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 mentions in documents to entities in a knowledge base. One of the crucial tasks is to identify the disambiguating context of the mention, and joint assignment models leverage the relationships within the knowledge base. We demonstrate how joint assignment models can be approximated with information retrieval. We build on pseudo-relevance feedback and use the source corpus to build an neighborhood relevance model that we show is more effective than local models for ranking KB entities. Our results demonstrate that simple text based features combined with a supervised Learning to Rank model result a model that matches or outperforms the top performing system on in-KB accuracy in the TAC KBP entity linking task.

1. INTRODUCTION

Entity linking is important because most information on the web 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 by linking mentions of entities in free text to Wikipedia-like knowledge bases, in which entities are inter-linked and further are associated with free text. Entity links enable navigation between documents and entities, and to related documents by the induced link structure. Entity linking is a fundamental building block that supports a wide variety of extraction, summarization, and data mining tasks. For example, starting with an entity, the links to documents where it is mentioned can be used to identify sources for extracting relevant facts, such as a person's name, who they are married to, or where they work.

The major challenge in entity linking is uncertainty. 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:

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

A typical entity linking system consists of four steps: 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., [17, 4, 18], leverage contextual entities as disambiguating evidence in step 3. The downside is that the candidate set of step 2) is acquired by a pipeline of heuristics, resulting in arbitrary large candidate sets on the order of hundreds of entities for ambiguous matches, where the consequences of the interplay between step 2 and step 3 are not well aligned.

We advocate an information retrieval approach that uses one probabilistic model to approach steps 1-3. State-of-the-art entity linking methods only employ IR to a minor degree, where this work pushes the boundaries to maximize the use of IR methods. The formalism of graphical models allows us to ground our work on models from both Information Extraction and Information Retrieval.

For a given query, a good candidate entity fulfills three properties: The names match, textual context of the query is contained the article text, and named entities surrounding the query are reflected in entities neighboring the answer entity. Further notions of contextual similarity can be included, but throughout the paper we focus on name variants, phrases, and neighboring entities to model the user intent for the query. Starting with an underlying graphical model, corresponding query model, query analysis and indexing component are derived.

Entity linking provides some unusual challenges. The typical IR setting addresses short queries by using relevance models to add more terms to the query model. In entity linking, the query is embedded in a document, providing an abundance of context which could be included. 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 American Broadcasting Companies, Inc. The named entity span "Australia" is not relevant for the true answer. It might actually misguide the process to link to the wrong entity "Australian Broadcasting

Corporation Television".

We introduce the neighborhood relevance model to estimate the salience of context with the goal of filtering and weighting (as opposed to expanding) the query model. The neighborhood relevance model is based on ideas of pseudo-relevance feedback and latent concept expansion to leverage collocation evidence across other similar documents in the corpus. 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 25 entities.
- Query-specific approach for identifying salient neighboring entities using external and 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.

2. BACKGROUND AND RELATED WORK

2.1 Related Work

Early work on entity linking was done 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 text based ranking features and do not use Wikipedia-specific features such as category hierarchies, disambiguation pages, and extracted concepts.

Our work is related to that of Gottipati and Jiang [8] who apply a language modeling approach to entity linking. They expand the original mention query with contextual information from the language model of the query 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 [10]. 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. [12, 11] provide an overview of the recent systems and approaches. This includes state-of-the-art in approaches to entity linking, such as an updated Cucerzan system [4] which consistently performs at or near the top of the rankings.

Instead of linking individual mentions one at a time, recent work [3, 18, 21, 13, 9] focuses on linking the set of mentions, M, that occur in the local document d. These models perform collective (or joint) inference over the mentions in document to identify a coherent assignment of KB entries to mentions. In our work we leverage the set of mentions M in the document as context in an information retrieval model. One important way that our work differs is that we focus on identify salient entity mentions in the context, because mentions in the document may be spurious or only tangentially related if the document contains multiple topics.

