Exploring Re-ranking Approaches for Joint Named-Entity Recognition and Linking

Avirup Sil
Temple University
Computer and Information Sciences
Philadelphia, PA
avi@temple.edu

ABSTRACT

Recognizing names and linking them to structured data is a fundamental task in text analysis. Existing approaches typically perform these two steps using a pipeline architecture: they use a Named-Entity Recognition (NER) system to find the boundaries of mentions in text, and an Entity Linking (EL) system to connect the mentions to entries in structured or semi-structured repositories like Wikipedia. However, the two tasks are tightly coupled, and each type of system can benefit significantly from the kind of information provided by the other. In this proposal, we present a joint model for NER and EL, called NEREL, that takes a large set of candidate mentions from typical NER systems and a large set of candidate entity links from EL systems, and ranks the candidate mention-entity pairs together to make joint predictions. In our initial NER and EL experiments across three datasets, NEREL significantly outperforms or comes close to the performance of two state-of-the-art NER systems, and it outperforms 6 competing EL systems. On the benchmark MSNBC dataset, NEREL provides a 60% reduction in error over the next-best NER system and a 68% reduction in error over the next-best EL system.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text analysis; I.3.1 [**Content Analysis and Indexing**]: Linguistic processing

Keywords

Named Entity Recognition, Entity Linking, Entity Disambiguation

1. INTRODUCTION

Entity Linking (EL) [1, 4, 25, 25, 24, 8] is the task of identifying when a name that appears in text refers to a known entity in a reference set of named entities, such as a relational database or the set of articles in Wikipedia. EL is a common first step in information management and text processing tasks like information extraction [14] and entity search [2, 16]. It is also an important first step for semantic understanding of text, such as in grounded semantic parsing [13].

Existing EL systems depend on knowing where the set of names in a text appear. Nearly all EL systems therefore use a pipeline architecture: they run a Named-Entity Recognition (NER) [7, 23] system on the text to identify the boundaries of names. They then use the names that were identified by the NER system, typically called *mentions*, as the input set of names for the EL task. (Many NER systems provide additional information besides the boundaries of the mentions, such as a type label like Person, Location, or Organization. Since our EL system will link mentions to a relational database with a more extensive and fine-grained ontology, we ignore this extra output of the NER system, and consider just the mention boundaries.) Unfortunately, as is common with pipeline architectures for probabilistic models, errors from the NER system propagate to the EL system. Any mention that goes undetected by the NER system clearly cannot be linked correctly by the downstream EL system. Other common errors from NER systems include incorrectly splitting a multi-word name into multiple mentions; joining together two mentions that happen to appear near one another; or leaving off one or more words from a multi-word mention. Each of these errors can cause the downstream EL system to generate too many, too few, or just wrong links.

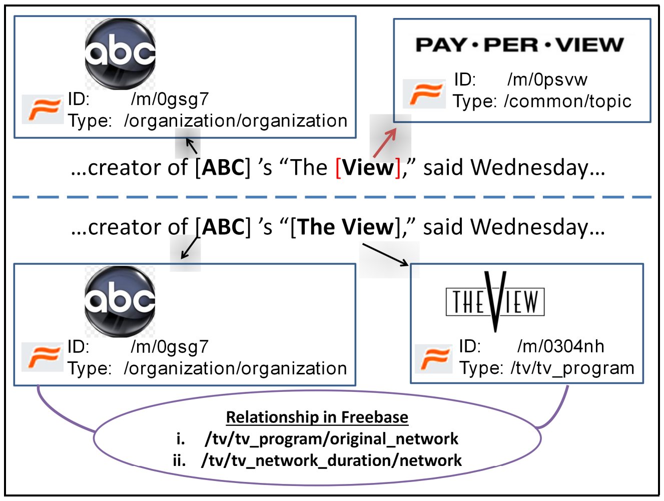
abc
ID: /m/0gsg7
Type: /organization/organization
PAY·PER·VIEW
ID: /m/0psvw
Type: /common/topic
…creator of [ABC] 's "The [View]," said Wednesday…
…creator of [ABC] 's "[The View]," said Wednesday…
abc
ID: /m/0gsg7
Type: /organization/organization
THE VIEW
ID: /m/0304nh
Type: /tv/tv_program
Relationship in Freebase
i. /tv/tv_program/original_network
ii. /tv/tv_network_duration/network

Figure 1: Above: Incorrect mention boundaries and entity link (for "View") from a state-of-the-art NER system and our pipeline EL system. Below: Correct output from our joint model, together with some of the additional relational information from Freebase that helps the joint model recover the correct mention boundaries and entity links.

Figure 1 shows an example phrase that has been incorrectly processed by a pipeline of a state-of-the-art NER system [23] and our own baseline EL system that links to the Freebase database[1]. The word "the" commonly precedes certain mentions, as in "the [Rocky Mountains]," and so it is reasonable for the NER system to place the mention boundaries around just "View" in this example. Likewise, it is reasonable for an EL system to predict that "View" refers to

[1]www.freebase.com

"Pay-Per-View," a common method of watching certain television programs, since most of the surrounding text is related to television programming.

We propose (and develop) a joint NER and EL model that we use to re-rank candidate mentions and entity links produced by base NER and EL models. Our base NER techniques for example, would produce candidate mentions like ABC, View, The View, and Wednesday from the above phrase. Our base EL system would then propose several promising entity links for each candidate mention, including Pay-Per-View and The View (an American television program on a broadcast channel called ABC). The reranking model then chooses among the set of all possible mention and entity link labelings for the whole phrase to determine the best choice. Since the reranking model considers labels for ABC and (The) View simultaneously, it can use features for known relationships between the television channel ABC and the television program The View to encourage these as outputs. For efficiency, such features are not available to the pipeline models, which consider only one mention at a time. We use the pipeline models to prune the set of all possible candidate mentions and entity links to a manageable size while maintaining high recall. The reranking model can then use more sophisticated features for collective classification over this pruned set.

Previous work [4, 21, 10, 12] has considered collective classification for entities, but never for the combination of NER and EL. Ratinov *et al.* [24] argue that collective classification for entity linking provides only a small benefit over a purely-local model. However, we find that by combining NER and EL, the set of correct mentions that are available to be linked improves drastically, and EL performance improves as a result.

