Bi-directional Linkability From Wikipedia to Documents and Back Again: UMass at TREC 2012 Knowledge Base Acceleration Track

Jeffrey Dalton
University of Massachusetts, Amherst
jdalton@cs.umass.edu

Laura Dietz
University of Massachusetts, Amherst
dietz@umass.edu

1 Introduction

This notebook details the participation of UMass in the Cumulative Citation Recommendation task (CCR) of the TREC 2012 Knowledge Base Acceleration Track. UMass' objective is to introduce a single model for Knowledge Base Entity Linking and KB Acceleration stream filtering using bi-directional linkability between knowledge base (KB) entries and mentions of the entities in documents. Our system focuses on estimating bi-directional linkability between documents and Knowledge Base entities which measures compatibility in two directions: (1) from a KB entity to documents and (2) from mentions of entities in documents to their KB entries. The KB entity to document direction, is modeled as a retrieval task where the goal is to identify the most relevant documents for an entity in the evaluation time range. We observe that the other direction, from mention to KB entity, is very similar to the TAC Knowledge Base Population Entity Linking Task. The major goal of our participation is to explore how these two directions, from KB to documents and back can be modeled together to combine evidence from both linking directions.

For KBA, the goal is to identify documents from a stream that are central for a given entity. Our submissions consist of three stages: First, potentially relevant documents are retrieved from the stream. Second, potential mentions of the target entity are identified in the retrieved documents. Third, bi-directional links between the potential mentions and the target entity are established or dismissed, giving rise to a filtered set of central documents. Notice, that the third stage is closely related to the entity linking problem of TAC KBP.

The baseline run gathers name variations from the Wikipedia KB entry and incorporates them into the probabilistic retrieval of stream documents. Our experimental runs further include important NER spans and contextual terms using Latent Concept Expansion from annotated documents from the training time range. Also some experimental runs leverages bi-directional linkability using a supervised re-ranking approach trained on TAC KBP entity linking data as a measure on how likely potential mentions in the stream document refer to the target KB entry.

Our experiments show that incorporating entity context from query expansion methods provides significant gains both in precision and recall over the baseline, with all of our experimental runs outperforming the median. Further, our best performing run uses linkability evidence from both directions by using the TAC Entity Linking model.

2 Method

Our method to estimate linkability in both directions uses graphical latent variable models that combine probabilistic retrieval and extraction models. In each direction, we first generate a high recall set of candidates using the Markov Random Field retrieval model to construct a query model that includes a model of entity context. The retrieval model includes name variations, surrounding words and NER spans which are identified from text associated with the target entity. We experiment with various methods for estimating the model of an entity, using Latent Concept Expansion (LCE) (Metzler and Croft, 2007) to incorporate cross-document evidence from the corpus using relevance feedback and pseudo-relevance feedback. The result is a focused set of candidate documents and knowledge base entries, ranked by the likelihood of referring to the same entity. This set is either used directly, or acts as input to more advanced inference methods.

3 Corpus Processing

Our retrieval models are implemented using Galago[1], an open source retrieval engine which is part of the Lemur

[1] http://www.lemurproject.org/galago.php

Month	Documents	Collection Length	Index Size (GB)	Total Size (GB)
October 2011	36,547,282	54,33,597,431	22	245
November 2011	55,434,234	14,529,421,474	55	673
December 2011	62,773,692	16,058,713,120	62	739
January 2012	60,799,418	16,983,265,272	64	781
February 2012	58,147,836	18,488,791,637	67	833
March 2012	50,857,928	19,388,982,395	67	871
April 2012	33,796,674	14,217,201,526	51	835
May 2012	395,732	447,158,725	1	21
Total	358,752,796	100,113,534,149	389	4998

Table 1: KBA Galago Shard Statistics

toolkit. Galago supports indexing of large scale data in a distributed cluster environment with a MapReduce-like framework called TupleFlow.

Both the KBA stream corpus and the Wikipedia knowledge base are indexed to efficiently support bi-directional linking queries.

