

# Scaling Wikipedia-based Named Entity Disambiguation to Arbitrary Web Text

Anthony Fader, Stephen Soderland, and Oren Etzioni

Turing Center

Department of Computer Science and Engineering

University of Washington

Box 352350

Seattle, WA 98195, USA

{afader, soderlan, etzioni}@cs.washington.edu

## Abstract

This paper investigates the “named-entity disambiguation” task on the Web—identifying the referent of a string, found on an arbitrary Web page. The GROUNDER system, introduced in this paper, addresses two challenges not considered by previous work: how to utilize *a priori* information (e.g., Bill Clinton is more prominent on the Web than Clinton County) to improve disambiguation, and how to compose this prior information with contextual evidence.

GROUNDER addresses both challenges by leveraging the user-contributed knowledge in Wikipedia and providing a novel formulation of the task. On a sample of strings drawn from the Web, GROUNDER achieves precision of 1.0 at recall 0.34, and precision 0.90 at recall 0.60.

## 1 Introduction and Motivation

The problem of determining the referent of a word or phrase has its roots in the philosophy of language where Gottlob Frege analyzed the distinction between the meaning and referent of the phrase “the morning star”, and Hilary Putnam considered whether “water” refers to the same substance in a hypothetical “twin earth” where water has the same functional role but a different chemical composition [Putnam, 1975]. In the AI and database literatures, more pragmatic versions of the problem have been explored under the headings entity deduplication, reference reconciliation, and more [Yates and Etzioni, 2007; Singla and Domingos, 2005]. We are interested in the problem as it manifests on arbitrary Web text, which means that we cannot restrict the entities to particular types as in [Singla and Domingos, 2005] for example. In contrast with [Yates and Etzioni, 2007], we leverage the user-contributed knowledge in Wikipedia.

Thus, this paper investigates the “named-entity disambiguation” task on the Web—identifying the referent of a string, found on an arbitrary Web page, leveraging the set of entities described by Wikipedia articles. Seminal work by Bunescu and Pasca [2006] and Cucerzan [2007] introduced this task. However, their work suffers from a key limitation:

it does not factor *a priori* information into the disambiguation decision.

Not all entities are “created equal”—some are *a priori* more likely to serve as the referents of textual strings. Consider, for example, the string “Clinton”—it could potentially refer to Bill, Hillary, or Roger Clinton or even to Clinton County. However, *a priori* the string is much more likely to refer to Bill or Hillary Clinton than to Clinton County given Bill and Hillary’s prominence on the Web. Of course, we can’t ignore the possibility that, in some contexts, the string *does* refer to Clinton County. Contextual evidence might cause us to map the string “Clinton” to Clinton County, but it would have to be fairly strong evidence. Thus, we need a means of quantifying *a priori* information, and a method for combining it with contextual evidence to yield disambiguation decisions.

This paper reports on the GROUNDER system, the first named-entity disambiguation system that composes *a priori* prominence information with contextual evidence to yield superior disambiguation decisions. Our contributions are as follows:

- We introduce a novel formulation of the task based, and highlight the value of a prior over entities for this task.
- We present an overview of the GROUNDER system, which operationalizes this formulation into a working system that draws its prior automatically from Wikipedia.
- We report on a set of experiments that demonstrate the value of using prior information in concert with contextual evidence. GROUNDER achieves a precision of 90% at a recall of 60%.

The remainder of the paper is organized as follows. Section 2 describes closely related work. Section 3 provides an overview of the GROUNDER system. Section 4 describes our experimental results, and Section 5 concludes with a discussion of future work.

## 2 Previous Work

The problem of named entity disambiguation has a long history in the database community, where the task is presented as a classification problem: given a vector of similarities between two records in a database, output whether they refer

