A Neighborhood Relevance Model for Entity Linking

Jeffrey Dalton
University of Massachusetts
140 Governors Drive
Amherst, MA, U.S.A.
jdalton@cs.umass.edu

Laura Dietz
University of Massachusetts
140 Governors Drive
Amherst, MA, U.S.A.
dietz@cs.umass.edu

ABSTRACT

Entity Linking is the task of mapping a string in a document to its entity in a knowledge base. One of the crucial tasks is to identify disambiguating context; joint assignment models leverage the relationships within the knowledge base. We demonstrate how joint assignment models can be approximated with information retrieval. We introduce the neighborhood relevance model which uses relevance feedback techniques to identify the salience of entity context using cross-document evidence. We show that this model is more effective than local document models for ranking KB entities. Experiments on the TAC KBP entity linking task demonstrate that our model is the best performing system for strings that are linkable to the knowledge base.

1. INTRODUCTION

Entity linking is important because most content created is unstructured text in the form of news, blogs, forums, and microblogs such as Twitter and Facebook. A key challenge is to link these unstructured text documents to the Web of Data. Entity linking bridges the structure gap between text documents and linked data by identifying mentions of entities in free text and linking them to knowledge bases. It enriches unstructured documents with links to people, places, and concepts in the world. Entity linking is a fundamental building block that supports a wide variety of information extraction, document summarization, and data mining tasks. For example, linked entities in documents can be used to expand existing knowledge base entries with new facts and relationships.

The major challenge in entity linking is ambiguity. An entity mention in text may be ambiguous for a wide variety of reasons: multiple entities share the same name (e.g. Michael Jordan), entities are referred to incompletely (e.g. Justin for Justin Bieber), by pseudonyms or nicknames (Christopher George Latore Wallace is also known as The Notorious B.I.G.), and are often abbreviated (e.g. UW for the University of Wisconsin as well as University of Washington).

The entity linking problem has been studied over several years in the TAC Knowledge Base Population venue with the following task definition:

Entity Linking: Given a string mention q in a document, predict the entity c^* in the knowledge base which the string represents, or NIL if no such entity is available.

A typical entity linking system consists of four phases: 1) query expansion, 2) candidate generation, 3) entity ranking, and 4) handling NIL cases. The goal of the first two steps is to achieve a high-recall set of Wikipedia entities. Given the candidate set, most effective approaches, e.g., [15, 4, 16], leverage contextual entities as disambiguating evidence in step 3. One issue is the candidate generation step is often performed using string matching heuristics, resulting in large candidate sets that may contain hundreds or thousands of entities for ambiguous matches. The connection between candidate generation and ranking are often separate and not well aligned.

We advocate an information retrieval approach that uses one probabilistic model to approach steps 1-3. We introduce our linking system, KB Bridge. Supplementary materials for this work is available on the KB Bridge website[1]. Existing entity linking methods only employ IR to a minor degree. We model the entire linking task as a retrieval problem, including formalising joint assignment models within the retrieval framework. The graphical modeling framework allows us to ground our work on models from both information extraction and information retrieval.

For a given entity mention, the correct knowledge base entry is likely to share important pieces of contextual information: lexically similar names, shared topical similarity reflected in word usage, and shared named entities. Additional context could be included, but throughout the paper we focus the previously described context: name variations, surrounding sentences, and neighboring entities.

Entity linking provides some unusual challenges. The typical IR setting addresses short queries by using relevance feedback [17] to expand the query model. In entity linking, the query is an entity string embedded in a longer document, providing an abundance of context which could be leveraged. However, not all context is equally helpful, either because of ambiguity, heterogeneity in topic, or spurious collocations. Consider the example "ABC shot the TV drama Lost in Australia." with the task of linking "ABC" to the entity "American Broadcasting Company". The named entity span "Australia" is not relevant for the true answer. It might

OAIR'13, May 22-24, 2013, Lisbon, Portugal.
Copyright 2013 CID 978-2-905450-09-8 .

[1] http://ciir.cs.umass.edu/~jdalton/kbbridge

actually misguide the process to link to the wrong entity "Australian Broadcasting Corporation".

We introduce the neighborhood relevance model to estimate the salience of context with the goal of filtering and weighting (as opposed to expanding) the context model. The neighborhood relevance model is based on ideas of pseudo-relevance feedback [20] and latent concept expansion [13] to leverage cooccurrence evidence across topically similar documents. Our main contributions are:

- An unsupervised approach to entity linking based upon the Markov Random Field information retrieval model that provides competitive performance out-of-the-box.
- A unified retrieval based approach to linking combining candidate generation and ranking in a single retrieval framework, with more than 95% recall in the highest ranked 10 entities.
- A mention-specific approach for identifying salient neighboring entities using across-document evidence based on relevance feedback.
- Demonstrating the benefits of the entity neighborhood relevance model in combination with a supervised learning to rank framework, resulting in the best ranking for entities linkable to the knowledge base for the TAC KBP task.

2. BACKGROUND AND RELATED WORK

2.1 Related Work

Early work on entity linking was performed by Bunescu and Pasca [2] and Cucerzan [3] to link mentions of topics to their Wikipedia pages. In contrast to their models, we focus on a retrieval approach that leverages text based ranking without exploiting extensive Wikipedia-specific structure.