As the trend for joint resolution of entities within documents and clustering NIL entries increases, a relevant related task is cross-document coreference resolution, where the goal is to determine whether two mentions of an entity refer to the same instance across all documents. Work in this field was done by Bagga and Baldwin [1] and Gooi and Allan [7] who used entity language models built from the context the entities occur in to disambiguate ambiguous entity mentions. In this work we do not focus on clustering all mentions in text across documents, instead we focus on linking mentions of entities in a single document using cross-document evidence to a knowledge base.

2.2 Graphical Models and Factor Graphs

Graphical models provide the mathematical framework for formalizing intuition on how available data and requested quantities are connected. Casting data and quantities of interest as random variables $\vec{X} = X_1, X_2, \ldots X_n$, dependencies between two (or more) variables are encoded by factor functions ϕ that assign a non-negative score to each joint configuration $\vec{x}$ of their variables. The configuration of all variables is scored by the likelihood function, which is represented by the normalized product over all factors.

This paper makes use of undirected graphical models, which are also called Markov Random Fields. In these models, the likelihood $\mathcal{L}(\vec{x})$ of the configuration $\vec{x}$ has an alternative log-linear representation over the cliques $\vec{y}$ of variables X in the graphical model, $\mathcal{L}(\vec{x}) = \frac{1}{Z}\prod_{\vec{y}\subset\vec{x}} \exp\{\langle\theta_{\vec{y}}, \mathrm{f}(\vec{y})\rangle\}$. Here $\langle .,. \rangle$ represents the inner product of f, feature vector of the clique, and its parameter vector $\theta_{\vec{y}}$. Z refers to the marginal probability over all possible configurations to ensure that $\mathcal{L}$ satisfies the laws of a probability. The parameter vector can be either hand set or learned from training data with discriminative optimization methods.

Graphical models can be nested, in which case the likelihood function $\mathcal{L}$ of the inner graphical model can be used as a factor ϕ or a weighted feature f in the outer model. The model can be used to derive predictions $\vec{x}^*$ by maximizing the likelihood or equivalently the log likelihood as $\vec{x}^* = \arg\max_{\vec{x}} \log \mathcal{L}(\vec{x})$.

2.3 Graphical Models in IR: Sequential Dependence Model

Markov Random Fields are widely used in information retrieval and most unigram, n-gram, and term dependence models can be expressed as a graphical model, involving random variables for the query terms $q_1, q_2, \ldots q_n$ and a document d. Each document is then scored according to the likelihood function $\mathcal{L}(d, q_1, q_2, \ldots q_n)$.

The sequential dependence model is a retrieval model for a multi word query. Each of the query terms $q_1, q_2, \ldots q_n$ is cast as a random variable together with a document d. The model makes use of three classes of factors: term, bigram, and windowed bigram, where factors of the same class are sharing the same parameter θ. The model includes a term factor $\phi^{\mathrm{T}}(q_i, d)$ between each query term and the document variable. For each pair of adjacent query terms q_i, q_{i+1} it includes a bigram factor $\phi^{\mathrm{O}}(q_i, q_{i+1}, d)$ and an windowed bigram factor $\phi^{\mathrm{U}}(q_i, q_{i+1}, d)$. Each document d is then scored according to

$$\log \mathcal{L}(d, q_1, \ldots, q_n) \propto \sum_i \log \phi^{\mathrm{T}}(q_i, d) + \sum_i \log \phi^{\mathrm{O}}(q_i, q_{i+1}, d) + \sum_i \log \phi^{\mathrm{U}}(q_i, q_{i+1}, d)$$

The graphical model paradigm allows each of the factors ϕ to arise from feature vectors f of arbitrary length. However, the original work of Metzler at al. [16] uses only a single scalar feature per factor so that the inner products default to a scalar multiplication. Factor ϕ^{t} is induced by the feature that represents the Dirichlet-smoothed log-probability $p(q_i|d)$ of the single term q_i in d. Given the Dirichlet smoothing parameter μ and document term frequency $n_{q_i,d}$, document length $n_{\cdot,d}$, and collection term frequency $n_{q_i,\cdot}$ and collection size $n_{(\cdot,\cdot)}$ it is given by $p(q_i,d) = \log \frac{n_{q_i,d}+\mu\frac{n_{q_i,\cdot}}{n_{(\cdot,\cdot)}}}{n_{\cdot,d}+\mu}$.