The remainder of this paper is organized as follows: The next section discusses related work. Section 3 describes the Freebase database, and why it is suited to the EL task. Section 4 presents our joint model for NER and EL. Section 5 describes our methods for finding candidate mentions and entities, including a pipeline EL system for Freebase. Section 6 gives our experimental results, and Section 7 concludes.

2. RELATED WORK

All but one previous entity linking system that we are aware of uses a pipeline architecture. Ratinov *et al.* [24] point out that finding mentions using NER is a non-trivial task, and hence their experiments report linking performance using gold-standard mentions, as do several other recent EL systems [25, 5, 9]. Nevertheless, in practice NER performance limits the performance of these EL systems. Our work relies on pipeline NER and EL techniques [1, 20, 30, 6, 24, 15, 14, 17, 18, 11, 19, 25] to provide high-quality candidate mentions and entity links. We then subject these candidates to further processing by a joint NER and EL model that outperforms state-of-the-art NER systems and state-of-the-art EL systems on their respective tasks.

Guo *et al.* [8] are the only authors we are aware of that consider joint mention detection and entity disambiguation, but their techniques were developed for, and are best-suited for, microblog text. Because microblog texts are so short, Guo *et al.* can use computationally expensive machinery; they use a structural SVM algorithm [26, 28], which requires NP-hard inference. Their technique can consider nearly every token as part of a candidate mention, again because of the short length of microblog texts. In contrast, our techniques are better suited for longer documents. We use linear maximum-entropy models to re-rank a set of candidate mentions and entities provided by efficient NER and EL base models. A more minor difference is that Guo *et al.* link to Wikipedia; our technique links to both Wikipedia and Freebase—a large, user-contributed, relational database. Also, Guo *et al.*'s techniques cannot identify mention boundaries that have no corresponding Wikipedia entries, whereas our techniques can identify mentions with no corresponding entity in our reference set; we follow the Text Analysis Conference's (TAC) guidelines in linking such mentions to the special symbol NIL.

3. ENTITY LINKING WITH FREEBASE

Most existing large-scale entity linkers use Wikipedia titles as a reference set. We chose to use a combination of Freebase and Wikipedia as our reference set because it provides us all of the information in the Wikipedia articles, as well as a well-developed ontology of types and binary relations available in Freebase. Freebase is a freely-available, user-contributed repository of structured data containing almost 40 million unique entities. In comparison, Wikipedia contains 4.2 million entities and DBpedia [2] contains 3.77 million entities (all numbers are for the English versions). The Freebase schema is divided into *domains*, each of which contains its own set of *types* (such as `people`, `books`, `bridges`, *etc.*) and *properties*, like `date of birth` for a person or `latitude` and `longitude` for a location. Freebase commons, which we use in our experiments, contains 86 different domains, including such diverse areas as schools, geography, sports, and astronomy. Every entity in Freebase is given a unique *ID*, and separately a set of possible names that can refer to the entity, through the built-in `name` and `alias` relations. Many Freebase entities also have links to their corresponding Wikipedia pages, when available. Furthermore, Freebase includes a built-in, nontrivial API for finding candidate entity links for a given name, which is based on the user-contributed `name` and `alias` relations.

4. JOINT NAMED-ENTITY RECOGNITION AND ENTITY LINKING

4.1 Examples

While EL systems can go wrong even when provided correct mentions, we have found that in practice a large number of EL errors are caused by poor mention boundaries. Table 1 shows examples of three common errors by NER systems that propagate to EL systems. Other types of errors include completely-missed mentions, and correct mentions but incorrect links. Joint NER and EL can help with many of these problems by allowing for features that, for instance, favor joining two mentions that link to the same entity over having the same entity repeated twice.

4.2 Problem Formulation

Labeled data for joint NER and EL consists of a set of documents D, and for each document $d \in D$, a set $Entities(d)$ of triples of the form (l, r, e). Here, l and r indicate the left and right boundaries of the mention of some named entity, and e indicates the unique identifier of the named entity in a reference set. Given a labeled training set T of this form, the task is to learn a model that can predict $Entities(d)$ for new documents d.

4.3 Decomposition into Connected Components

Classification over all possible sets of (mention, entity)-pairs for a document is generally intractable. Typical approaches to EL have addressed this by decomposing the task into a pipeline: first, identify the set of mentions using a standard NER model. Second, determine a local model for $P(E|d, l, r)$, where E is a random variable over entity ids e. Many recent EL approaches have investigated a third step, which is to select up to K tuples of entity ids $(e_1, \ldots, e_n)$, one e_i for each mention in the document found by the NER model, and then build a model to rank these tuples of entity ids [1, 4]. However, Ratinov *et al.* [24] argue that this kind of global model provides a relatively small improvement over the purely-local approach, and Sil *et al.* [25] dispense with the global classification step entirely.

We instead adopt a different approach to decomposing the task, which offers a way to jointly classify mention boundaries and entity links. Rather than assume that the first-stage NER model produces the correct mention boundaries for linking, we assume that

[2]http://dbpedia.org/About

Error Type	Bad Example	Correct Example
Under-segmentation	[Orange County, Calif.]→`Orange County`	[Orange County]→`Orange County` [Calif.]→`California`
Over-segmentation	[Blue Cross]→`BCBS of Alabama` and [Blue Shield of Alabama]→`BCBS of Alabama`	[Blue Cross and Blue Shield of Alabama]→ `BCBS of Alabama`
Mis-segmentation	[Florida Supreme]→`(FL Supreme Court)` Court	[Florida Supreme Court]→`FL Supreme Court`

Table 1: Examples of three kinds of NER errors that cause a pipeline EL system to miss links or add extra links. Mention boundaries are shown in [square brackets], and entity links are indicated by →.

it provides a high-recall set of potentially-overlapping candidate mentions. One way to produce this set is to develop a probabilistic NER model $P(M|d, l, r)$, where M is a binary random variable indicating whether l and r are the exact boundaries of a mention. By selecting a relatively low threshold for this distribution, the model can produce a high-recall set of candidate mentions for d that have some reasonable probability of being correct. In practice, we have found that an alternative approach of getting candidate mentions (described in Sec. 5.1), which uses an NP chunker and an existing NER system, to be efficient and effective. Let A be the set of all sequences $a_i = (m_1, \ldots, m_{n_i})$ of mentions m such that each m is taken from the candidate set, and no two mentions in the same sequence have overlapping boundaries. As in the pipeline model, we also develop a local model $P(E|d, l, r)$, and select several promising candidates from the local model for each candidate mention. The joint classification task is over all tuples B of the form $b_i = (m_1, e_1, \ldots, m_{n_i}, e_{n_i})$, where each e_j is taken from the local EL model's candidate set for m_j. We call these *entity-mention tuples*.