Our work is related to that of Gottipati and Jiang [5] who apply a language modeling approach to entity linking. They expand the original query mention with contextual information from the language model of the document. We use the local weighting as a starting point for estimating the entity salience and compare against it as a baseline.

Entity linking has been studied in a variety of recent venues. At INEX the "Link the Wiki" task explored automatically discovering links that should be created in a Wikipedia article [7]. More recently, it is one of the principle tasks studied at the ongoing Text Analysis Conference Knowledge Base Population track (TAC KBP). Ji et al. [9, 8] provide an overview of the recent systems and approaches.

Instead of linking individual mentions one at a time, recent work [3, 16, 19, 11, 6] focuses on linking the set of mentions, M, that occur in the query document d. These models perform joint inference over the link assignments to identify a coherent assignment of KB entries. In our work we leverage the set of mentions M in the document as context in an information retrieval model. In this work we instead focus on identifying salient entity mentions in the context, because mentions in the document may be spurious or only tangentially related. This is especially true if the document contains multiple topics.

2.2 Joint Neighborhood Assignment Model

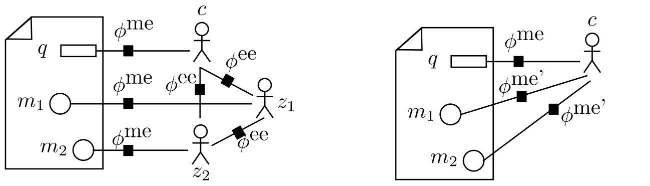
Graphical models [10], such as Markov Random fields, are a popular tool in both information extraction and information retrieval. Graphical models provide the mathematical framework for formalizing intuitions on how available data (e.g., the query mention) and quantities of interest (e.g. entity in the knowledge base) are connected. After casting data and quantities of interest as random variables, dependencies between two (or more) variables are encoded by factor functions ϕ that assign a non-negative score to each combination of variable settings. Factors ϕ are often expressed by a log-linear function of a feature vector. The joint configuration of all variables is scored by the likelihood function $\mathcal{L}$, which is represented by the normalized product over scores of all factor functions for the given variable settings.

The factor functions ϕ are designed to encode our intuition on which variable choices go well together. For instance, early approaches to entity linking [1] heuristically extract a set of entity candidates and use a graphical model with single factor $\phi^{\mathrm{me}}(q, c)$. The factor ϕ^{me} formalizes the intuition that query mentions q and entities c are compatible when their names match and the document surrounding q has terms similar to the article of the entity candidate c.

This basic model is extended in joint neighborhood assignment models [3, 16, 12]. The idea is that knowledge base entities which are mentioned in the same documents are also likely to be structurally related in the knowledge base. When the query mention is ambiguous, several candidate entities have an equally high score under the factor $\phi^{\mathrm{me}}(q, c)$. Joint assignment models explicitly incorporate entity mentions m from the same document as the query mention q, which we call neighbor mentions henceforth. Each neighbor mention m_i is associated with a latent variable z_i referring to the respective entity for the neighbor. We expect the neighbor entities z to help disambiguate the true candidate c. This expectation is formalized by a second factor function $\phi^{\mathrm{ee}}(z, c)$, which captures intuitions on when entities z and c are mentioned in the context of each other. For the factor function ϕ^{ee}, Ratinov et al. [16] set similarity of inlinks (and outlinks) for c and z; Cucerzan [4, 3] use the overlap in Wikipedia categories and presence of links from z to c and vice versa. For each neighbor mention m_i a set of candidates are heuristically identified, where the variable z_i represents a choice of the candidate.

The model is visualized in Figure 1a, where variables q and m_i are observed, but settings of variables c and z_i are chosen from respective candidate sets. A factor is represented as a small black box which connect its input variables. If each neighboring mention m is linked to its correct entity candidate z_i, the true candidate c will achieve a high score of $\phi^{\mathrm{ee}}(z_i, c)$ for most of the neighbors z_i. This information is combined across all neighbors and taken together with the name-factor $\phi^{\mathrm{me}}(q, c)$. The dilemma is that linking all m_i to z_i requires that we solve the entity linking problem as part of the solution. For this model, the joint inference problem does not have a closed-form solution and therefore requires approximate inference such as belief propagation [12] or iterative heuristics [4, 16].

As the task is to link only the single query mention, the neighboring entity links are to be marginalized out by integration over z. Candidates c for the query are scored by Equation 1.



(a) Joint Assignment Model(b) Neighborhood Query Model

Figure 1: Neighborhood Models

$$\mathcal{L}(c) = \phi^{\rm me}(q,c) \cdot \textstyle\prod_m (\int \phi^{\rm me}(m,z) \cdot \phi^{\rm ee}(z,c)\, dz) \qquad (1)$$

3. QUERY MODEL

In this section we integrate graphical models for entity linking as developed in the information extraction community with graphical models used for information retrieval. We utilize the fact that query models, such as query likelihood and the sequential dependence model [14] have an underlying graphical model that gives rise to a score of a document.