3.1 KBA Stream Corpus

The cleansed documents with NER information from the KBA stream corpus are indexed with Galago, stripping out HTML tags. No stemming or stopword removal is performed. In order to filter the stream by time stamp and source type (e.g. linking, social, news), we index this information in Galago fields. Further NER information is preserved in the documents, to be used in relevance feedback queries.

For efficiency we create a separate index shared for each month. Indexing each shard took between four and eight hours. Per-shard collection statistics are given in Table 1.

3.2 Wikipedia Knowledge Base

For both KBA and TAC KBP we use a Freebase Wikipedia Extraction (WEX) dump of English Wikipedia from January 2012 which provides the Wikipedia page in machine-readable XML format and relational data in tabular format. The Freebase dump contains 5,841,791 entries. We filter out non-article entries, such as category pages. The resulting index contains 3,811,076 KB documents and over 60 billion words.

The goal is to create an index with fields for: anchor text (within Wikipedia as well as from the Web), Wikipedia categories, Freebase names, Freebase types, redirects, article titles, and full-text for each article. Most of this information is contained in the WEX dump. We also incorporate external web anchor text to Wikipedia entries using the the Google Cross-Wiki dictionary, which contains 3 billion links and 297 million associations from 175 million unique anchor text strings.

The anchor extraction from the WEX dump is performed using the SPARK parallel processing framework,[2] which allows fast in-memory computation over large scale data in a cluster. The final merge of full-text and WEX meta-data with Google Cross-Wiki dictionary is performed using Hadoop MapReduce using the PIG parallel processing language.

4 KB Entities to Documents

For each target entity from Wikipedia, the first step is to retrieve a high recall set of stream documents. First, name variants and potentially disambiguating context is extracted from the target's Wikipedia article. We leverage the stream corpus to reconsider and re-weight disambiguating context by confidence. From these ingredients, we build a retrieval query against the stream corpus.

The goal is to identify:

- the target entity's name,
- name variants by which the entity is referred to,
- disambiguating contextual words,
- disambiguating related named entities.

4.1 Extracting Name Variants and Disambiguating Context

The canonical name of the target entity is taken from the title of the Wikipedia article.

Name variants for the Wikipedia entry are gathered from the title field, redirects, Freebase names, disambiguation links, and incoming anchor text.

Related named entities are taken from titles of in- and outlinks of the target's Wikipedia page.

4.2 Entity Modeling using Latent Concept Expansion

We estimate disambiguating context from external document evidence using Latent Concept Expansion (LCE) (Bendersky and Croft, 2008). LCE is a query expansion

[2] http://www.spark-project.org/

#combine:0=(λ_{T}+λ_{NV}):1=λ_{CW}:2=λ_{NER}(
  #combine:0=λ_{T}:1=λ_{NV}(
    #seqdep(entity-name)
    #combine(#seqdep(nv_0) ... #seqdep(nv_n))
)
  #combine:0 = ϕ_0^{CW} : ... k : $\phi_k^{\text{CW}}(cw_0, \ldots, cw_k)$
  #combine:0 = ϕ_0^{NER} : ... k : ϕ_k^{NER}(
    #seqdep(ner_0), ..., #seqdep(ner_k)
)
)

Figure 1: LCE query for retrieving relevant stream documents in Galago query syntax. The query includes the entity name, name variants, context words, and NER spans.

technique for estimating contextual evidence built upon the Markov Random Field retrieval framework. We use LCE to model dependencies between related entities by including NER name spans as types of concepts. LCE estimates the context of an entity from documents that are relevant to the target entity. The intuition is that the reliability of words and named entities increases the more often they occur in documents relevant to the target entity.

For relevance feedback we use the set of relevant documents from the pre-cutoff sample documents. In one experimental run we also add post-cutoff documents using pseudo-relevance feedback.

The top k words under the LCE model are used as disambiguating contextual words with weights ϕ^{CW}.

We apply LCE to estimate the confidence in named entities extracted from the Wikipedia link structure. Further, the set is combined with NER spans that frequently occur in the relevant document. After aggregating the top k named entities are used as disambiguating related named entities with weights ϕ^{NER}.

We decided against using only NER context from LCE, because the corpus may be biased towards one event in time, and the link information from Wikipedia is an important source of long-term hand-constructed entity context information.