The cost of high recall in the set of candidate mentions is that the set of candidate mentions is large, and as a result B contains far too many candidate tuples for tractable classification. To decompose the problem, we observe that most words in (non-microblog) text are not part of named-entities. We adopt a simple heuristic of separating all candidate mentions that are three or more words apart. In all three of the corpora in our experiments, this simple heuristic results in a partitioning of the mentions into small sets of mentions that appear near one another. We refer to these sets as the *connected components* of d, or $CC(d)$. We perform classification over the set of entity-mention tuples $B(cc)$ that are formed using candidate mentions and entities within the same connected component $cc \in CC(d)$. As an example, the phrase "ABC's The View" from Figure 1 would constitute a connected component, since several candidate mentions (ABC and View) are separated by three or fewer tokens. Two of the entity-mention tuples for this connected component would be ([ABC], `ABC`, [View], `Pay-Per-View`) and ([ABC], `ABC`, [The View], `The View`).

In over 98% of our training documents, all of the connected components contained tuples b_i with $n_i \leq 10$ mentions. The few remaining large connected components were from documents describing the results of cricket matches, where long lists of player names were separated only by one or two statistics each. For such long sequences, we heuristically divide them into chunks of a length that is the maximum length that our implementation can handle, which is 15. With relatively low thresholds for selecting candidate mentions and entities, the number of tuples in each connected component in our training data was 31.02 on average, while still allowing a maximum possible recall of 96%. In contrast, the maximum possible recall for a pipeline EL system using a standard NER system in our datasets is 82%. The joint model adds an extra layer of processing on top of the pipeline model, but in experiments the re-ranking step has nearly identical running time as the combination of our pipeline NER and EL models. Overall, this simple decomposition leads to a tractable joint NER and EL classification task.

4.4 Re-ranking Model

We use a maximum-entropy model to estimate $P(b|d, cc)$, the probability of an entity-mention tuple b for a given connected component $cc \in CC(d)$. The model involves a vector of real-valued feature functions $\mathbf{f}(b, d, cc)$ and a vector of real weights $\mathbf{w}$, one weight per feature function. The probability is given by

$$P(b|d, cc, \mathbf{w}) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(b, d, cc))}{\sum_{b' \in B(cc)} \exp(\mathbf{w} \cdot \mathbf{f}(b', d, cc))} \tag{1}$$

We use L2-regularized conditional log likelihood (CLL) as the objective function for training:

$$CLL(T, \mathbf{w}) = \sum_{(b,d,cc) \in T} \log P(b|d, cc, \mathbf{w}) + \sigma \|\mathbf{w}\|_2^2$$

where $(b, d, cc) \in T$ indicates that b is the correct tuple of entities and mentions for connected component cc in document d in training set T, and σ is a regularization parameter. Note that this training objective treats each cc equally, even though some will contain more mentions and entities than others. In contrast, we use standard experimental evaluations (precision and recall) which will treat each mention and entity equally. We experimented with modified versions of the training objective to account for this difference, but these had little empirical impact, possibly because there was a relatively small variance of 1.02 in the number of mentions per b tuple.

We used LBFG-S for gradient-based convex optimization. The gradient for our objective function is given by

$$\frac{\partial CLL}{\partial w_i} = 2\sigma w_i + \sum_{(b,d,cc) \in T} f_i(b, d, cc) - \sum_{\substack{d \in T \\ cc \in CC(d) \\ b' \in B(cc)}} P(b'|d, cc, \mathbf{w}) f_i(b', d, cc)$$

The training algorithm is guaranteed to converge to the globally optimal parameter setting for our objective function over the training data.

4.5 Features

The primary advantage of the discriminative re-ranking approach is that it provides the flexibility to include many novel features into NER and EL. For instance, we include features that penalize NER boundaries that oversegment a mention by counting how often an entity-mention tuple b links to the same entity multiple times in a row. We also include features that help to learn and encourage common sequences of entity types, like (job title, employee name) and (city, country or state). These features help both to prevent undersegmentation in NER and to improve linking accuracy. Details of our features are below.

Token-level features: Given b, d, and cc, we define features that find the minimum (**MIN**), maximum (**MAX**) and the average (**AVG**) number of tokens inside the mentions in b, and a **NUM** feature for the total number of mentions in b. We also use features that measure how many tokens are not part of any mention in b, but are part of a mention in some other $b' \in B(cc)$. Let $Non\_Mention\_Tokens(b, cc)$

be the set of all such tokens, and $All\_Tokens(cc)$ be the set of all tokens for cc. We construct the **NON-MENTION-COUNT** feature as $|Non\_Mention\_Tokens(b, cc)|$, and the **NON-MENTION-RATIO** feature as $\frac{|Non\_Mention\_Tokens(b,cc)|}{|All\_Tokens(cc)|}$.

Capitalization: Similar to [8], we include one binary feature to check if all candidate mentions in b are capitalized, one binary feature to check if all words in all mentions in b are capitalized, and one binary feature to check if there is any mention that has capitalized, non-mention tokens to its immediate left or right.

Features from entity links: We include a number of features that make use of the entity links in b, as well as the types and known relations between entities, as stored in Freebase. The most basic versions of these features include: **COUNT-NIL**, which counts the number of entities that link to NIL, to help penalize tuples b that are poor matches with Freebase; **COUNT-EXACT-MATCH**, which counts the number of mentions whose surface form matches exactly with one of the names for the linked entity stored in Freebase; **ALL-EXACT-MATCH**, which is true if all mentions in b match a Freebase name exactly; and **ACRONYM-MATCH**, if the mention's surface form is an acronym for a name or alias of the linked entity in Freebase.