3.1 Neighborhood Query Model

We demonstrate how the retrieval engine can be used to infer the joint inference problem of Equation 1 whenever factor functions $\phi^{\rm me}$ and $\phi^{\rm ee}$ are expressible as query operators. This allows us to combine the candidate generation (step 2) with the entity ranking (step 3) and have it carried out by a retrieval engine that optimizes over all possible entities in the knowledge base at once. In contrast to the joint neighborhood assignment approaches, no candidate set has to be identified beforehand.

The key insight is that for particular choices of features for the name-match factor $\phi^{\rm me}(m,z)$ and the entity-link factor $\phi^{\rm ee}(z,c)$, it is possible to solve the integral over z in Equation 1, with smart preprocessing and indexing. For instance, consider the following simple factor functions for $\phi^{\rm me}$ and $\phi^{\rm ee}$. We choose $\phi^{\rm me}(m,z)$ to yield 1 if the mention m is a string match to the title of z and 0 otherwise; and $\phi^{\rm ee}(z,c)$ to yield 1 if z links to c. In this case, we can replace the integral over z with a factor $\phi^{\rm me'}(m,c)$ that yields 1 if the mention m matches a title of an inlink of c. Such a factor can be analytically solved by transforming the Wikipedia snapshot so that the indexed article of entity c is enriched with titles of Wikipedia pages that link to c in a separate field (or extents). A search query for the string m in the field of inlinks represents a graphical model that represents the score of $\phi^{\rm me'}$.

For many choices of factor functions for $\phi^{\rm me}$ and $\phi^{\rm ee}$, the knowledge base can be transformed to allow for efficient replacement factors $\phi^{\rm me'}$ that can be directly optimized within the retrieval model framework. By integrating z out, the joint neighborhood assignment model depicted in Figure 1a becomes the neighborhood query model in Figure 1b with the following likelihood function.

$$\mathcal{L}(c) = \phi^{\rm me}(q,c) \cdot \prod_m \phi^{\rm me'}(m,c) \qquad (2)$$

Many complex factor functions are possible, but for the remainder of this publication we use two simple, yet effective factor functions ϕ: Factor $\phi^{\rm me}(q,c)$ encodes matches of the string q and terms from the surrounding text in any field of the Wikipedia article of the candidate c, this includes names as well as the full-text of the article. Factor $\phi^{\rm me'}(m,c)$ matches the string representation of m in the full-text and titles entities that link to (and are linked from) the Wikipedia entry of c.

3.2 Relevance-weighted Neighborhood Query Model

As pointed out in the introduction, not all contextual entities are equally salient. For each contextual entity m, its salience for disambiguating query q is denoted by $\rho_q(m)$, ranging on a scale between 0 and 1. If the salience $\rho_q(m)$ is 0, we want to remove the effect of $\phi^{\rm me'}(m,c)$ on the likelihood function. Based on the geometric mean, which is the natural choice for averages of probabilities, we achieve the weighting with the geometric interpolated model of Equation 3.

$$\mathcal{L}(c) = \phi^{\rm me}(q,c) \cdot \prod_m \left(\phi^{\rm me'}(m,c)\right)^{\rho_q(m)} \qquad (3)$$

Notice, that the unweighted model follows as a special case where all saliences are 1.

We also introduce the parameter $\lambda^{\rm M}$ that allows trading off between the direct similarity of the query and candidate as expressed by $\phi^{\rm me}(q,c)$ and the aggregated influence of the M contextual entities. Exploiting that the sort-order induced by $\mathcal{L}$ is invariant with respect to logarithms, we cast the optimization in log-space as in Equation 4.

$$\log \mathcal{L}(c) = \log\phi^{\rm me}(q,c) + \lambda^{\rm M} \frac{1}{M} \sum_m \left(\rho_q(m) \log \phi^{\rm me'}(m,c)\right) \qquad (4)$$

3.3 Context from Query Document

The factor $\phi^{\rm me}$ formalizes our intuition on compatibility between the query mention q and the true candidate c. This includes name matches of the string representation of m with names listed in the knowledge base (e.g. title, redirect, anchor text). We further extend it to other similarity measures that are independent of the neighbor mentions (which are represented by $\phi^{\rm me'}$).

Name variations v of the query string can be extracted from the query document, to add robustness to the entity linking inference. This is especially important if the query string is an acronym or an ambiguous reference to the entity. We also incorporate the words, s, of the sentence that surround the query mention or one of its name variations to preserve verbs, adjectives, and standing expressions that might disambiguate the candidate.

We introduce separate weight parameters $\lambda^{\rm Q}$, $\lambda^{\rm V}$, $\lambda^{\rm S}$ to individually control influence of name-matches of the query mention, name-matches of name variations v, and sentence context respectively. Accordingly, model the factor function $\phi^{\rm me}(q,c)$ by the likelihood of a graphical model itself, represented by a log-linear function of potential functions $\phi^{\rm name}$ for name-similarity and $\phi^{\rm sent}$ for sentence context.

