name variants and contextual information (terms and/or related entities) from Wikipedia [7, 8, 15, 16, 20] or from the document stream [7, 8, 15].

5. FEATURES

The choice of features has a direct impact on the accuracy of machine learning algorithms. A variety of features has been proposed at TREC 2012 KBA with the aim of going beyond the term space, capturing phenomena specific to the CCR task. We wish to emphasise that we recognise the importance of feature engineering for this task and it is worthy of a study on its own account. In this paper, however, our focus is on the comparison of classification vs. ranking approaches and not on the features themselves. Therefore, we use the features proposed in [3], a total of 68, and resort to a high-level overview of them. We would also like to note that the features are only computed for document-entity pairs where the entity is mentioned in the document.

Document features. Surface level features that are based solely on the characteristics of the document and are independent of the target entity: the length of various document fields, such as body, title, and anchor text; the source type (news, social, or linking); whether the document is in English.

Entity features. This group consists of a single feature: the number of entities that are known to be related to the target entity (i.e., already recorded as related in the knowledge base).

Document-entity features. One group of features characterises the occurrences of the target entity in the document: the number of occurrences in different document fields; first and last positions in the document body; the "spread" of the entity's mentions across the document body. These are computed using both strict and loose name matching. The second subset of features focuses on other entities that are known to be related to the target: counts of related entity mentions in various document fields (body, title, and anchor text). The last batch of features measures the textual similarity between the stream document and the target entity's article in the knowledge base (that is, the entity's Wikipedia page): Jaccard similarity, cosine similarity with TF-IDF term weighting, and the Kullback-Leibler divergence.

Temporal features. Two sources are used to capture if something is "happening" around the target entity at a given point in time. First, based on Wikipedia page view statistics: average hourly page views; page views volume in the past h hours, both as an absolute value and relative to the normal volume; whether there is a burst in the past h hours (where h is $1, 2, 3, 6, 12, 24$). Second, the same set of features, but based on the volume of documents in the stream that mention the target entity: average hourly page views; absolute and relative volumes; whether there is a burst detected.

Table 1: 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
Classification								
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
Ranking								
Random Forests	**.390**	.369	**.722**	**.718**	**.463**	.480	**.776**	**.790**
RankBoost	.339	.356	.697	.691	.405	.452	.745	.766
LambdaMART	.354	.351	.646	.624	.410	.463	.673	.701
TREC bests								
HLTCOE [14]	.359	**.402**	.492	.555	.416	**.481**	.508	.576
UDel [16]	.355	.331	.597	.591	.365	.419	.597	.613

6. EXPERIMENTAL EVALUATION

For classification we employ two decision tree classifiers, as in [3]: J48 and Random Forest (RF).[2] We use the implementations of the Weka machine learning toolkit[3] with default parameter settings. For ranking we experimented with different learning-to-rank methods that are currently available in RankLib.[4] Because of space limitations, we only report on the top performing method from each class: Random Forests for pointwise, RankBoost for pairwise, and LambdaMART for listwise. Table 1 displays the results. Runs and detailed evaluation results are made available at http://bit.ly/16RraPB.

We find that all classification methods deliver similar performance. With a single exception, RF always outperforms J48. Both

[2]We also experimented with SVM and Naive Bayes in prior work, but the performance of those were far below that of decision trees.
[3]http://www.cs.waikato.ac.nz/~ml/weka/
[4]http://people.cs.umass.edu/~vdang/ranklib.html

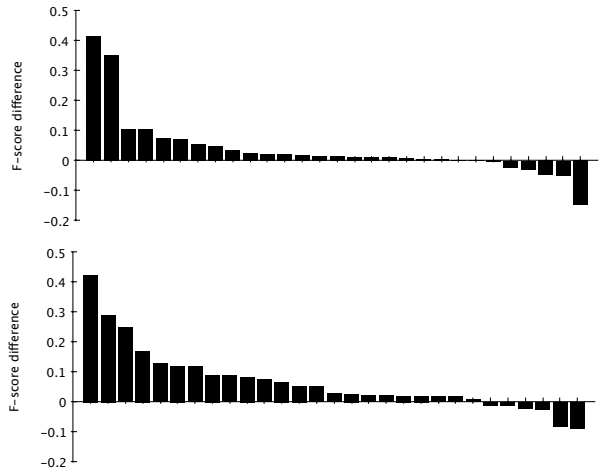

Figure 2: Random Forests ranking vs. Random Forest 3-step classification. (Top) best overall cutoff value; (Bottom) best cutoff value per topic.

the 2-step and 3-step approaches gain approximately equal benefits when moving from single to per-entity cutoffs.