The **DISTINCT-LINKS-PER-MENTION** measures the ratio of distinct entities in b to mentions in b. This feature helps to penalize over-segmented phrases. This is because two mentions that should have been joined together often have high probability for links to the same entity, as in a bracketing of "[Home] [Depot]" that splits this name for a business into two parts, each of which has a high probability of linking to the company `Home Depot`. Typically, any value of less than 1 for this feature is an indication that b is oversegmented, as it is uncommon to refer to the same entity twice in the short span of text that makes up one of our connected components, although appositives are an obvious exception.

To further establish the relation between mention candidates and their links, we define two features that mimic the Normalized Google Distance (NGD) as in [8, 24, 3]. For every consecutive pair of entities (e, e') that belongs to mentions in b, the **NGD-FREEBASE** feature computes $\sum_{(e,e')\in b} \frac{|types(e) \cap types(e')|}{|types(e) \cup types(e')|}$. The Wikipedia version, **NGD-WIKIPEDIA**, computes $\sum_{(e,e')\in b} \frac{|Wiki(e) \cap Wiki(e')|}{|Wiki(e) \cup Wiki(e')|}$, where $Wiki(e)$ denotes the set of tokens in the Wikipedia page of e, excluding stop-words.

Two features, **ENTITY-TYPE-PMI** and **ENTITY-TYPE-PRODUCT-PMI**, make use of Freebase's type system to find patterns of entities that commonly appear next to one another. Let $T(e)$ be the set of Freebase types for entity e. We remove common Freebase types which are associated with almost every entity in text, like `/location/location` or `/people/person`, since they have lower discriminating power. From the training data, the system first computes pointwise mutual information (PMI) [29] scores for the Freebase types of consecutive pairs of entities, (e_1, e_2):

$$PMI(T(e_1), T(e_2)) = \frac{\sum_{(e,e')\in T} \mathbf{1}[T(e_1) = T(e) \wedge T(e_2) = T(e')]}{\sum_{e\in T} \mathbf{1}[T(e_1) = T(e)] \times \sum_{e\in T} \mathbf{1}[T(e_2) = T(e)]}$$

where the sum in the numerator is taken over consecutive pairs of entities (e, e') in training data. The feature **ENTITY-TYPE-PMI** adds these scores up for every consecutive (e_1, e_2) pair in b. The feature **ENTITY-TYPE-PRODUCT-PMI** does the same, but uses this alternative variant of the PMI score:

$$ProductPMI(T(e_1), T(e_2)) = \prod_{i,j} \frac{\sum_{(e,e')\in T} \mathbf{1}[T_i(e_1) \in T(e) \wedge T_j(e_2) \in T(e')]}{\sum_{e\in T} \mathbf{1}[T_i(e_1) \in T(e)] \times \sum_{e\in T} \mathbf{1}[T_j(e_2) \in T(e)]}$$

The **BINARY-RELATION-COUNT** feature encourages entity tuples b where the entities have a known relationship with one another. For instance, for the text "Home Depot CEO Robert Nardelli", we wish to encourage the tuple that links "Home Depot" to the entity id of the company `Home Depot`, and "Robert Nardelli" to the entity id of the person `Nardelli`. Freebase contains a relation called `organization_board_memberships` that indicates that `Nardelli` is a board member of `Home Depot`. **BINARY-RELATION-COUNT** counts the number of such relations between every consecutive pair of entities in b. This feature helps not just to improve entity linking accuracy; it also helps to prevent under-segmented phrases. For instance, an entity-mention tuple b that contains just a single mention for the whole phrase "Home Depot CEO Robert Nardelli" would have a lower count for this feature than the correctly-segmented version, even if the incorrect mention was linked to `Nardelli`.

Finally, we include a similar version of this feature that uses Wikipedia rather than Freebase. The **WIKIPEDIA-COOCCURRENCE-COUNT** feature computes, for every pair of consecutive entities $(e, e') \in b$, the number of times that e' appears in the Wikipedia page for e, and *vice versa*. It adds these counts up to get a single number for b.

5. BASE NER AND EL MODELS

5.1 Overgenerating candidate mentions

Previous work on EL systems [24, 8] have argued that EL systems often suffer because of the mention detection phase. The impact is mostly because of false negative mentions: any missed mention by the NER system is also a missed entity for EL. To improve on the pipeline model, our joint system requires *overgeneration* of candidate mentions: that is, a high-recall set of mentions with just enough precision to prevent the number of false positives from overwhelming subsequent processing.

Rather than modify an existing NER system to overgenerate mentions, we adopt an approach of combining multiple existing systems and heuristic techniques. We use the mentions found by the state-of-the-art UIUC NER system[3] In addition, we included strings matching the following heuristic tests as candidate mentions:

- All Noun-phrase (NP) chunks from the Illinois shallow parser [22]. We also retain all sub-expressions of up to 7 tokens of the NP chunks.
- All possible n-grams (for n $\leq$ 8) which match names and aliases of entities in Freebase.
- For any string selected by any of the above techniques, we add in one new mention that includes the word to the left, and another new mention that includes the word to the right.

5.2 A Base Entity Linker using Freebase

Let mention $m = (d, l, r)$, and let S be a reference set containing entity ids and the special symbol NIL, which indicates an entity that does not match any id in the reference set. The task of a base (or local) EL model is to predict $P(E|d, l, r)$, where random variable E ranges over S. We call our base EL model BASEEL.

[3]We used the best settings possible, which meant using both CoNLL training and development data as the training set.