The resulting optimization criterion of the candidate answer c for the query model given the query q, V name variations v, S contextual phrases s, and M neighbor mentions m is given in Equation 5.

$$
\begin{aligned}
\log \mathcal{L}(c) &= \lambda^{\mathrm{Q}} \mathrm{log} \phi^{\mathrm{name}}(q,c) \\
&+ \lambda^{\mathrm{V}} \frac{1}{V} \sum_{v} \log \phi^{\mathrm{name}}(v,c) \\
&+ \lambda^{\mathrm{S}} \frac{1}{S} \sum_{s} \log \phi^{\mathrm{sent}}(s,c) \\
&+ \lambda^{\mathrm{M}} \frac{1}{M} \sum_{m} \Big(\rho_q(m) \log \phi^{\mathrm{me}'}(m,c) \Big)
\end{aligned} \tag{5}
$$

3.4 Joint Inference with Galago

Using log-linear models for factors ϕ with features that are readily available in the Indri and Galago[2] query languages, we can leverage the retrieval engine to optimize Equation 5. This is possible because the geometric interpolations with weights λ and ρ are expressed with the weighted `#combine` operator.

We rely on the sequential dependence model [14], which is a query model for models dependencies between adjacent query words. It strikes a balance between a model that ignores the order between the query terms (e.g. query likelihood model) and a model that requires the exact ordering of the words (n-gram or phrase model). For a given sequence of terms $t_1, t_2, \ldots, t_n$, the sequential dependence model scores documents by a graphical model combining term factors $\phi^{\mathrm{t}}(t_i)$, bi-gram factors $\phi^{\mathrm{O}}(t_i, t_{i+1})$, and factors capturing whether two subsequent query terms t_i and t_{i+1} occur within a window of 8 words.

We use simple factors in the remainder of the paper. For the name-match factor $\phi^{\mathrm{name}}(q,c)$ that tests all of the entity's indexed document for presence of the string representation of the query mention q. Because q consists of multiple terms, we use the sequential dependence model. We use a query likelihood operator for ϕ^{sent}. For the neighbor factor $\phi^{\mathrm{me}'}$ we also use the sequential dependence model.

With these feature functions, the optimization criterion of Equation 5 is equivalent to the Galago query given in Figure 2.

We restrict ourselves to a simple factor functions to demonstrate general applicability. Field-retrieval models provide further options, e.g., to let ϕ^{me} distinguish matches in different name fields (title, anchor text, etc); and $\phi^{\mathrm{me}'}$ distinguish matches in the full-text, the title of in-links and titles of out-links. The factor functions can make use of any feature that can be expressed in Galago query language, while still allowing to be optimized inside the retrieval engine.

4. NEIGHBORHOOD RELEVANCE MODEL

In the previous section we introduced a query model containing salience-weighted entity mentions m. We now discuss methods for estimating these salience weights $\rho_q(m)$ in an unsupervised manner. The idea is to assume a high salience of m for the query mention q, if both are frequently mentioned together. It is important to note that even unambiguous mentions are not necessarily useful for disambiguating other mentions.

```
#combine:0=$\lambda^{\mathrm{Q}}$:1=$\lambda^{\mathrm{V}}$:2=$\lambda^{\mathrm{S}}$:3=$\lambda^{\mathrm{M}}$(
  #seqdep($q$)
  #combine(#seqdep($v_0$) . . . #seqdep($v_V$))
  #combine(#seqdep($s_0$), . . . , #seqdep($s_S$))
  #combine:0 = $\rho(m_0)$ : . . . $k = \rho(m_k)$(
    #seqdep($m_0$), . . . , #seqdep($m_k$)
  )
)
```

Figure 2: Salience-weighted neighborhood query model in Galago syntax.

4.1 Local Document Neighborhood Model

As a first approach, salience of the neighboring mentions on the query mention can be estimated from the document surrounding the query. This technique was used by Gottipati and Jiang [5] to build a multinomial language model of entity mentions from the query document d_q with occurrence count n_{m,d_q}. We refer to this simple estimation technique as the local model.

$$
\rho_q^{\mathrm{local}}(m) = \frac{n_{m,d_q}}{\sum_{m'} n_{m',d_q}} \tag{6}
$$

Gottipati also tested weighting schemes that incorporates distance, but found that these did not significantly improve the results.

We suspect that especially when the query is not the main focus of the query document, many contextual entities are not relevant for disambiguation and may actually lead to worse performance (see experimental evaluation).

4.2 Across-document Neighborhood Relevance Model

We suggest the neighborhood relevance model which estimates entity saliences ρ from across-document evidence. The idea is that a neighbor is important if it occurs frequently in the context of the query mention within the document as well as across other documents that are topically related and contain mentions of the query mention q. Our method is based on pseudo-relevance feedback [13], which analyses results a first retrieval pass to expand the query with related words.

We first identify the query string q, with name variations v, and neighborhood m, and using the local document saliences ρ^{local}. We use the query model given in Equation 5 to search for documents d that contain coreferent mentions. We identify coreferent mentions by comparison to the name variations v and q—we call them pseudo-coreferent mentions.