The best ranking method, Random Forests, is a pointwise one; this is in line with our expectations. The pairwise method (RankBoost) outperforms the listwise approach (LambdaMART) when both central and relevant are accepted; for central-only there is no clear winner. We observe the same trends when moving from single to per-entity cutoffs as with classification approaches. In overall, we find that the best ranking method outperforms all classification approaches, while the other two deliver competitive performance.

For reference, we also report results on the two top performing official TREC submissions. The HLTCOE approach has very high SU scores for **C**; in the single cutoff case it cannot be matched. However, on all the other metrics and settings, the Random Forests ranking method outperforms the best TREC approaches and does so by a considerable margin (ranging from 8.6% up to 52.7%).

7. TOPIC-LEVEL ANALYSIS

Average results might hide interesting differences on the level of individual topics; we continue with a topic-level analysis in this section. To remain focused, we use a single representative for each family of approaches: Random Forest 3-step for classification and Random Forests for ranking. Further, we limit ourselves to the "Central" evaluation setting (that is, only central documents are accepted as relevant) and use F1 as our evaluation measure.

Since evaluation measures are computed as a function of the relevance cutoff, we consider two settings. First, we use the best overall cutoff value; this corresponds to column 2 in Table 1. In Figure 2 (Top) we plot topics in decreasing order of F1 score differences. Positive values mean that the ranking approach performs better, while negative values indicate the advantage of the classification approach on that particular topic. We find that ranking performs better on 14 topics, the difference is negligible on 13 (< 0.05), and classification wins only on 2.

Next, we perform the same comparison, but selecting the best cutoff value for each individual topic; this corresponds to column 6 in Table 1. Figure 2 (Bottom) displays the results. We can see that the ranking approach benefits a lot more from the cutoff optimisation, not just in terms of absolute score, but also on the level of individual topics. This suggests that the ranking approach holds more promise for additional improvements that might be achieved by optimising w.r.t. the cutoff parameter.

8. CONCLUSIONS

In this work, we have carried out a comparative study on two families of approaches to support the cumulative citation recommendation (CCR) task for knowledge base acceleration (KBA). Specifically, we have compared classification methods against ranking methods with respect to their ability to identify central documents from a content stream that would imply modifications to a given target entity in a knowledge base. Our results have shown that ranking approaches are a better fit for this task. Our conjecture is that this has to do with the particular evaluation methodology employed by the TREC KBA track; ranking methods directly emit results in the desired format, while classification methods need to resort to additional mechanisms that can translate their output to a ranking. Further gains might be achieved by optimising against the relevance cutoff parameter; we have found that a ranking-based approach has more room for improvements in this respect.

References

[1] J. Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer, 2002.

[2] 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.

[3] K. Balog, N. Takhirov, H. Ramampiaro, and K. Nørvåg. Multi-step classification approaches to cumulative citation recommendation. In *Proc. of OAIR'13*, pages 121–128, 2013.

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

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

[6] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proc. of AAAI'10*, 2010.

[7] 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.

[8] M. Efron, J. Deisner, P. Organisciak, G. Sherman, and A. Lucic. The University of IllinoisâĂŹ Graduate School of Library and Information Science at TREC 2012. In *TREC '12*, 2013.

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

[10] 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.

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

[12] U. Hanani, B. Shapira, and P. Shoval. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259, 2001.

[13] H. Ji and R. Grishman. Knowledge base population: successful approaches and challenges. In *Proc. of ACL HLT'11*, 2011.

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

[15] 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.

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

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

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

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

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