BASEEL, like our re-ranking model, is a maximum entropy model. Using Freebase and Wikipedia, we construct a vector of real-valued feature functions $\mathbf{f}(m, s)$ describing the degree to which m and $s \in S$ appear to match one another. The feature functions are described in the following subsections. The model includes a vector of real weights $\mathbf{w}$, one weight per feature function.

$$P(s|m, \mathbf{w}) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(m, s))}{\sum_{s' \in S} \exp(\mathbf{w} \cdot \mathbf{f}(m, s'))} \tag{2}$$

As with the joint model, we use L2-regularized CLL as the objective function, and LBFG-S for gradient-based convex optimization. We use only gold-standard mentions for training the base EL model. Our manually annotated data contains only positive examples for entity ids. To generate negative examples, we use Freebase's candidate generation technique (described in Section 3) to identify a set of potentially confusing bad matches. Let $\phi_{FB}(m)$ denote the set of candidate entities that Freebase returns for mention m, plus NIL and the correct entity s if s is not already in $\phi_{FB}(m)$. Using this set as the complete set of possible labels for m, the gradient for our objective function is given by

$$\frac{\partial CLL}{\partial w_i} = \sum_{(m,s) \in T} f_i(m, s) - \sum_{s' \in \phi_{FB}(m)} P(s'|m, \mathbf{w}) f_i(m, s') + 2\sigma w_i$$

5.2.1 *Freebase Features*

BASEEL uses a combination of information from Wikipedia and Freebase to judge the similarity between a mention m and Freebase entity s. We first describe how it extracts information from Freebase. Table 2 lists all of the Freebase features used in BASEEL. As a starting point, we leverage the four domain-independent features described in our initial work in Sil *et al.* [25], adapted to Freebase. Intuitively, these features measure the number of attribute values and attribute names for entity s in Freebase that appear in the context of m. We split the attribute counts into four different kinds, the **COUNT ALL** through **COUNT LABEL** features in Table 2. As an example, consider the entity `Timberlake`, a performing artist. This entity appears in Freebase relations called `film`, `band member`, and `date of birth`, among many others. Within these relations are tuples that contain `Timberlake`, as well as a mixture of other entity IDs and numeric values. For instance, in the `band member` relation is a tuple containing

$$t = (\texttt{Timberlake}, \texttt{'N Sync}, 1995, 2002, \texttt{Tenor})$$

We collect the names of all entity IDs like `'N Sync` that appear in such tuples into a set $Att_{name}(\texttt{Timberlake})$, and numeric values like 1995 are collected into a set $Att_{num}(\texttt{Timberlake})$. Freebase also contains attribute names for the attributes in each relation; for example, "band," "member," and "period (start)" are the names of three attributes in the `band member` relation. The names of the attributes for relations that `Timberlake` appears in are gathered into the set $Att_{label}(\texttt{Timberlake})$.

Consider this snippet of Web text:

> Last Saturday night we witnessed N'Sync band member Timberlake performing with Britney Spears for the first time.

For the mention of "Timberlake," the **COUNT ALL** and **COUNT UNIQUE** features would have a value of 1 each, because the phrase "N Sync" in the text matches one of the names in $Att_{name}(\texttt{Timberlake})$. The **COUNT LABEL** feature would have a value of 2, since "band" and "member" in the Web text both match entries in $Att_{label}(\texttt{Timberlake})$. **COUNT NUMERIC** would have a value of zero.

Initial experimental results indicate that a linear classifier over these four simple features, adapted from Sil *et al.* (our previous work) to use on Freebase, already perform better than two state-of-the-art Wikifiers. However, Freebase has more information to exploit, beyond what these four features measure. We begin by observing that a fairly common error during training is to assign a high probability to an incorrect entity that has a very high **COUNT ALL** feature, but very low (or zero) counts for other features. Similarly, in some cases the model assigns a high probability to an incorrect entity with a high **COUNT LABEL** feature, but low values for other features. On the other hand, correct entities tend to have positive and somewhat balanced values for all four of these features. For example, a document describing a Justin Timberlake concert in London mentions "London" several times, and the singer's band and a well-known song once. The incorrect entity `Timberlake Wertenbaker`, a British playwright who resides in London, has a high value for **COUNT ALL**, but low values for all other features. The singer has a small but positive value for **COUNT ALL**, **COUNT UNIQUE**, and **COUNT LABEL**. In response, we introduce an **ENTROPY** feature to measure the balance of the feature values across Sil *et al.*'s features. Entities which have relatively similar values for all four features will have a high value for the **ENTROPY** feature, and a low value if the original four features are unbalanced.

Our final kind of Freebase feature takes advantage of its type system. Every entity in Freebase is associated with one or more types, and the types are structured in a hierarchical taxonomy. For example, the performing artist entity `Timberlake` comes with the types: `Music_Artist`, `Music_Group_Member`, `TV_Actor`, `Film_Actor`, and others. Likewise, `Britney Spears` is a `Music_Artist`, `TV_Actor`, and `Lyricist`. `Timberlake, Ohio` has the types `Town` and `Statistical_Region`. As the text in the snippet of Web text above suggests, seeing "Britney Spears" in the context of "Timberlake" is an indicator that "Timberlake" refers to the singer, rather than the town. In this example, `Britney Spears` and `Timberlake` share two types, while `Britney Spears` and `Timberlake, Ohio` share none.

Our **TF** feature counts, for each type t that an entity s belongs to, the number of mentions in the context of m that have any candidate entities with the same type. The feature then adds up these counts for all the types of s. We also compute variations of this feature that compute the average and max score over all of the different types of s, rather than the sum. Note that for the sake of efficiency, we make no attempt at collective classification in BASEEL (*c.f.* [24]); the **TF** feature for "Timberlake" does not depend on the predicted entity for "Britney Spears", but rather on the set of all candidate entities for "Britney Spears."

Analogously to the well-known Inverse Document Frequency heuristic, we define a separate weighting scheme for measuring the importance of a given type for determining the similarity of s to the context of m. For instance, `Timberlake` also belongs to the type `Person`. However, this type is far too inclusive and common to be of much use for discriminating between entities in most circumstances. On the other hand, the type `Music_Artist` is much less common in text, so the fact that `Timberlake` shares this type with `Britney Spears` should be more informative than the fact that these two entities share the type `Person`. We first compute a Document Frequency score $DF(t)$ for all types t in Freebase by counting the number of documents in our training data that had some candidate entity belonging to that type. Let $\#D$ denote the total number of candidate entities in the training data. We then compute a **TF-IDF** feature by reweighting the type frequency for type t by

$$IDF(t) = \log \frac{\#D}{DF(t)}$$

As with the TF feature, we compute three variations of the feature that sum, average, and max over all types t of a given entity s.