Given a set of retrieved pseudo-relevant documents D, we use the retrieval probability of the document as an estimate of how likely the pseudo-coreferent mention refers to the same entity as the query q. As most retrieval frameworks return only unnormalized (rank-equivalent) retrieval scores $\mathcal{L}(d)$, the estimate has to be approximated with $\frac{\mathcal{L}(d)}{\sum_{d' \in D} \mathcal{L}(d')}$.

Counting the occurrence frequency $n_{m,d}$ of string-identical mentions of m, we build a multinomial language model across the pseudo-relevant documents with relevance-model weighting as follows.

[2] http://www.lemurproject.org/

$$\rho_q^{\text{nrm}}(m) = \frac{1}{\sum_{d' \in D} \mathcal{L}(d')} \sum_{d \in D} \frac{n_{m,d}}{\sum_{m'} n_{m',d}} \mathcal{L}(d) \qquad (7)$$

In other words, the salience of a mention m in the neighborhood is expressed by accumulating relative retrieval probabilities of documents according to how often they contain the mention.

Any corpus that contains documents with similar properties as the query document is a reasonable target corpus for the feedback query. In this work, we use the complete TAC source corpus, but other choices such as restrictions to news or blogs, depending on the query document are possible.

Typically, relevance feedback models are used to expand the query with new terms. This model is capable of introducing new entity mentions m that are not contained in the query document. However, since the context of the query document is already very rich, a preliminary experiment demonstrated that it is better to use the relevance model to reduce and weight the context found in the query document.

5. KB BRIDGE: ENTITY LINKING SYSTEM

In this section we describe KB Bridge, our retrieval-based entity linking system which is implemented using the Galago search engine and the MRF information retrieval framework. The system links entity mentions in the source document to knowledge base entities. The ranking of the entities is a two-stage process. First, entities are ranked using the Galago retrieval model described in Figure 2. The ranking is then refined with a supervised learning to rank model using RankLib[3]. The final step is NIL handling which determines if the mention is in the knowledge base or whether it is unknown.

5.1 Knowledge Base Representation

Our system addresses text-driven knowledge bases in which each entity is associated with free text, with relationships between entities from hyperlinks or other sources. Wikipedia is one representation of such a knowledge base, but our system would likely perform well on other knowledge bases.

In order to efficiently search over knowledge bases with millions or billions of entities we use an information retrieval system. For these experiments, we index the full text of the Wikipedia article, title, redirects, Freebase name variations, internal anchor text, and web anchor text.

5.2 Document Analysis

The first step in linking is to identify the entity query span q in the document and to find disambiguating contextual information for the query model introduced in Section 3. This includes name variations v, contextual sentences s, and other neighboring mentions m.

In the TAC KBP challenge, the entities of type person, organization, or location are the main focus of the linking effort and so the system detects entities using standard named entity recognition tools, including UMass's factorie[4] and Stanford CoreNLP [5]. These provide the mentions spans to derive query mentions q, name variations v, and neighboring entities m. Beyond the standard entity classes, our approach is general enough to also link other entity types if a suitable detector is incorporated.

Given a target entity mention, q, the system needs to identify name variations, v, in the document, such as "Steve" to "Steve Jobs" or "IOC" to "International Olympic Committee". The goal is to identify alternative names that are less ambiguous than the query mention. We use the within-document coreference tool from UMass's factorie, together with capitalized word sequences that contain the query string (ignoring capitalization and punctuation for the matching) to extract name variations v. From the set of coreferent mentions, we extract the sentences s they occur within. After removing stopwords, casing and punctuation they represent non-NER context such as verbs, adjectives, and multi-word phrases.

5.3 Cross-document evidence

Instead of analyzing one document in isolation, KB Bridge leverages topically similar entity mentions across documents. A full-text index of a corpus that contains similar documents is constructed. We then generate a query according to Figure 2, using local salience weights ρ^{local}, to retrieve documents from the collection. The documents are ranked by the likelihood of containing relevant entity mentions for original query mention, q. These documents are used in the neighborhood relevance model to identify salience weights, ρ^{nrm}.

5.4 KB Entity Ranking

The query model with salience weights from local document analysis and the neighborhood relevance model $\rho^{\text{nrm}}(m)$ is executed against the search index of KB entries as shown in Equation 5. Our system supports any feature function expressible in Galago query notation. Beyond this initial ranking, we can further refine the ranking using more complex features in learning to rank models.

The ranking is refined using supervised machine learning in a learning to rank (LTR) model. The refinement employs more extensive feature comparisons which would be expensive to compute over the entire collection. For these experiments we use the LambdaMART ranking model, a type of gradient boosted decision tree that is state-of-the-art and captures non-linear dependencies in the data. The model includes dozens of features. A description of the features used in the model is found in Table 1.

5.5 NIL Handling

After the entities are ranked, the last step is to determine if the top-ranked entity for a mention is correct and should be linked to the KB entry or instead refers to an entity not in the knowledge base, in which case NIL should be returned. For these experiments, we use a simple NIL handling strategy. We return NIL, if the supervised score of the top ranked entity is below a threshold τ. The NIL threshold τ is tuned on the training data. For the special case of the TAC KBP challenge, the reference knowledge base is a subset of Wikipedia. We exploit this fact by returning NIL whenever the top ranked Wikipedia entity is not contained in the reference knowledge base.