Further it uses the score of ordered bigrams for $\log \phi^{\mathrm{O}}(q_i,q_{i+1},d) = \theta^{\mathrm{O}} \cdot \#1(q_i,q_{i+1})$ and unordered bigrams within a window of eight terms for $\log \phi^{\mathrm{U}}(q_i,q_{i+1},d) = \theta^{\mathrm{U}} \cdot \#\mathrm{uw8}(q_i,q_{i+1})$.

We are going to use the open source retrieval engine Galago,[1] which is part of the Lemur project. Galago comes with the implementation of the sequential dependence model above, acessible via the operator `#seqdep(`$q_1,\ldots,q_n$`)`.

Galago further allows to nest retrieval models via the `#combine` operator to score documents geometric mean interpolation. For models $\mathcal{M}_i$ and interpolation weights λ_i, the likelihood function $\mathcal{L}(d) = \prod_i \mathcal{M}_i(d)^{\lambda_i}$ is accessible via syntax `#combine:1=`λ_1`:2=`λ_2`:...:n=`λ_n`(`$\mathcal{M}_1\ \mathcal{M}_2\ \ldots\ \mathcal{M}_n$`)`.

2.4 Graphical Models in IE: Candidate-based Neighborhood Model

Markov random fields are equally popular in the information extraction community. Early approaches to entity linking [5], use a graphical model with single factor $\phi^{\mathrm{me}}(q,c)$. Each of the candidates c are scored by $\log \mathcal{L}(c,q) \propto \langle \theta, \mathrm{f}(q,c)\rangle$ where the feature vector includes a variety of similarity functions between the query string, and the article's title, redirect, anchor text, as well as TF-IDF weighted cosine similiarity between terms in the query document and the Wikipedia article. The parameter θ is trained discriminatively with a learning to rank approach.

Ratinov [19] extended this basic model by explicitly incorporating contextual entity mentions m, each with a respective set of candidates z. The idea is that entities which are mentioned in the same documents are also likely to be linked on Wikipedia. Therefore, if each contextual mention m is linked to its correct candidate z^*, the links between KB entries z^* and the candidates entries c for the query will reveal the true answer c^*. This intuition is modeled in the likelihood function of Equation 1, which requires two compatibility measures: One compatibility measure between mentions in the text to KB entries ϕ^{me}, as well as a compatibility measure among KB entries ϕ^{ee}.

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

As the task is to link only the query mention, the contextual entity links are marginalized out by integration over z. The dilemma is that linking m's to z's requires to solve the entity linking problem as part of the solution. Therefore the problem has to be addressed by joint inference which in this case does not have a closed-form solution, and therefore require approximate inference.

3. QUERY MODEL

In this section we close the gap between the graphical models for entity linking as developed in IE community and the graphical models for information retrieval.

One shortcoming of Ratinov's model is that it requires generated candidate sets for the query and contextual mentions which, with current methods, is not only time consuming, but can also result in very large sets of candidates z that need to be integrated over in Equation 1.

Another issue is that not all contextual mentions m are equally relevant for the query, as we argued above, some are spurious or misleading. We address both issues in the following.

3.1 Neighborhood Query Model

We demonstrate how the retrieval engine can be used to optimize Equation 1 whenever factor functions ϕ^{me} and ϕ^{ee} can be expressed as query operators. The consequence is a tight integration of the candidate generation (step 2, in the pipeline) with the entity ranking (step 3), optimizing over all possible candidates on Wikipedia at once.

