Cumulative Citation Recommendation: A Feature-aware Comparisons of Approaches

Gebrekirstos G. Gebremeskel[*]
CWI, Amsterdam, The Netherlands
gebre@cwi.nl

Jiyin He[†]
CWI, Amsterdam, The Netherlands
j.he@cwi.nl

Arjen P. de Vries[‡]
CWI, Amsterdam, The Netherlands
arjen@acm.org

ABSTRACT

In this work, we conduct a feature-aware comparison of approaches to Cumulative Citation Recommendation (CCR), a task that aims to filter and rank a stream of documents according to their relevance to entities in a knowledge base. We conducted experiments starting with a big feature set, identified a powerful subset and applied it to comparing classification and learning-to-rank algorithms. With few set of powerful features, we achieve better performance than the state-of-the-art. Surprisingly, our findings challenge the previously known preference of learning-to-rank over classification: in our study, the CCR performance of the classification approach outperforms that using learning-to-rank. This indicates that comparing two approaches is problematic due to the interplay between the approaches themselves and the feature sets one chooses to use.

1. INTRODUCTION

Knowledge Bases (KBs) such as Wikipedia have gained popularity and can be considered an important knowledge resource in our daily lives. KB curators need to constantly watch for new information and populate and maintain KBs so that they stay up-to-date, useful and accurate. However, the number of entities in a KB on one hand, and the huge amount of new information content on the Web on the other hand makes population and maintenance a challenging task. To address this, the Text REtrieval Conferences (TREC) introduced the Knowledge Base Acceleration (KBA) track in 2012[1]. TREC-KBA seeks to partially automate KB population and maintenance by recommending relevant documents to KB curators. TREC-KBA's main task, CCR, aims at filtering a stream to identify those documents that are citation-worthy to KB entities of interest.

A number of studies [1, 2, 3, 4] experimented with various types of features and approaches. These studies, while experimenting with a large number of features, never examined the power of individual features. Feeding many features into a classifier may, however, make the model unnecessarily complex, increase the chance of over-fitting and amplify the curse of dimensionality. Different approaches are, in the absence of common features, compared with each other to determine which one performs better. It is difficult to judge whether the observed performance difference is due to the approaches themselves or the (different) sets of feature used.

In this paper, we study the contributions to performance of individual features with the goal of selecting a few powerful ones. Keeping the set of fixed selected features , we compare the best performing approaches used in the literature. The contributions of the study are: (1) a fair comparison of feature effectiveness from several previous studies, (2) identifying a powerful subset of features leading to an effectiveness beyond that of the state-of-the-art CCR systems, and (3) demonstrating that with the reduced but more effective set of features, previous findings that certain approaches outperform others do not hold, suggesting that we can not compare approaches independently from the features used.

The rest of the paper is organized as follows: in sections 2, 3 and 4, we discuss data and problem description, related work, and methods used. In section 5, we discuss our experiments, followed by results and analysis in 6. Finally, in section 7, we state our conclusions.

2. DATA AND TASK DESCRIPTION

2.1 Data

We use the TREC KBA-CCR-2012 dataset[2]. It consists of 29 Wikipedia entities and a time-stamped stream of documents containing news, social media content, and content from bitly.com URLs.

TREC-KBA provided relevance judgments for training and testing. Documents with citation-worthy content to a given entity are annotated as *central*, and those with tangentially relevant content are annotated as *relevant*. Documents with no relevant content and spam are labeled *neutral* and *garbage*.

[*]Dr. Trovato insisted his name be first.
[†]
[‡]
[1]http://trec-kba.org/
[2]http://trec-kba.org/kba-ccr-2012.shtml

2.2 Task

Given a stream of documents of news items, blogs and social media on one hand and Wikipedia entities on the other, we conduct a feature study to identify a small set of effective features that are then used to compare different approaches employed in CCR.

3. RELATED WORK

Three different categories of approaches to solving the task of CCR have been proposed in previous work, categorized as string-matching, classification and learning to rank (LTR). With string-matching, entities are represented by a small set of key strings that capture entity occurrences, and documents that match the strings are retrieved as relevant [1, 2]. The best performing method uses an entity's name mention and mentions of related entities [2] as features. The method ranks documents by using a function that assigns a base score to a document that mentions the entity by name. Mentions of related entities increase the base score.

The best performing method from the second category compares two multi-step methods. An initial step filters the stream for potentially relevant documents. The 3-step approach uses a classifier to separate *garbage* and *neutral* from *relevant* and *central*, and a second classifier to separate *relevant* from *central*. The 2-step approach directly trains a classifier to separate *garbage* and *neutral* from *central*. Relevant annotations are excluded from the training stage in order not to introduce confusing examples. The 2-step approach achieves a better performance than that of the 3-step approach.