6. EXPERIMENTAL EVALUATION

6.1 Setup

We base our experimental evaluation on four data sets from the TAC KBP entity linking competition from 2009 to 2012.

[3] http://cs.umass.edu/~vdang/ranklib.html
[4] http://factorie.cs.umass.edu/
[5] http://nlp.stanford.edu/software/corenlp.shtml

Feature Set	Type	Description
Character Similarity	q, v	Lower-cased normalized string similarity: Exact match, prefix match, Dice, Jaccard, Levenstein, Jaro-Winkler
Token Similarity	q,v	Lower-cased normalized token similarity: Exact match, Dice, Jaccard
Acronym match	q	Tests if query is an acronym, if first letters match, and if KB entry name is a possible acroynm expansion
Field matches	q, v	Field counts and query likelihood probabilities for title, anchor text, redirects, alternative names fields
Link Probability	q, v	p (anchor | KB entry) - the fraction of internal and external total anchor strings targeting the entity
Inlink count	document prior	Log of the number of internal and external links to the target KB entry
Text Similarity	document	Normalized text similarity of document and KB entity: Cosine with TF-IDF, KL, JS, Jaccard token overlap
Neighborhood text similarity	document	Normalized neighborhood similarity: KL Divergence, Number of matches, match probability
Neighborhood link similarity	document	Neighborhood similarity with in/out links: KL divergence, Jensen-Shannon Divergence, Dice overlap, Jaccard
Rank features	retrieval	Raw retrieval log likelihood, Normalized posterior probability, 1/retrieval_rank
Context Rank Features	retrieval	retrieval scores for each contextual components: q, v, s, m_nrm, m_local

Table 1: Features of the query mention and candidate Wikipedia entity.

Over the years, the TAC organizers and the Linguistic Data Consortium came up with evaluation queries with varying characteristics both in terms of ambiguity (average unique mentions per entity) and variety (average number of entities per mention).

6.1.1 Data

The TAC KBP Knowledge Base was constructed from a dump of English Wikipedia from October 2008 containing 818,741 entries. The source collection includes over 1.2 million newswire documents, approximately 500 thousand web documents and hundreds of transcribed spoken documents. Across all years there are a total of 12,130 query entity mentions. We use all query mentions with odd numbered IDs as training data, and the even for evaluation. We inspected the distribution of the queries in the split to ensure they were representative for both NIL and queries with a ground truth entity ("in-KB") as well as the entity type distribution (Per/Org/GPE). The training set contains 6043 queries, 3034 with a ground truth entity c^* and 3009 NIL queries. The evaluation set contains 6087 queries with 3058 NIL and 3029 in-KB. This training set is used to learn parameters of our query model, as well as parameters of the supervised reranker. We learn across all years and evaluate year-by-year for comparison with previous results.

6.1.2 2012 Wikipedia dump

For evaluating a retrieval approach to linking, we use a more recent dump of Wikipedia. We use a Freebase dump of the English Wikipedia from January 2012, which contains over 3.8 million articles. It includes the full-text of the article along with metadata including redirects, disambiguation links, outgoing links, and anchor text. We also use the Google Cross-Wiki dictionary [18] for external link information from the web. We derive a mapping between our snapshot and the official TAC KBP knowledge base using title matches and article redirects. Using a more recent snapshot of Wikipedia is a common practice employed by the top performing linking systems in TAC. The full snapshot provides full text as well as rich category and link structure.

6.2 Context Modeling

We first evaluate the contributions of the different types of context for the entity query. The context includes the entity query string q, name variations v, the sentences s surrounding the query or name variations, as well as neighboring entity spans m. The combinations of these context components is indicated by Q, V, S, or M in the method prefix.

We evaluate three context weighting methods. The first is uniform weighting (QVM). Second is the local document model by Gottipati [5] (indicated by local). Third is our neighborhood relevance model (indicated by the suffix nrm). We compare both for estimating the salience $\rho(m)$ of neighboring entity mentions m. Baselines are the methods using only the query string (Q), the combination of query and name variations (QV), as well as the local context weighting (QVM_local). Our suggested methods are QVSM_nrm and QVM_nrm. These models are the full query model with neighborhood relevance weighting with and without sentences.

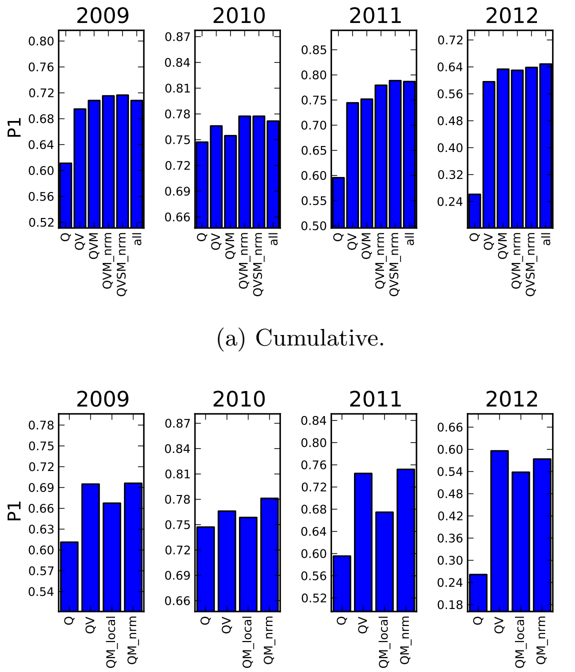
For each of the compared methods, we train separate λ parameters on the training data using a coordinate ascent learning algorithm. Estimated λ parameters differ across methods. For the QVSM_nrm model the estimated parameters are: $\lambda^{\mathrm{Q}} = 0.321$, $\lambda^{\mathrm{V}} = 0.293$, $\lambda^{\mathrm{S}} = 0.155$, and $\lambda^{\mathrm{M}} = 0.230$.