The key insight is to solve the integral over z (cf. Equation 2), with smart preprocessing and indexing: The Wikipedia snapshot is transformed so that the article of entity c is enriched with information about the contextual entities m and their KB counterparts z. The factor $\phi^{\mathrm{me}'}(m,c)$ can therefore be directly optimized within the retrieval model framework.

$$\phi^{\mathrm{me}'}(m,c) = \int \phi^{\mathrm{me}}(m,z) \cdot \phi^{\mathrm{ee}}(z,c)\, dz \qquad (2)$$

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

With the introduction of the factor $\phi^{\mathrm{me}'}$, Equation 1 is rewritten as Equation 3.

3.2 Relevance-weighted Neighborhood Query Model

As pointed out before, not all contextual entities are equally relevant. For each contextual entity m the 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 $\phi^{\mathrm{me}'}(m,c)$ on the likelihood function. Based on the geometric mean, which is the natural choice for probabilities, we achieve the weighting with the geometric interpolated model of Equation 4.

$$\mathcal{L}(c) = \phi^{\mathrm{me}}(q,c) \cdot \prod_m \Big(\phi^{\mathrm{me}'}(m,c)\Big)^{\rho_q(m)} \qquad (4)$$

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

We want to further introduce parameters λ^{Q} and λ^{M} that allow the trade-off between the direct similarity of the query and candidate as expressed by $\phi^{\mathrm{me}}(q,c)$ and the aggregated influence of the 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 5.

$$\log \mathcal{L}(c) = \lambda^{\mathrm{Q}} \log \phi^{\mathrm{me}}(q,c) + \lambda^{\mathrm{M}} \sum_m \Big(\rho_q(m) \log \phi^{\mathrm{me}'}(m,c)\Big) \qquad (5)$$

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

3.3 Extended Context in Query Model

We further study variations on the relevance-weighted neighborhood query model given in Equation 5.

Name variances 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. However, the name variance extraction may be less reliable, which is expressed in an additional trade-off parameter λ^{V}.

We also incorporate non-entity context in the form of surface phrases of the sentence that surround the query mention or one of the mentioned name variances. The sentence context s is balanced with the parameter λ^{S}, and the compatibility with the candidate answer c is expressed in the factor $\phi^{\mathrm{se}}(s, c)$.

The resulting optimization criterion of the candidate answer c for the query model given the query q, V name variants v, S contextual phrases s, and M contextual entity mentions m is given in Equation 6.

$$\begin{aligned}\log \mathcal{L}(c) &= \lambda^{\mathrm{Q}} \log \phi^{\mathrm{me}}(q, c) \\ &+ \lambda^{\mathrm{V}} \frac{1}{V} \sum_{v} \log \phi^{\mathrm{me}}(v, c) \\ &+ \lambda^{\mathrm{S}} \frac{1}{S} \sum_{s} \log \phi^{\mathrm{se}}(s, c) \\ &+ \lambda^{\mathrm{M}} \frac{1}{M} \sum_{m} \Big(\rho_q(m) \log \phi^{\mathrm{me}'}(m, c) \Big)\end{aligned} \tag{6}$$

3.4 Joint Inference with Galago

Using log-linear models for factors ϕ with features that are readily available in the Indri and Galago query languages, we can leverage the retrieval engine to optimize Equation 6. This is possible because the weighted sums with weights λ and ρ are expressed with the `#combine` operator (cf. Section 2.3), with submodels that express $\phi(x) = \exp\{\langle \theta, \mathrm{f}(x) \rangle\}$. Inspecting the inner product $\langle \theta, \mathrm{f}(x) \rangle = \sum_i \theta_i \mathrm{f}_i(x)$ reveals another `#combine` operator with weights θ_i acting on the features $\mathrm{f}_i(x)$. If all features $\mathrm{f}_i(x)$ can be expressed in the Galago query language, Equation 6 can be directly optimized inside the search engine.