Related to [3], the authors have proposed to use LTR instead of classification [4]. The classification and learning-to-rank approaches of [3, 4] shared the same set of 68 distinct features. The authors conclude from their experiments that LTR approaches outperform classification approaches.

Our study is an independent reproduction of previously published findings, along with improvements. Specifically, we use the 2-step approach of [3, 4], reconsider the features proposed in [1, 2, 3, 4], and demonstrate empirically that a small subset is sufficient and leads to improved results. We demonstrated that with the reduced but more effective set of features, a classification-based approach outperforms a learning-to-rank-based approach. This finding deviates from results in previous study [4].

4. METHOD

We take the 68 features provided as accompanying data for [3, 4][3] as they are and add 5 others (adapted from [1, 2]), making a total of 73 initial features. The features consist of 5 document, 1 entity, 24 document-entity, 38 temporal, and 5 adapted or new features. Document and entity features are computed from processing the documents and entities respectively. Document-entity features are computed by aggregating scores over strings for which a match has been found in a document. For example, if we consider the Personalized Page Rank (PPR) feature, for each entity, there are 100 related entities each with their own PPR score. When processing a document entity pair, if a document matches strings from the entity's pre-constructed related entities, we aggregate the scores and the sum becomes the PPR score for that document-entity pair. Temporal features are meant to capture when important events related to the entities happen by measuring spikes in their respective Wikipedia views and the streaming documents.

The 2-step approach that we use consists of filtering followed by classification(as in [3]) or learning-to-rank(as in [4]). The first step filters the stream for documents that are potentially relevant using DBpedia name variants of the Wikipedia entities. The second step trains classification or learning to rank (LTR) algorithm. In both cases, we treat *central* as positive, and *garbage* and *neutral* as negative examples. However, *relevant* is excluded from the training stage not to introduce confusing examples.

For classification, we train J48 (CL-J48) and Random Forest (CL-RF) decision tree classifiers, as implemented in WEKA[4]. For LTR, we use the Random Forest (LTR-RF) approach as implemented in RankLib.[5] Thus, we take the same settings as described in [4] and [3].

5. EXPERIMENTS

5.1 Feature reduction

We followed two steps to select a small set of effective features: preliminary elimination and subsequent forward selection. Preliminary elimination was done in two ways. First, we ran an experiment with and without temporal features and observed that the collective contribution of temporal features to performance was negligible. Next, from document-entity features, we excluded all features that are based on partial matching such as features that use the matching of a person's last name. These features are already integrated in our new or adapted features. The preliminary elimination step helps reduce the large feature set to a smaller manageable set for the subsequent forward selection method. After preliminary elimination, there remain 26 features (15 document-entity, 6 document, and 5 new or adapted) listed in 5.2. Next, we apply the forward selection method on these remaining features: add one feature at a time and study its contribution to performance. Based on this, we select an even fewer, but effective set of features.

5.2 List of features

The selected feature set is listed here. The context features are new in the sense they are not used for CCR before. GCLD is as used in [1], and PPR is an adaptation from [2]. The rest of the features are as implemented in [4] and [3].

Google's Cross Lingual Dictionary (GCLD). This is a mapping of strings to Wikipedia concepts and vice versa [5]. The GCLD corpus estimates two probabilities: (1) the probability with which a string is used as anchor text to a Wikipedia entity and (2) the probability that indicates the strength of co-reference of an anchor with respect to other anchors to a given Wikipedia entity. We use the product of both for each string.

[3]http://krisztianbalog.com/files/resources/oair2013-kba/runs.zip.

[4]http://www.cs.waikato.ac.nz/∼ml/weka/

[5]http://people.cs.umass.edu/∼vdang/ranklib.html

PPR. For each entity, we computed a PPR score from a Wikipedia snapshot, keeping the top 100 entities along with the corresponding scores.

Surface Form (sForm). For each entity, we gathered DBpedia redirects, labels and names.

Context (contxL, contxR). From the WikiLink corpus [6], we collected context sentences (2 left and 2 right) and generated n-grams between uni-grams and quadro-grams. we select the 5 most frequent n-grams for each context.

LengthTitle. Term count of document title.

LengthBody. Term count of document body.

LengthAnchor. Term count of document anchor(s).

Source . Document source (news, social, or linking).

English 0,1. Document's language is English or not.

MentionsTitle. No. of occurrences of the target entity in the document title.

MentionsBody . No. of occurrences of the target entity in the document body.

MentionsAnchor. No. of occurrences of the target entity in the document anchor(s).

FirstPos. Term position of the first occurrence of the target entity in the document body .