Figure 3 visualizes an ablation study for the context components using precision at rank one for evaluation. Figure 3a shows the cumulative improvements as context is added and weighted with the neighborhood relevance (QVM_nrm). QVM with uniform neighborhood weighting performs similarly to QVM_local weighting (not shown). We observe that adding sentence context does not significantly improve performance.

Figure 3b details the individual contributions of contextual components (omitting sentences). It is interesting that the QV method (entity name plus name variations) and QM_nrm (entity name plus weighted entity spans) are comparable in effectiveness. This is useful when no high quality name variations are extractable from the text, as is the case in informal text from social media. The cumulative figure above shows that when combined these features yield further improvement. Across all years the neighborhood relevance model achieves better effectiveness than the local model.

6.3 Ranking Distribution

In the previous section we examined the effectiveness of the different contextual components on the top-ranked result. Table 2 presents the contextual ranking models evaluated using mean reciprocal rank (MRR). Similar to the previous results, it shows that the most effective models include the neighborhood relevance weighting scheme (nrm). QVM_nrm and QVSM_nrm are significantly better than the QV baseline. The only exception is in 2010, when the queries are



(a) Cumulative.

(b) Individual Contributions.

Figure 3: Ablation study for the suggested method in terms of Precision @ 1.

Method	**2009**	**2010**	**2011**	**2012**
Q	0.702	0.824	0.698	0.385
QV	0.772	0.838	0.821	0.686
QM_nrm	0.773	**0.849***	0.825*	0.666
QM	0.746	0.829	0.758	0.636
QVM_nrm	**0.795***	0.845	0.849*	0.715*
QVM_local	0.784*	0.829	0.831	0.730*
QVS	0.771	0.834	0.822	0.697*
QVSM_nrm	0.792*	0.845	**0.850***	0.726*
QVSM_local	0.780*	0.836	0.837*	0.719*
all context	0.786*	0.841	0.848*	**0.735***

Table 2: Ranking results on TAC by year with varying context methods with mean reciprocal rank (MRR). The best results for each year are highlighted in bold. Results that are statistically significant with $\alpha = 5\%$ over the QV baseline are indicated with *.

Method	**2009**	**2010**	**2011**	**2012**
QVM_nrm	0.795	0.845	0.849	0.715
QVM_nrm LTR	0.913	0.936	0.918	0.805

Table 3: Learning to rank refinement results with mean recipocal rank (MRR). All LTR results are statistically significant with $\alpha = 5\%$ over the unsupervised QVM_nrm

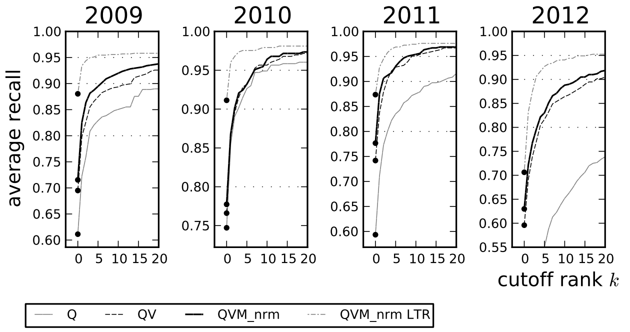


Figure 4: Average recall at rank cutoff k.

easier. In this case only the QM_nrm method is significantly better. Additionally, QVM_nrm is significantly better than the local weighting (Gottipati) for 2009-2011. However, there is no significant difference in 2012. We hypothesize that the reason the neighborhood model does not improve over the local model in 2012 is because the queries are significantly more ambiguous and the quality of the retrieved feedback documents is lower.

We refine the retrieval ranking using a supervised learning to rank model. The the features in the ranking model are described in Table 1. The top 100 results from the best ranking, QVM_nrm are reranked. The results of this are shown in Table 3. The results show significant improvement over the initial retrieval ranking leveraging more features that perform more extensive contextual comparison. The results for 2012 are still well below the other years, indicating the difficulty of these queries even leveraging the more complex contextual features. This indicates that a better feature representation is needed to address some of these difficult to resolve mentions.

6.3.1 *Recall*

The previous results use mean reciprocal rank to measure the retrieval effectiveness. We now examine the rank distribution in more detail, examining the recall at a given rank cutoff. The entity recall is critical because it is an upper bound on the effectiveness of downstream systems. To achieve a minimum 90% recall threshold across all years requires hundreds of candidates for the query (Q) model, 20 for QV, 16 for QVM_nrm, and only 3 for QVM_nrm LTR. The learning to rank model achieves at least 95% recall across all years within 10 results.