5.2.2 *Wikipedia Features*

For many Freebase entities s, there exists a link to the corresponding Wikipedia page, which we denote by $W(s)$. Hence, we have access to unstructured text which we can use to compute useful

Freebase Feature Functions	
COUNT ALL:	$\sum_{n \in Att_{name}(s)} \text{Count}(m,n)$
COUNT UNIQUE:	$\sum_{n \in Att_{name}(s)} \mathbf{1}[\text{Count}(m,n) > 0]$
COUNT NUM:	$\sum_{n \in Att_{num}(s)} \text{Count}(m,n)$
COUNT LABEL:	$\sum_{n \in Att_{label}(s)} \text{Count}(m,n)$
ENTROPY:	$\sum_{i \in \{\text{Sil } et\ al. \text{ feature indices}\}} -f_i(m,s) \log f_i(m,s)$
TF-SUM:	$\sum_{t \in \text{T}(s)} \sum_{s' \in \text{CE}(m)} \mathbf{1}[t \in T(s')]$
TF-MAX:	$\max_{t \in \text{T}(s)} \sum_{s' \in \text{CE}(m)} \mathbf{1}[t \in T(s')]$
TF-AVG:	$\text{avg}_{t \in \text{T}(s)} \sum_{s' \in \text{CE}(m)} \mathbf{1}[t \in T(s')]$
TF-IDF-SUM:	$\sum_{t \in \text{T}(s)} IDF(t) \left(\sum_{s' \in \text{CE}(m)} \mathbf{1}[t \in T(s')] \right)$
TF-IDF-MAX:	$\max_{t \in \text{T}(s)} IDF(t) \left(\sum_{s' \in \text{CE}(m)} \mathbf{1}[t \in T(s')] \right)$
TF-IDF-AVG:	$\text{avg}_{t \in \text{T}(s)} IDF(t) \left(\sum_{s' \in \text{CE}(m)} \mathbf{1}[t \in T(s')] \right)$

Table 2: BASEEL's feature functions for Freebase. Count(m, n) denotes the number of times that string n appears in the document containing mention m. T(s) denotes the set of Freebase types for entity s. CE(m) indicates the set of all candidate Freebase entities for all other mentions in the document containing m.

Wikipedia Feature Functions
cosine similarity($Text(W(s))$,$Text(m)$)
cosine similarity($Text(W(s))$,$Context(m)$)
cosine similarity($Context(W(s))$,$Text(m)$)
cosine similarity($Context(W(s))$,$Context(m)$)
$\forall i \in \{1, .., 5\}$. $Text(m)_{tw_i}$
$\forall i \in \{1, .., 5\}$. $Context(m)_{tw_i}$

Table 3: BASEEL's Wikipedia feature functions.

additional features that can help our linking algorithm decide if a target concept best describes the input mention. Table 3 describes these features. Our Wikipedia features are simplified versions of the features used in the local model of Ratinov *et al.* [24]. Ratinov *et al.* also consider a more complex "global" model that performs collective classification, with a small improvement in performance over the local model. As mentioned previously, we forego this type of collective classification, although combining combining joint NER-EL and collective EL is an important task to consider for future work.

The basic intuition behind these Wikipedia features is that a mention m appearing in document d is more likely to refer to entity s described on Wikipedia page $W(s)$ if $W(s)$ has high textual similarity to d. For each Wikipedia page p, we construct a vector space model of p, which we denote as $Text(p)$. We also create a more localized vector space model of the entity for page p by creating a vector space model from the 100 closest words to the first occurrence of a mention of the entity on p, or the first 100 words of p if we cannot identify a mention of the entity. We call this vector space model $Context(p)$. Similarly, we create vector space models $Text(m)$ and $Context(m)$. We then use cosine similarity over these vector space models as features.

In addition to the features inspired by Ratinov *et al.*, we extracted the top 5 most frequently occurring words (excluding stop-words) from each Wikipedia target page p, which we refer to as tw_1 through tw_5. For each tw_i, we create two features, one that counts how often tw_i occurs in the document containing m, and one that counts how often tw_i occurs in the 100 word window surrounding m.

domain	$\|M\|$	$\mathbf{E}\|\phi_{FB}(m)\|$	NIL	In Freebase
ACE	257	43.65	0%	100%
MSNBC	747	28.15	10%	89%
CoNLL	5616	27.85	20%	79%
Wikipedia	158715	12.62	0%	100%

Table 4: Number of mentions, average number of candidate referents, % of mentions that are NIL, and % of mentions that are in Freebase (and Wikipedia) in our datasets. We train our system on the Wikipedia dataset.

6. INITIAL EXPERIMENTS

We evaluate the performance of NEREL first on the NER task against 2 state-of-the-art NER systems, and second on the EL task against 6 state-of-the-art EL systems.

Datasets: We use 3 standard datasets for EL as our test sets: the ACE dataset, as annotated by Ratinov *et al.* [24] for Wikipedia links; the MSNBC dataset, as collected and annotated by Cucerzan [4] for mention boundaries and Wikipedia links; and the CoNLL-2003 NER dataset (testb) [27], with YAGO2 and Freebase IDs added by Hoffart *et al.* [11]. Table 4 provides key statistics on these datasets. For datasets annotated with Wikipedia links, we automatically extracted the corresponding Freebase IDs using Freebase's built-in links to Wikipedia. In less than 1% of the examples for all the 3 test sets, Wikipedia contained a correct entity, but Freebase did not. We left the Wikipedia page as the correct label for these examples, and our BASEEL and NEREL systems always gets these examples wrong. In 10% of the MSNBC examples, Wikipedia had no entry, and the examples were marked NIL (ACE had no NIL entries). We left these NIL cases even when Freebase does contain a correct entity, to provide a fairer comparison with wikifiers. The MSNBC data consists of news articles covering 10 domains, 2 articles from each domain: Business, Entertainment, Politics, Science, etc, and is thus more diverse than either the CoNLL or ACE datasets. The ACE dataset includes hand-labeled entity links for only the first nominal mention of each annotated coreference chain, and there are many correct mentions and entities which have not been labeled. This gold standard allows us to estimate the recall of an NER system, but not the precision, since many of the system's guesses that don't match the gold standard may still be correct. To estimate precision for NER on this dataset, we took a random sample of 100 examples of each system's guesses, and manually judged whether they were correct or not. For EL, we measured precision and recall only for the gold-standard mentions in the dataset.