4.3 Retrieving Relevant Stream Documents

For a given entity E, A query model M_E is build from the gathered name variants nv and k weighted disambiguating contextual words cw and k named entities ner. The query is given in Galago's query syntax in Figure 1.

The query model scores the documents in the collection using a log-linear weighted combination of the matches of the concepts K and rank the documents using this score. The score the occurrence of a concept in document using the log of the probability of a concept, k, given a document D with Dirichlet smoothing, i.e.,

$$f(k, D) = \log \frac{tf_{k,D} + \mu \frac{tf_{k,C}}{|C|}}{|D| + \mu} \quad (1)$$

where $tf_{k,D}$ is the number of occurrences of the concept in the document, $tf_{k,C}$ is the number of occurrences in the collection, $|D|$ is the number of terms in document, $|C|$ is the number of words in the collection, and μ is the smoothing parameter that is set empirically.

The model is run using the Galago search engine to score all of the stream documents. The result of retrieval is a linkability score in the direction of Wikipedia entity to documents which can be used as-is or re-ranked further.

5 Entity Mentions to Wikipedia

We estimate linkability in the opposite direction from document mention to Wikipedia entity using a linking model developed for the TAC KBP Entity Linking task.

5.1 Identifying Mentions of the Target Entity

For each candidate document retrieved for a target entity, we extract potential mentions of the target. For each document, we select a canonical target entity mention by searching for the name or name variants. Matches are identified with string matches ignoring case and punctuation, preferring stricter matches and high confidence name variants. If no canonical matches can be found, a dummy empty mention is created.

5.2 Re-ranking Mentions to Match the Target Entity

Next, each of the canonical mentions is linked against Wikipedia entities—which is the direction evaluated in the entity linking task of the TAC KBP competition. We train supervised discriminative ranking model with TAC entity linking data from years 2010 and 2011. It incorporates features based on string similarity of names, similarity of term vectors, and name confidence based on ambiguity of anchor texts.

A full list of features is given in Table 2. For the supervised ranking we use the RankLib[3] Learning to Rank toolkit. We experimented with various models including AdaRank and Coordinate Ascent. A coordinate ascent model was used based on its performance on the TAC 2011 linking evaluation data.

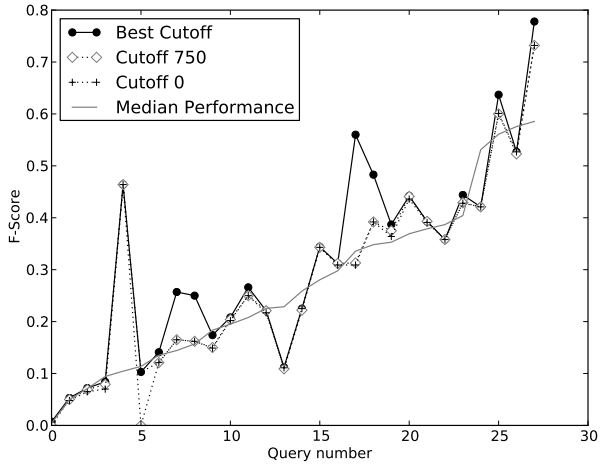

Figure 2: F-Score performance over queries at different cutoff thresholds. Queries are sorted by difficulty in terms of median F-score.

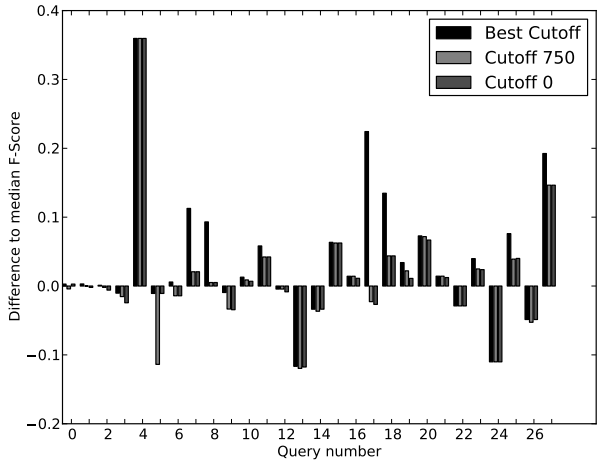

Figure 3: Difference in F-score to the median performance over queries.