For instance we could use a feature vector for $\mathrm{f}^{\mathrm{me}}(q, c)$ that separates scores of q in the Wikipedia title field, redirect field, and anchor text field. For feature vector $\mathrm{f}^{\mathrm{me}'}(m, c)$ we could separate scores of m in the article full text from the titles of in- and outlinks. But for simplicity we use feature vectors with only a single entry, the sequential dependence model score, for all factors $\mathrm{f}^{\mathrm{se}}(s, c)$, $\mathrm{f}^{\mathrm{me}}(q, c)$, $\mathrm{f}^{\mathrm{me}'}(m, c)$.

With these feature functions, the optimization criterion of Equation 6 is equivalent to the following Galago query.

$$\begin{aligned}&\#\text{combine:}0{=}\lambda^{\mathrm{Q}}\text{:}1{=}\lambda^{\mathrm{V}}\text{:}2{=}\lambda^{\mathrm{S}}\text{:}3{=}\lambda^{\mathrm{M}}(\\ &\quad \#\text{seqdep}(q) \\ &\quad \#\text{combine}(\#\text{seqdep}(v_0) \ldots \#\text{seqdep}(v_V)) \\ &\quad \#\text{combine}(\#\text{seqdep}(s_0), \ldots, \#\text{seqdep}(s_S)) \\ &\quad \#\text{combine:}0 = \rho(m_0) : \ldots \; k : \rho(m_0)(\\ &\qquad \#\text{seqdep}(m_0), \ldots, \#\text{seqdep}(m_k) \\ &\quad) \\ &)\end{aligned} \tag{7}$$

4. NEIGHBORHOOD RELEVANCE MODEL

In the previous section we introduced a query model containing relevance weighted entity mentions m. We now discuss methods for estimating these weights $\rho_q(m)$ in an unsupervised manner. As previously mentioned, even unambigious mentions are not necessarily useful for disambiguation. We introduce a model for determining the importance of these neighborhood salience weights $\rho(m)$ using pseudo-relevance feedback [15]. 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.

The approach is based on the assumption that these pseudo-relevant documents mention the same target entity. In other words, the goal is to identify documents containing pseudo-coreferent mentions. We use these pseudo-document mentions to determine the strength of association between entities in the neighborhood. If a neighboring mention in the query document is not relevant, it will only be contained in few or none of the pseudo-relevant documents. If it represents salient disambiguation context, it is assumed to occur in many documents of the retrieved set.

4.1 Local Document Neighborhood Model

We can estimate beliefs about the salience of the entity neighborhood from the source document. This technique was used by Gottipati and Jiang [8] 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.

$$\rho_q^{\mathrm{local}}(m) = \frac{n_{m,d_q}}{\sum_{m'} n_{m',d_q}} \tag{8}$$

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

We find that whenever 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 evidences. Having identified the query string q, with name variants v, and neighborhood m, and using the local document saliences ρ^{local}, we search for coreferent mentions in pseudo-relevant documents—we call them pseudo-coreferent mentions.

We use the query model given in Equation 6 to retrieve pseudo-relevant documents d from the source corpus. $\mathcal{L}(d)$, the retrieval score under the query model, represents its relevance to the query q. Given a set of pseudo-relevant documents D, we can approximate the document relevance probability with $\frac{\mathcal{L}(d)}{\sum_{d' \in D} \mathcal{L}(d')}$. In combination with a multinomial language model, based on occurrence counts $n_{m,d}$ of mentions in the pseudo-relevant documents d, the neighborhood relevance model estimates salience weights $\rho_q^{\mathrm{nrm}}(m)$ as follows.

$$\rho_q^{\text{nrm}}(m) = \frac{1}{\sum_{d' \in D} \mathcal{L}(d')} \sum_{d \in D} \frac{n_{m,d}}{\sum_{m'} n_{m',d}} \mathcal{L}(d) \qquad (9)$$

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.

Typically relevance 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 information retrieval based entity linking system which is implemented using the Galago search engine and the MRF-IR retrieval framework. The linking system links mentions in the query 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 Equation 7. We then optionally, in the second stage, the ranking is refined with a supervised learning to rank model e.g. RankLib[2]. 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, where links between entities are extracted from hyperlinks directly or via relation extraction systems. Wikipedia is one representation of such a knowledge base, but our system works as well on other knowledge bases with full text data.