LastPos . Term position of the last occurrence of the target entity in the document body.

Spread. Spread, i.e., distance between first and last occurrences.

SpreadNorm . Spread, normalized by the document length.

FirstPosNorm. Term position of the first occurrence of the target entity in the document body normalized by the document length.

LastPosNorm . Term position of the last occurrence of the target entity in the document body normalized by the document length.

SpreadNorm . Spread, normalized by the document length.

RelatedTitle . No. of different related entities mentioned in the document title.

RelatedBody . No. of different related entities mentioned in the document body.

RelatedAnchor. Number of different related entities mentioned in the document anchor(s).

jac. Jaccard similarity between the document and the entity's Wikipedia page.

cos. Cosine similarity between the document and the entity's Wikipedia page.

kl. KL-divergence between the document and the entity's Wikipedia page.

5.3 Baseline runs

We use three baselines, one from each category (string-matching, classification and LTR) that achieves the highest performance. For string-matching, we use [2] (LRE-KBA). For classification, the 2-step approach is used as a baseline (MC-RF). The third baseline, representing the state-of-the-art LTR category, which also uses 2-step approach, but trains a LTR algorithm instead of a classifier [4] (MC-LTR-RF).

5.4 Evaluation

We use the official TREC-KBA evaluation metrics [7]. Peak F scores averaged across the entities are used to compare system performances. Also, we use scaled utility (SU), the secondary TREC KBA official metric. SU measures the ability of a system to reject non-relevant documents and accept relevant documents. We use TREC-KBA 2012's evaluation script to generate our performance scores.

6. RESULT AND ANALYSIS

Figure 1 shows the performance (F) of the three algorithms against feature addition. The features are sorted from left to right, in descending order, in terms of information gain. The plus sign on a feature indicates that we incrementally add the feature into the feature set to the left of it.

From Figure 1, we see that the performance of the three algorithms increases with the addition of features to the initial feature set, reaches a maxima and then decreases. We can see that the three algorithms reach their respective maxima within the first 13 features. The addition of features do not

Figure 1: Performance (F) of classification and LTR algorithms against feature addition.