Our training dataset consists of 10,000 random Wikipedia pages, where all of the phrases that link to other Wikipedia articles are treated as mentions, and the target Wikipedia page is the label. The dataset is made available by Ratinov *et al.*. For training, we disregard mentions for which there is no Freebase entity that links to the target Wikipedia page; for the remaining mentions, we use the corresponding Freebase entity as the label. We ended up with 158,715 labeled mentions with an average of 12.62 candidates per mention. The total number of unique mentions in the data set is 77,230 with a total of 974,381 candidate entities and 643,810 unique candidate entities.

Evaluation Metric: For NER evaluation, we measure precision, recall and F1 scores as shown in [7, 23]. To evaluate EL accuracy, we report on a Bag-of-Freebase ids (BOF) evaluation analogous to the Bag-of-Titles (BOT) F1 evaluation as introduced by [21, 24]. In BOF-F1, we compare the set of IDs output for a document with the gold set of IDs for that document (ignoring duplicates), and utilize standard precision, recall, and F1 measures. We separately computed exact-match accuracy, but the results were very similar to the BOF-F1 score, and we chose to report BOF-F1 as it allows for comparison with previous systems. For more details on BOT-F1, see [24]. Note that by construction, BOT-F1 and BOF-F1 provide identical numbers for our experiment.

Competitors: We compare NEREL with 2 state-of-the-art NER systems: the current state-of-the-art NER system for the benchmark CoNLL 2003 test set, the UIUC NER system [23]; and the Stanford NER system [7]. The former uses a regularized averaged perceptron model and external gazetteers for strong performance. The latter uses Conditional Random Fields and Gibbs sampling to incorporate long-distance dependencies into its NER model; the constraints ensure that the same phrase should be labeled consistently within the same document. We downloaded these two systems and ran the configurations that were reported to perform best on the CoNLL dataset. We found the results to be almost the same as reported by their authors.

The six EL competitors for NEREL include: 1) Cuc07 [4], which was the first system to use collective classification of entity links. 2) MW08 [21], which uses the set of unambiguous mentions in the text surrounding a mention to define the mention's context. MW08 uses Normalized Google Distance to compute the similarity between this context and the candidate Wikipedia entry. 3) Kul09 [12] is an extension of both Cuc07 and MW08. It uses a hill-climbing approach to compute the joint probability distribution of all entity link assignments for a document. 4) Rat11 [24], the current state-of-the-art wikifier, has both a local and a global component. The local model uses techniques similar to traditional EL systems [20, 1] and similar to the Wikipedia-features used in our BASEEL model. The global component uses the predictions of the local model for all mentions in a document as the context of a mention, and they use both NGD and PMI [29] to compute the similarity between this context and a candidate Wikipedia page. 5) Hof11 [11], a state-of-the-art database EL system, links mentions to the YAGO2 knowledge base. Their technique uses a dense coherence graph algorithm to measure mention-entity similarity. 6) Sil12 represents our re-implementation of Sil *et al.*'s [25] domain-independent database EL system, which we used as a starting point for our BASEEL model. The four wikifiers and Hof11 use a collective classification framework, but only for collectively classifying sets of entity links, not for combining NER and EL. Sil12 and BASEEL are purely local models.

Parameter Settings: NEREL includes two important parameters: σ, the regularization weight; and the number of candidate entities we select from BASEEL. We set the value of σ by trying five possible values in the range [0.1, 10] on held-out data. We chose 5 random documents as development data from the CoNLL-2003 training set. We found $\sigma = 0.5$ to work best for our experiments. We chose to select 3 candidate entities from BASEEL for each candidate mention (or fewer if BASEELhad fewer than 3 links with nonzero probability), as this higher thresholds led to a computationally intractable number of entity-mention tuples. As mentioned previously, using 3 candidate links per mention resulted in a maximum possible recall for NEREL of 0.96 across our test sets, which is significantly higher than the maximum possible recall of 0.82 for pipeline models.

Results: Table 5 shows the performance of NEREL on the NER task. NEREL achieves the highest recall and F1 scores on two of the three datsets, and second-highest on the third dataset (CoNLL). The Stanford NER system has the best precision on all three datasets, and the UIUC system has the best overall performance on CoNLL. On MSNBC, NEREL outperforms both the state-of-the-art UIUC and Stanford NER systems by 0.14 and 0.87 gains in F1 score respectively. On the other datasets, differences are smaller, but NEREL consistently performs very well. We suspect that the gazetteers in the UIUC system help significantly on the CoNLL dataset. Another problem for NEREL on CoNLL was the presence of several long chains of 20 or more entities, which occurred in several articles reporting the results of cricket matches. NEREL tended to have much lower accuracy on these long chains of entities, which behaved very differently from sequences of entities in normal text.

Dataset	System	Prec	Rec	F1
ACE	UIUC	92.0	89.3	90.7
	Stanford	**97.0**	84.8	90.5
	NEREL	92.0	**92.4**	**92.2**
MSNBC	UIUC	69.8	75.4	72.5
	Stanford	**86.0**	72.2	78.5
	NEREL	83.7	**91.0**	**87.2**
CoNLL	UIUC	91.2	**90.5**	**90.9**
	Stanford	**95.1**	78.3	85.9
	NEREL	86.8	89.5	88.2

Table 5: NEREL outperforms two state-of-the-art NER systems on two out of three datasets, and outperforms one of them on the third dataset.

	ACE			MSNBC		
Systems	Prec	Rec	F1	Prec	Rec	F1
MW08*	-	-	72.8	-	-	68.5
Rat11*	-	-	77.3	-	-	74.9
Sil12*	82.1	74.7	78.2	87.6	67.1	75.9
BASEEL	84.6	77.0	80.7	**89.3**	68.4	77.5
NEREL	**85.5**	**86.4**	**85.9**	85.8	**83.4**	**84.6**

Table 6: NEREL outperforms all competitors on the ACE and MSNBC datasets. We obtain the numbers for MW08 and Rat11 from [24], which does not report precision and recall. * indicates that the EL results are based on tests where gold-standard mentions are given as input.