6 Experimental Results

6.1 Setup

We now describe the parameter setting used for the model. For scoring with Equation 1 we use the default smoothing value, $\mu = 2000$. It is important to note that we only used background term statistics from the training time range. For the free parameters in our Sequential Dependence (SD) sub-models we estimate the parameters using training data from the TAC KBP 2010 entity linking data, setting $\lambda_{T_D} = 0.29$, $\lambda_{O_D} = .21$, and $\lambda_{U_D} = 0.50$. These parameters place greater emphasis on the ordered window and term proximity, which is logical since the queries consist largely of names. We use the LCE context model to retrieve and rank the stream documents. We manually set the concept type weights as: $\lambda_T = 0.3$, $\lambda_{NV} = 0.3$, $\lambda_{CW} = 0.2$, $\lambda_{NER} = 0.2$. These parameter setting are similar to the default LCE settings, which provides half the weight to the original query and half to expansion terms.

The result of running the query is an unnormalized log probability. To produce a score in the 1 to 1000 confidence range, we exponentiate the log probability and normalize by the maximum score.

6.2 Run comparison

In this section, we compare the runs submitted to the CCR task of the TREC 2012 KBA Track. The runs we submitted are variations of the models described previously. The descriptions of the runs follow:

1. NV Full Stream – This a baseline run using the entity name and name variations only, scoring the full stream (TTR + ETR) documents , with $\lambda_T = 0.5$, $\lambda_{NV} = 0.5$ $\lambda_{CW} = 0$, $\lambda_{NER} = 0$. The highest scoring 6000 documents are returned by the run. (submitted run ID:FS_NV_6000)

2. NV – This run uses entity name and name variations only, scoring the post-cutoff (ETR) documents , with $\lambda_T = 0.5$, $\lambda_{NV} = 0.5$ $\lambda_{CW} = 0$, $\lambda_{NER} = 0$. The highest scoring 1500 documents are returned by the run. (submitted run ID: PC_NV_150050)

3. LCE10 – This run employs explicit relevance feedback on the TTR documents using Latent Concept Expansion to estimate related context words (CW) and NER names (NER) using 10 expansion concepts per type. The parameter setting used are: $\lambda_T = 0.3$, $\lambda_{NV} = 0.3$, $\lambda_{CW} = 0.2$, $\lambda_{NER} = 0.2$. The highest scoring 1500 documents are returned by the run. (submitted run ID:PC_RM10_150050)

4. LCE20 – This run employs explicit relevance feedback on the TTR documents using Latent Concept Expansion to estimate related context words (CW) and NER names (NER) using 20 expansion concepts per type. The parameter setting used are: $\lambda_T = 0.3$, $\lambda_{NV} = 0.3$, $\lambda_{CW} = 0.2$, $\lambda_{NER} = 0.2$. The highest scoring 1500 documents are returned by the run. (submitted run ID: PC_RM20_150050)

5. LCE10+TAC – This run uses LCE10 to retrieve a candidate set of results. Then, TAC EL queries are generated from these candidates. The supervised TAC EL ranker is applied and the results re-ranked with respect to the target entity only. The highest scoring 1500 documents are returned by the run. (submitted run ID:PC_RM10_TACRL50)

[3] http://www.cs.umass.edu/~vdang/ranklib.html

6. LCE10 + TAC + PRF – The goal of this run is to improve recall using pseudo-relevance feedback (PRF) over the entire post cutoff stream. The initial query is generated from relevance feedback using LCE on the pre-cutoff training documents using the results from LCE10. Then, the top 50 retrieved documents are re-ranked using the TAC entity linking supervised ranker. The highest scoring 10 documents are used to generate a PRF query model over the post-cutoff (ETR) document set. The PRF query model uses The parameter settings: $\lambda_T = 0.3$, $\lambda_{NV} = 0.3$, $\lambda_{CW} = 0.2$, $\lambda_{NER} = 0.2$. The highest scoring 2000 documents are returned. (completed after deadline)

A summary of the results are shown in Table 3. It's clear that the LCE context models outperform using name variants only. Additional improvement is made applying the TAC supervised ranking model to results retrieved using LCE. It does not appear that pseudo-relevance feedback using the evaluation time documents provided any additional benefit. This seems to indicate context models using only the training documents are just as effective as models incorporating evidence from the full stream. Overall, it appears that combining bi-directional evidence from LCE to rank documents and the TAC entity linking model outperforms other models.