In order to efficiently search over very large knowledge bases containing millions or even billions of entries we use a full-text retrieval system. For the knowledge base experiments we describe below, we index the full text of Wikipedia article, the title, redirects, Freebase name variations, and internal anchor text, web anchor text. The combination of both text and structured metadata in the search index allows the execution of rich contextual query models. We can further utilize field indices to efficiently incorporate complex feature vectors in Section **??**.

For estimating the neighborhood relevance model, we index a larger unstructured corpus, preferrably with similar characteristics as the query documents. This allows the system to estimate the neighborhood relevance model weights ρ using topically similar documents from pseudo relevance feedback.

5.2 Document Analysis

The first step in linking is to identify the query span q and to find disambiguating contextual information for the query model introduced in Section **??**: name variants v, contextual sentences s, and and other entity spans m in the neighborhood.

If entities of type person, organization, or location are the main focus of the linking effort, Named Entity Recognition tools, such as from UMass's factorie [14] and Stanford CoreNLP [6] provide useful spans to derive query mentions q, name variants v, and neighboring entities m. The KB Bridge system is not limited to entities of these types, it can link any kind of KB entity, as long as a corresponding span detector is available.

For the name variants, v, the system identifies similar spans within the document that are likely to be coreferent, 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 variants v. All remaining spans are used as the neighborhood m.

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 KB Entity Ranking

Next, the information from document analysis is used to build a query model given in Equations 6 and 7 to rank the entries in the knowledge base. Our system supports all feature vectors for which entries are expressable in Galago query notation. To demonstrate generality for cases where rich meta data is absent, we use a feature representation where every factor is associated with a single feature, which represents the score of the sequential dependence model. We use this query model both for the relevance model and for retrieving KB entities.

To estimate the neighbor relevance model, we retrieve documents from the background corpus, using local salience weights ρ^{local}. From the retrieved document set D, the neighborhood relevance saliences ρ^{nrm} are estimated using Equation 9.

Finally, the query model with updated salience weights $\rho^{\text{nrm}}(m)$ is executed against the knowledge base to retrieve KB entities c which are optimal according to Equation 6.

5.4 Learning to Rank KB entries

To further improve the ranking, we leverage supervised machine learning in a learning to rank (LTR) system which re-ranks the retrieved set of KB entities. This ranking can employ more expensive textual feature comparison which are infeasible to score on the entire collection. For ranking, the system uses the LambdaMART model, a type of gradient boosted decision tree that is state-of-the-art in ranking and captures non-linear dependencies in the data. There are hundreds of features used in the reranking step. A description of the sets of features used in the model is found in Table 1. We note that the features used in the model are text-based and do not utilize the type information of NER types and Wikipedia categories.

5.5 NIL Handling

The NIL handling component determines if the top ranked KB entity for a mention should be linked or if there is no match in the knowledge base in which case NIL should be returned. 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. Queries for which NIL is returned should be clustered, however it has been shown that

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

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.

simple heuristics achieve strong results. We assign NILs to the same cluster whenever their top ranked candidates are the same, otherwise they are kept in their own singleton cluster.

For the special case of an TAC KBP entity linking system, we notice that the reference knowledge base is a subset of the full Wikipedia. We exploit this fact by further 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 - 2012. 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 newsire documents, approximately 500 thousand web documents and hundreds of transcribed spoken documents. Across all years there are a total of 12,130 queries. We use all queries with odd numbered query IDs as training data, and the even queries for evaluation. We inspected the distribution of the queries in the split and the NIL to in-KB as well as the type distribution (Per/Org/GPE) of the results are preserved. 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 re-ranker. For evaluation, we use even numbered query IDs on a year-by-year basis.