6.4 TAC KBP results

In this section, we evaluate the ranking as part of the entire linking pipeline described in Section 5. We report the micro-averaged accuracy because we do not focus on clustering NIL entity mentions. The results are in Table 4. The unsupervised retrieval QVM_nrm performs well, above the median in 2012 and competitive in previous years. The supervised ranking models improve effectiveness significantly. The in-KB ranking results outperform the best performing systems in 2009 through 2011 and are comparable in 2012. The main focus of this work is ranking, and this shows the effectiveness of our approach.

We now examine the overall (all) results, including the NIL handling. The results show that the QVM_nrm with LTR reranking and NIL handling outperforms the top system in 2009 and is competitive with the best performing systems in subsequent years. Applying the score threshold improves the overall accuracy despite decreasing in-KB effectiveness. This is because some correctly linked entities are marked as NIL, but are outweighed by the greater reduction in false

	2009			2010			2011			2012		
	in-KB	**NIL**	**all**	**in-KB**	**NIL**	**all**	**in-KB**	**NIL**	**all**	**in-KB**	**NIL**	**all**
QVM_nrm	0.810	0.703	0.764	0.768	0.764	0.766	0.766	0.767	0.766	0.584	0.623	0.605
QVM_nrm LTR	0.861	0.798	0.825	0.892	0.762	0.822	0.858	0.756	0.805	0.705	0.628	0.668
QVRM_nrm LTR NIL	0.847	0.848	0.847	0.883	0.843	0.862	0.833	0.857	0.845	0.676	0.758	0.714
Best Performer	0.765	-	0.822	0.823	-	0.864	0.801	-	0.870	0.687	-	0.721

Table 4: TAC Entity Linking performance in micro-average accuracy.

positive entity links. The NIL handling strategy based on thresholding the ranking score is effective, but could be improved further. Other linking systems use a supervised NIL classifier for this step, allowing them to perform well despite less effective in-KB ranking.

7. CONCLUSION

In this paper we propose an approach to entity linking based upon the Markov Random Field information retrieval model (MRF-IR). We focus on the task of ranking knowledge base entities. We demonstrate how concepts from joint neighborhood models can be incorporated within the MRF-IR framework. Further, we propose a neighborhood relevance model (NRM) that uses relevance feedback techniques to identify salient entity context across documents. Our experiments on the TAC KBP entity linking data show that the neighborhood relevance model outperforms other contextual models. When the ranking is refined with a learning to rank model the results beat the current best performing systems on in-KB ranking accuracy. Combined with a simple NIL handling strategy the overall effectiveness on all mentions is comparable to, and sometimes better than, other state-of-the-art entity linking systems.

Acknowledgements

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

8. REFERENCES

[1] Entity disambiguation for knowledge base population. In *COLING*, 2010.

[2] Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *European Chapter of the Association for Computational Linguistics (EACL)*, 2006.

[3] S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. *EMNLP*, 2007.

[4] S. Cucerzan. Tac entity linking by performing full-document entity extraction and disambiguation. *Proceedings of the Text Analysis Conference*, 2011.

[5] Swapna Gottipati and Jing Jiang. Linking entities to a knowledge base with query expansion. In *EMNLP*, 2011.

[6] Johannes Hoffart, Mohamed A. Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *EMNLP*, 2011.

[7] Darren W. Huang, Yue Xu, Andrew Trotman, and Shlomo Geva. Focused access to XML documents. chapter Overview of INEX 2007 Link the Wiki Track. 2008.

[8] Heng Ji and Ralph Grishman. Knowledge base population: successful approaches and challenges. In *ACL-HLT*, 2011.

[9] Heng Ji, Ralph Grishman, and Hoa Dang. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers*, 2011.

[10] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[11] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, 2009.

[12] Paul McNamee, Veselin Stoyanov, James Mayfield, Tim Finin, Tim Oates, Tan Xu, Douglas W. Oard, and Dawn Lawrie. HLTCOE participation at TAC 2012: Entity linking and cold start knowledge base construction. In *TAC KBP*, 2012.

[13] D. Metzler and W.B. Croft. Latent concept expansion using markov random fields. In *SIGIR*, 2007.

[14] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In *SIGIR*, 2005.

[15] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung. Cross-lingual cross-document coreference with entity linking. *Proceedings of the Text Analysis Conference*, 2011.

[16] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, 2011.

[17] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 14, pages 313–323. 1971.

[18] Valentin I. Spitkovsky and Angel X. Chang. A Cross-Lingual dictionary for english wikipedia concepts. In *LREC*, 2012.

[19] Veselin Stoyanov, James Mayfield, Tan Xu, Douglas W. Oard, Dawn Lawrie, Tim Oates, and Tim Finin. A context-aware approach to entity linking. In *AKBC-WEKEX*, 2012.

[20] J. Xu and W.B. Croft. Query expansion using local and global document analysis. In *SIGIR*, 1996.