Table 6 compares NEREL with previously reported results by MW08 and Rat11 on the ACE and MSNBC datasets, as well as the results of our implementations of Sil12 and BASEEL. NEREL achieves an F1 score of 85.9 on ACE and 84.6 on MSNBC, clearly outperforming all competitors. Compared with the state-of-the-art Rat11 wikifier, NEREL improves by 0.086 and 0.097 F1 on ACE and MSNBC respectively. Part of this is due to better linking accuracy, probably resulting from our use of Freebase's attributes, relations, and types for constructing features. Even the BASEEL and the relatively simple Sil12 system (ported to Freebase) outperform the Rat11 wikifier, despite the fact that the Rat11 wikifier is given gold-standard mentions as input, and BASEEL is given the output of the UIUC NER system.

However, another important aspect of NEREL's performance is its ability to correct poor mention boundaries that propagate to errors in linking. Compared with BASEEL, NEREL achieves significantly higher recall and F1 than BASEEL, similar to the differences between NEREL's NER performance and the performance of the UIUC NER system that BASEEL uses in its pipeline. Overall, NEREL improves by 0.052 and 0.071 F1 over BASEEL, its closest competitor, on these two benchmark datasets.

Table 7 illustrates the EL performance of NEREL on the CoNLL dataset. In order to compare with Hoffart *et al.* [11], we replicate their experimental evaluation: systems are evaluated according to their precision at their maximum possible recall level. Furthermore, no mention candidates with NIL links are considered; in Hoffart *et al.*'s experiments, this resulted in the removal of 20% of the mention candidates. In Hoffart *et al.*'s evaluation, each system was supplied with the mentions from the Stanford NER system, and thus their recall was limited by the recall of the Stanford NER system. Hoffart *et al.* do not report this recall, but in our experiments we found that the Stanford system had a recall of 0.783 on the CoNLL test set, or 0.801 when excluding NIL mentions. As before, NEREL

	Systems			
	NEREL	Hof11	Kul09	Cuc07
Prec@Rec=max	**84.22**	81.91	76.74	43.74

Table 7: NEREL outperforms all previously-tested competitors on the CoNLL-2003 (testb) test set. Precision is shown at the highest recall achievable by each system. All competitors to NEREL use only the correct mentions detected by the Stanford NER system as input, and thus their recall is limited by the recall of the Stanford NER system.

uses its overgeneration techniques to generate mention candidates. Its recall on this test set is 0.826; thus NEREL outperforms the next-best system, Hoff11, in both precision and recall.

7. CONCLUSION

Much of the previous research on entity linking has gone into improving linking accuracy over gold-standard mentions, but we observe that many of the common errors made by entity linkers in practice have to do with the pipeline architecture, which propagates errors from named-entity recognition systems to the entity linkers. We propose a re-ranking model that performs joint named entity recognition and entity linking. The discriminative re-ranking framework allows us to introduce features into the model that capture the dependency between entity linking decisions and mention boundary decisions, which existing models do not handle. Furthermore, the model can handle collective classification of entity links, at least for nearby groups of entities. The joint NER and EL model has strong empirical results in our initial experiments, outperforming a number of state-of-the-art NER and EL systems on several benchmark datasets while remaining computationally inexpensive. This illustrates that many of the problems with the pipeline architecture for EL can be overcome by the re-ranking framework.

8. OPEN RESEARCH DIRECTIONS

For future work we want to explore joint models for EL and coreference resolution and relation extraction. Current models for coreference resolution usually rely on a pipeline architecture using anaphoric as well as syntactic features where each candidate mention-pairs are considered independently of the other. We believe that these traditional models can be improved using re-ranking algorithms similar to NEREL as described in this proposal paper. Similar techniques can also be extended to the task of relation extraction, where mentions in a particular document can be automatically tagged with their target Freebase links and further processing can be performed by exploring relationships stored in Freebase among them. This can further boost the performance of a relation extraction system.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1218692.

9. REFERENCES

[1] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, 2006.

[2] Y. Chen and J. Martin. Towards Robust Unsupervised Personal Name Disambiguation. In *EMNLP*, pages 190–198, 2007.

[3] R. Cilibrasi and P. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.

[4] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–Ű716, 2007.

[5] A. Davis, A. Veloso, A. S. da Silva, W. Meira Jr, and A. H. Laender. Named entity disambiguation in streaming data. In *ACL*, 2012.

[6] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, 2010.

[7] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[8] S. Guo, M.-W. Chang, and E. Kıcıman. To link or not to link? a study on end-to-end tweet entity linking. In *NAACL*, 2013.

[9] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *SIGIR*, 2011.

[10] X. Han and J. Zhao. Named entity disambiguation by leveraging Wikipedia semantic knowledge. In *CIKM*, pages 215–Ű224, 2009.

[11] J. Hoffart, M. A. Yosef, I. Bordino, H. Furstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum1. Robust Disambiguation of Named Entities in Text. In *EMNLP*, pages 782–792, 2011.

[12] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, pages 457–Ű466, 2009.

[13] T. Kwiatkowski, L. Zettlemoyer, S. Goldwater, and M. Steedman. Lexical Generalization in CCG Grammar Induction for Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[14] T. Lin, Mausam, and O. Etzioni. Entity Linking at Web Scale. In *AKBC-WEKEX*, 2012.

[15] T. Lin, Mausam, and O. Etzioni. No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities. In *EMNLP*, 2012.

[16] G. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *CoNLL*, 2003.

[17] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM*, 2012.

[18] P. N. Mendes, M. Jakob, and C. Bizer. Evaluating DBpedia Spotlight for the TAC-KBP Entity Linking Task. In *TAC*, 2011.

[19] P. N. Mendes, M. Jakob, and C. Bizer. DBpedia for NLP: A Multilingual Cross-domain Knowledge Base. In *LREC*, 2012.

[20] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM*, pages 233–Ű242, 2007.

[21] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, 2008.

[22] V. Punyakanok and D. Roth. The use of classifiers in sequential inference. 2001.

[23] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 2009.

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

[25] A. Sil, E. Cronin, P. Nie, Y. Yang, A.-M. Popescu, and A. Yates. Linking Named Entities to Any Database. In *EMNLP-CoNLL*, 2012.

[26] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. *NIPS*, 2003.

[27] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Seventh Conference on Natural language learning at HLT-NAACL 2003-Volume 4*, 2003.

[28] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *JMLR*, 2006.

[29] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Procs. of ACL*, pages 417–424, 2002.

[30] Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney. Resolving surface forms to wikipedia topics. In *Coling*, pages 1335–Ű1343, 2010.