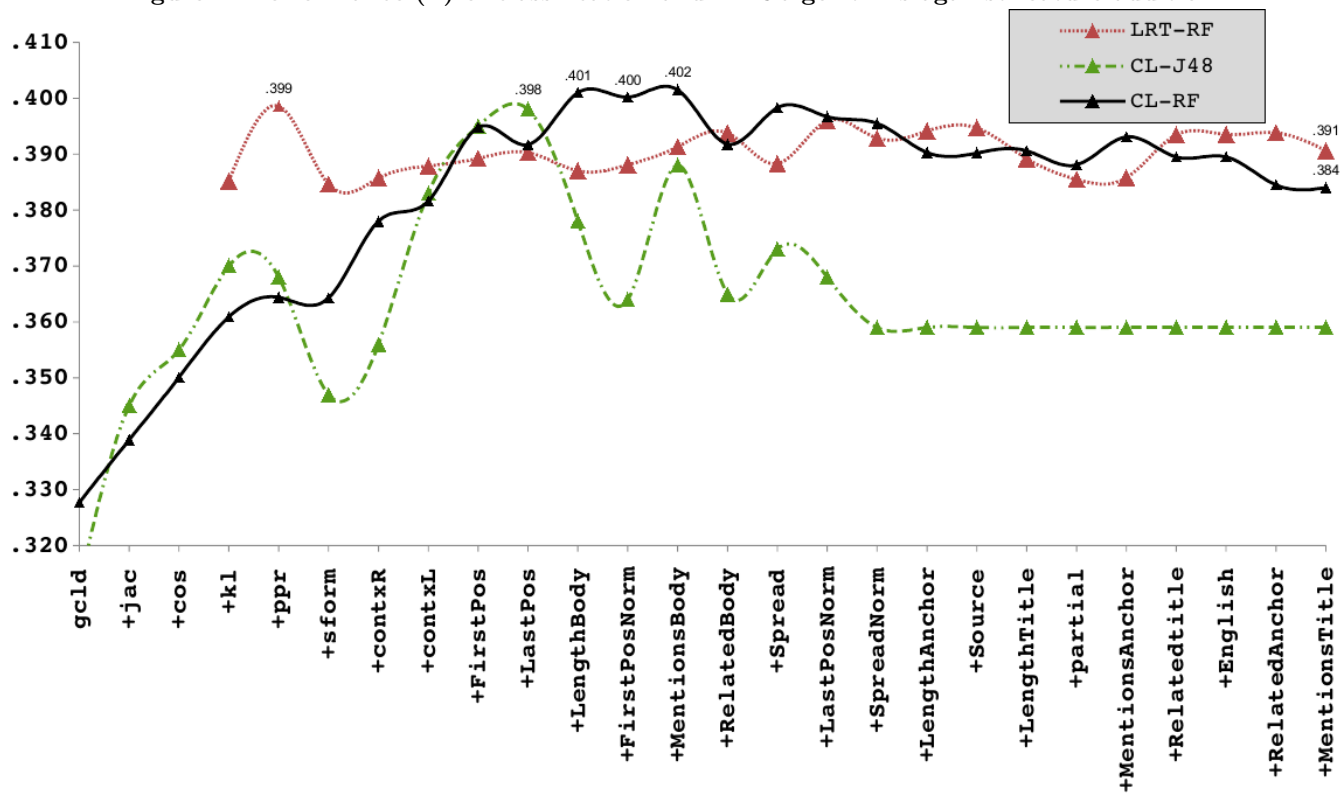

Table 1: Performances comparison of our approach (lower half) with baselines (upper half). Best scores are highlighted.

Method	F	SU
MC-RF	.360	.263
MC-LTR-RF	**.390**	**.369**
LRE-KBA	.377	.329
CL-RF	**.402**	**.396**
LTR-RF	.394	.411
CL-J48	.388	.306

improve results (in fact, performance deteriorates). Table 1 lists the best F scores as well as SU for each of the settings.

The scores in Table 1 show that our reduced feature set performs better than the baselines on both performance measures. The advantage of having a small set of powerful features is that they are easy to implement. The most informative features, as measured by information gain and contribution to performance, are the name variants (GCLD), similarity features (cos, jac, kl), related entities (PPR), context, position of entity mention in the document, and length of body text. These features can serve as baseline features for CCR task.

A surprising observation is that the approach using Classification Random Forest outperforms that using LTR Random Forest. This contrasts with the claim in previous work [4], that LTR algorithms outperform classification algorithms. Clearly, the conclusion that a certain approach outperforms another given a set of features does not mean that if the set of features is changed this conclusion still holds.

6.1 Statistical Significance of Results

Random Forest (RF) has achieved the best scores. Since RF results can vary from run to run, it becomes important to check their stability. To do so, we estimated the 95 confidence interval. For each addition of new feature, we run RF with 10 different random seed initialization and compute the confidence interval. The plot of Classification Random Forest (CL-RF) in Figure 1 is based on the mean performance for 10 different random initializations. The best result achieved with classification Random Forest is 0.402 ± 0.016 (95% confidence limits).

7. CONCLUSION

In this paper, we have studied the CCR challenge with a focus on feature selection and a subsequent comparisons of approaches. We started with a large feature set proposed in the literature, employed a preliminary feature elimination and a subsequent forward selection method to study the contribution of each element of the reduced feature set to performance We found that with reduced feature set, improved performance can be achieved compared to the full feature set both in terms of classification and learning-to-rank. We believe having a small selection of powerful features is advantageous because they (1) are easy to implement, and (2) achieve better performance. An important finding is that with the reduced but more effective set of features, a classification-based approach outperforms a learning-to-

rank-based approach, contradictory to what was found in a previous study. This suggests that when comparing CCR approaches, e.g., classification vs. learning to rank, conclusions do not only depend on the type of classifier or rankers, but also the set of features used, and we should be careful in generalizing conclusions

Acknowledgment

This study is financed by the COMMIT program as part of the Infiniti project.

8. ADDITIONAL AUTHORS

Jiymy Lin[§]
University of Maryland, College Park

jimmylin@umd.edu

9. REFERENCES

[1] S. Araujo, G. Gebremeskel, J. He, C. Bosscarino, and A. de Vries, "CWI at TREC 2012, KBA track and session track," *TREC*, 2012.

[2] X. Liu and H. Fang, "Leveraging related entities for knowledge base acceleration," *TREC*, 2013.

[3] K. Balog, H. Ramampiaro, N. Takhirov, and K. Nørvåg, "Multi-step classification approaches to cumulative citation recommendation," in *OAIR*, 2013, pp. 121–128.

[4] K. Balog and H. Ramampiaro, "Cumulative citation recommendation: Classification vs. ranking," in *SIGIR*. ACM, 2013, pp. 941–944.

[5] V. I. Spitkovsky and A. X. Chang, "A cross-lingual dictionary for english wikipedia concepts." in *LREC*, 2012, pp. 3168–3175.

[6] S. Singh, A. Subramanya, F. Pereira, and A. McCallum, "Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia," Tech. Rep. UM-CS-2012-015, 2012.

[7] 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," *TREC*, 2012.