6.1.2 Linking corpus: 2012 Wikipedia

For evaluating a large-scale text retrieval approach to linking, we use a more recent dump of Wikipedia that includes the full-text of the article along with other structured metadata including redirects, disambiguation links, outgoing links, anchor text, and full-text. We use a Freebase dump of the English Wikipedia from January 2012, which contains over 3.8 million articles including link and full text information. In addition, we use the Google Cross-Wiki dictionary[20] for external web link information. We derive a mapping between our snapshot and the official TAC KBP knowledge base using title matches and article redirects.

6.2 Methods

We first evaluate which kinds of context are mostly benefitial to be integrated into Equation 6: the query q, the name variants v, the sentences s surrounding the query or name variants, as well as the neighborhood context m. Which subset is included is indicated by Q, V, S, or M in the method prefix.

For methods that include neighborhood context, we study different estimation methods for the salience $\rho(m)$ of each neighbor m. This includes the local document model by Gottipati [8] (indicated by local), and our neighborhood relevance model (indicated by the suffix nrm).

We use the method based on the query string (Q), and the combination of query and name variants (QV), as well as the context weighting local context (QVM_local) as baselines.

Our suggested methods are QVSM_prf and QVM_prf, the full query model with neighborhood relevance weighting with and without sentences.

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

6.3 Context Entity Ranking

We first perform an intrinsic evaluation of the ranking methods on the in-KB queries. We measure the ranking effectiveness in identifying the correct KB entity. In this experiment we study the effectiveness of different combinations of context for disambiguation. Table 2 presents the ranking results in terms of the mean reciprocal rank metric (MRR). It demonstrates that the all best methods include the neighborhood relevance weighting scheme (nrm), and the suggested methods QVM_nrm and QVSM_nrm are significantly better than the QV baseline. The only exception is in 2010, when the queries are easier only the QM_nrm method is statistically significant. The method QVM_nrm method is significantly better than the weighting from Gottipati (QVM_local) local weighting for the years 2009, 2010, and 2011, with no significant difference in 2012. We hypoth-

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 *.

esize that this is because the queries in 2012 are significantly more difficult and therefore the quality of the feedback documents is likely significantly lower.

Figures 1 visualizes the ablation study for the suggested method. Figure 1a shows the cummulative improvements as context is added (QVM) and weighted with neighborhood relevance (QVM_nrm). We observe that adding sentences context does not significantly improve performance, and we do not report the sentence context results in subsequent experiments.

Figure 1b details the individual contributions of context components. Comparing the method which adds name variants QV versus the method which incorporates weighted context (QM_nrm) without name variants, reveals that the context is equally helpful as name variants expansion, if weighted with the neighborhood relevance model. This is especially useful when no high quality name variants are extractable from the text, as is the case in informal text from social media.

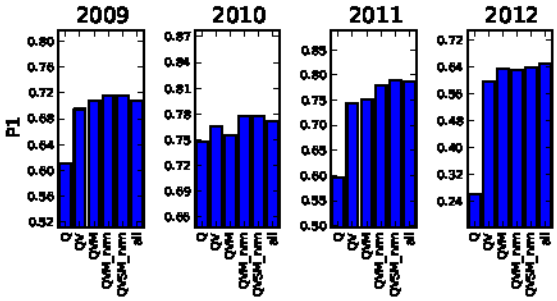
6.4 Learning to Rank

Once candidates are ranked, we can further refine the unsupervised retrieval ranking using supervised learning to rank. We use the features describe previously in Table 1 and apply the learning to rank model to our best current model, QVM_nrm. The results of this are shown in Table 3. The results show that significant improvement over the unsupervised baseline is possible using more expensive text features combined with supervised learning. While all years improve, the results for 2012 are still well below the other years, indicating that the difficulty of these queries.

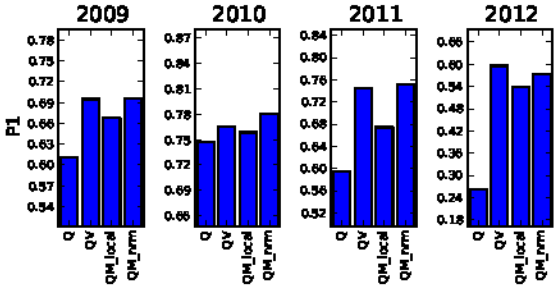
The distribution of the correct answers is important. For linking, only getting the top answer correct or not is considered. However, down stream systems using the entity linking information could benefit from getting the correct answer in a small number of results. Figure 2 visualizes the distribution of the answer in the ranked results. The results show that all the methods return 90% of entities in the top 10 results. After learning to rank refinement is performed 95% recall is achieved using only the top 5 results.