6.3 Further Analysis

We examine the query-by-query performance of the our top performing run, LCE10+TAC model in Figure 2 and Figure 3 and how it compares with other teams. The results show that for our optimal cutoff over 68.9% of our runs are above the median. However, if our overall best average cutoff is used, 55.2% of queries are above the median. Our best performing runs are Basic_Element_(music_group), Jim Steyer, Nassim_Nicholas_Taleb, and James_McCartney. The worst performing queries in order are Basic_Element_(company), Boris_Berezovsky_(businessman), Satoshi_Ishii, Darren_Rowse, and William_D._Cohan. It is interesting to note that all the cutoff values correlate highly, with 750 and cutoff 0 both perform comparably despite retrieving very different numbers of results. Consequently, choosing a particular cutoff value to evaluate is difficult. The reasons for the correlation is unclear, but we hypothesize that it may have to do with the distribution of the annotation data since only judged negative documents are counted as false positive examples. It's likely that higher cutoff values that retrieve fewer results will perform better if additional negative examples are included in the evaluation.

Method	Best F-Score
NV Full Stream	0.277
NV	0.274
LCE10	0.298
LCE20	0.293
LCE10+TAC	**0.305**
LCE10+TAC+PRF	0.299
TREC Avg. Median	0.289

Table 3: Comparison of Best F-Score of the runs. Best result appears in boldface.

7 Conclusions

In our submissions to KBA we utilize bi-directional linkability between Wikipedia and documents to estimate centrality. We attempted to combine evidence from both directions: from an entity to documents and back. We present a single model that uses graphical latent variable models with probabilistic retrieval to generate a focused set of candidates, rank the results, and combine evidence from cross-document entity context. Our experiments show that incorporating evidence from mention to entity using a TAC linker can result in improvements over LCE models that only use evidence in one direction from entity to documents.

One potential weakness of our submission is that we did not deal with queries that may be highly temporal in nature. We do not explicitly model the stream structure of the KBA corpus in our runs. However, we note that there is significant recent work using temporal relevance feedback (Keikha et al., 2011) that could used to estimate a more dynamic model of the entity.

The current use of bidirectional information in our model is limited. The TAC entity linking model is used mainly as a re-ranker. Combining these two tasks is challenging because the linkability evidence is not symmetric, with different sources of evidence in either direction. We intend to explore ways of modeling context and combining these two linkability directions further in our future submissions.

8 Acknowledgment

This work was supported in part by the Center for Intelligent Information Retrieval, in part under subcontract #19-000208 from SRI International, prime contractor to DARPA contract #HR0011-12-C-0016, and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

M. Bendersky and W.B. Croft. 2008. Discovering key concepts in verbose queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 491–498.

M. Keikha, S. Gerani, and F. Crestani. 2011. Temper: A temporal relevance feedback method. *Advances in Information Retrieval*, pages 436–447.