6.5 TAC KBP results

In previous sections we focused on the ranking of in-KB entities only. In this section, we evaluate the ranking as part of the entire linking pipeline described in Section ??.



(a) Cummulative.



(b) Individual Contributions.

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

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

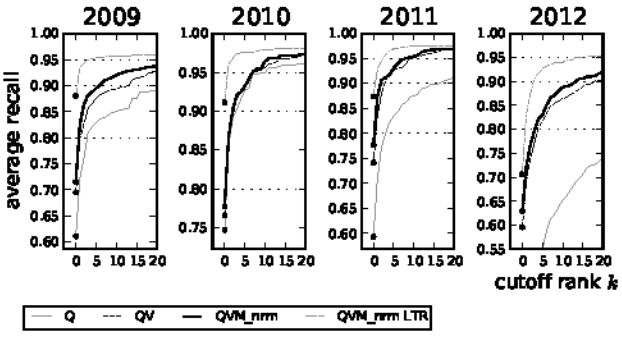


Figure 2: Recall at rank cutoff k.

For this experiment we use the micro-averaged accuracy because we do not focus on NIL clustering. The results are reported in Table 4. We observe that the unsupervised retrieval QVM_nrm performs well, above the median 2012 and competitive in previous years. The supervised models improve effectiveness. When the NIL score threshold is applied we observe that the in-KB effectiveness decreases, but the overall accuracy numbers increase because of a greater decrease in false positive links. The results show that the

	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-avg accuracy.

QVM_nrm with LTR reranking and the nil threshold applied outperform the top system in 2009 and are competitive with the best performing system in subsequent years.

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. The information retrieval system uses only text-based features without exploiting knowledge from Wikipedia. We demonstrated how joint neighborhood models can be expressed within the MRF-IR framework. Further, we proposed a neighborhood relevance model (NRM) that uses relevance feedback to identify salient entity mentions in the context of the query document. Our experiments on the TAC KBP entity linking data show that the neighborhood relevance model outperforms or is en par with other contextual models. We also demonstrated that a learning to rank model using text based features outperforms the current best performing systems on in-KB ranking, and even a very simple NIL handling strategy results in overall numbers that are comparable to the 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] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Computational Linguistics*, 1998.

[2] Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *European Chapter of the Association for Computational Linguistics (EACL-06)*, 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] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *International Conference on Computational Linguistics*, 2010.

[6] J.R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005.

[7] Chung H. Gooi and James Allan. Cross-Document coreference on a large scale corpus. In Daniel and Salim Roukos, editors, *HLT-NAACL*, 2004.

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

[9] 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.

[10] Darren W. Huang, Yue Xu, Andrew Trotman, and Shlomo Geva. Focused access to XML documents. chapter Overview of INEX 2007 Link the Wiki Track, pages 373–387. Springer-Verlag, 2008.

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

[12] Heng Ji, Ralph Grishman, and Hoa Dang. Overview of the TAC2011 knowledge base population track. In *Text Analysis Conference*, 2011.

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

[14] Andrew McCallum, Karl Schultz, and Sameer Singh. Factorie: Probabilistic programming via imperatively defined factor graphs. In *NIPS*, 2009.

[15] D. Metzler and W.B. Croft. 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, 2007.

[16] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 472–479, New York, NY, USA, 2005. ACM.

[17] 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.

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

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

[20] Valentin I. Spitkovsky and Angel X. Chang. A Cross-Lingual dictionary for english wikipedia concepts. In *Conference on Language Resources and Evaluation*, 2012.

[21] 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.