D. Metzler and W.B. Croft. 2007. Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318. ACM.

Feature Name	Type	Description
wordMatch	name variants	Number of words occuring in both names
wordMiss	name variants	Number of words missed in the query string
substringTest	name variants	1.0 if one name is substring of the other (ignoring casing); otherwise 0.0
editDistance	name variants	Levenshtein String edit distance between query mention and Wikipedia title
tokenDice	name variants	Dice coefficient on name token sets
tokenJaccard	name variants	Jaccard index on name token sets
totalSourcesMatching	name variants	Counts matching in multiple sources, e.g. anchor text, title, freebase name, and redirect
exactMatchCount_anchor-exact	name variants	Number of wikipedia anchor texts that matches the query string (ignoring casing and punctuation)
exactMatchBool_anchor-exact	name variants	1.0 if above score non-zero; otherwise 0.0
exactMatchCount_web_anchor-exact	name variants	Number of web anchor texts that matches the query string (ignoring casing and punctuation) according to the Google Cross-Wiki dictionary
exactMatchBool_web_anchor-exact	name variants	1.0 if above score non-zero; otherwise 0.0
exactMatchCount_fbname-exact	name variants	Number of freebase names that matches the query string (ignoring casing and punctuation)
exactMatchBool_fbname-exact	name variants	1.0 if above score non-zero; otherwise 0.0
exactMatchCount_redirect-exact	name variants	Number of redirect page titles that matches the query string (ignoring casing and punctuation)
exactMatchBool_redirect-exact	name variants	1.0 if above score non-zero; otherwise 0.0
exactMatchCount_title-exact	name variants	Number of page titles that match the the query string (ignoring casing and punctuation)
exactMatchBool_title-exact	name variants	1.0 if above score non-zero; otherwise 0.0
weakAlias	name variants	1.0 if names match according to dice, acronym, or substring test; otherwise 0.0
fieldLikelihood_anchor	name variants	Unigram Query likelihood (as unnormalized log-prob) of the query mention under the Wikipedia anchor text's language model
fieldProbability_anchor	name variants	N-gram probability of the query mention under the Wikipedia anchor text's language model
fieldLikelihood_fbname	name variants	Unigram Query likelihood (as unnormalized log-prob) of the query mention under the Freebase name dictionary's language model
fieldProbability_fbname	name variants	N-gram probability of the query mention under the Freebase name dictionary's language model
fieldLikelihood_redirect	name variants	Unigram Query likelihood (as unnormalized log-prob) of the query mention under the redirect pages' language model
fieldProbability_redirect	name variants	N-gram probability of the query mention under the redirect pages' language model
fieldLikelihood_web_anchor	name variants	Unigram Query likelihood (as unnormalized log-prob) of the query mention under the web anchor text's language model
fieldProbability_web_anchor	name variants	N-gram probability of the query mention under the web anchor text's language model
fieldLikelihood_title	name variants	Unigram Query likelihood (as unnormalized log-prob) of the query mention under the title's language model
fieldProbability_title	name variants	N-gram probability of the query mention under the title's language model
diceTestFullCharacterScore	name variants	Dice coefficient of character sets.
diceTestFullCharacter	name variants	1.0 if above score > 0.9; otherwise 0.0
diceTestAlignedCharacterScore	name variants	Maximum character dice score of left- and right aligned character sets.
diceTestAlignedCharacter	name variants	1.0 if above score > 0.9; otherwise 0.0
diceTestFullWordScore	name variants	Dice coefficient words sets; lower cased and tokenized on white space and punctuation.
diceTestFullWord	name variants	1.0 if above score > 0.9; otherwise 0.0
diceTestAlignedWordScore	name variants	Maximum character dice score of left- and right word sets; lower cased and tokenized on white space and punctuation.
diceTestAlignedWord	name variants	1.0 if above score > 0.9; otherwise 0.0

Feature Name	Type	Description
galagoscore	name, context words, ner	Retrieval score of this candidate, taken from the Galago candidate retrieval model.
galagoscoreNorm	name, context words, ner	Retrieval score of this candidate, normalized over all candidates in the retrieved set.
inlinks	entity	Log number of Wikipedia inlinks - a measure of popularity
stanfExternalinlinks	entity	Log number of web inlinks - a measure of popularity
linkProb	entity	If a name matches the Wikipedia anchor text, probability that the matching anchor text refers to only this entity (versus other entities)
externalLinkProb	entity	If a name matches the web anchor text, probability that the matching anchor text refers to only this entity (versus other entities)
cosineFeature-doc	document	TF-IDF weighted cosine similarity of terms between the query document and Wikipedia article.
jaccardFeature-doc	document	Jaccard coefficient of document term vectors (of query document and article)
jsdivergenceFeature-doc	document	Jensen-Shannon divergence between Dirichlet smoothed document language models (of query document and article)
klfFeature-doc	document	KL divergence of the query document's Dirichlet smoothed language model and the article's language model.

Table 2: Features of the query mention and candidate Wikipedia entity.