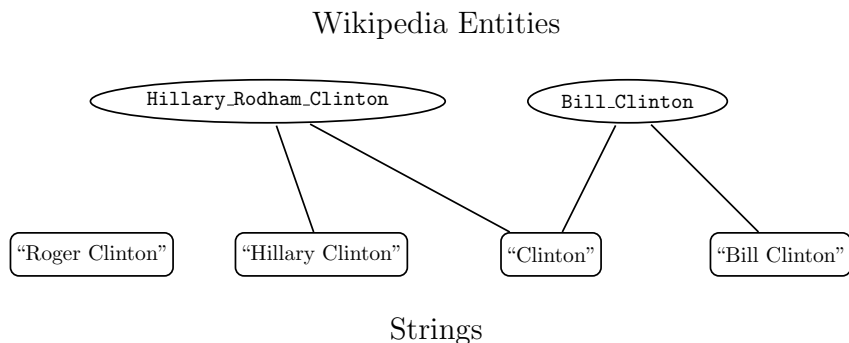


Figure 1: A possible relationship between Wikipedia entities (top) and a set of strings (bottom). Mapping ambiguous strings such as “Clinton” to the correct entity pose a harder problem than unambiguous strings such as “Hillary Clinton”.

to the same entity or not [Fellegi and Sunter, 1969]. Work that is closer to ours leverages a graph of known relationships between entities to find the entity that best matches a given reference [Bhattacharya and Getoor, 2006a]. Our work is similar in that it relies on the Wikipedia graph of entities to obtain a measure of entity prominence.

There has also been work on providing a probabilistic framework for named entity disambiguation in text [Li *et al.*, 2004; Bhattacharya and Getoor, 2006b]. However, these frameworks make assumptions about the types of named entities and cannot be applied directly to the problem of named entity disambiguation on the Web.

The work presented in this paper builds on the ideas described in papers by Bunescu and Pasca [2006] and by Cucerzan [2007], both of which attempt to disambiguate named entities by mapping them to Wikipedia articles. Each of these papers make use of the fact that the context surrounding an ambiguous string gives useful evidence for disambiguating it. Given an ambiguous string  $s$  and its context, their systems find all Wikipedia articles that can be referred to as  $s$ , and “ground”  $s$  to the article whose content best overlaps with the context of  $s$ .

Bunescu and Pasca [2006] measure this overlap using tf-idf cosine similarity. Bunescu and Pasca found that the text of Wikipedia articles is often not enough to disambiguate an ambiguous string, despite its sense being clear from the context. To address this issue, they use a supervised learning technique to enrich a given Wikipedia article’s term vector with words from articles in the same category. They evaluate their system on Wikipedia articles and obtain accuracies ranging from 77.2% to 84.8%.

Cucerzan [2007] takes a similar approach, but uses context vectors consisting of key words and short phrases extracted from Wikipedia. Cucerzan’s system also attempts to disambiguate all named entities in a single context simultaneously, adding the constraint that the target Wikipedia articles should be from the same categories. He evaluated his system on Wikipedia articles and a set of 100 news articles, obtaining accuracies of 88.3% and 91.4%, respectively.

Our work builds on these two approaches and follows their lead on using Wikipedia as a database of candidate entities. However, these previous systems focus only on contextual

evidence for disambiguation and fail to include an explicit notion of prior information about the relationship between a string and its referent entities. We show that for the named entity disambiguation problem, prior information about entity prominence turns out to be very useful. This prior information can be used in conjunction with contextual information and we show that doing so leads to better performance than either component in isolation.

Another important difference between this work and the previous two is that we evaluate our system on domain-independent Web text, as opposed to only news and Wikipedia articles. We obtained performance on arbitrary Web text that is consistent with the performance on the previous systems.

### 3 The GROUNDER System

This section introduces the GROUNDER system, which uses a novel formulation of the named entity disambiguation problem. The key insight used in GROUNDER is that *a priori* information about the ambiguity of a string is valuable for named entity disambiguation. We call this type of information a *prominence prior* over strings and entities. We can think of a prominence prior as representing the GROUNDER system’s real-world knowledge about the ambiguity of a string  $s$ : what are the entities  $s$  could possibly refer to and with what probability?

In order to leverage this type of prior information, we are faced with two challenges.

The first challenge is finding a source of information to use as our prominence prior. Our solution to this problem relies on the fact that Wikipedia’s network structure naturally encodes a measure of entity prominence: highly linked articles tend to be about more prominent entities than articles with fewer incoming links. In addition to this, Wikipedia’s article titles and redirect pages provide a large amount of information about the entities a string can refer to. We combine these two sources of information to compute a prominence prior, which lets us answer questions about the likely referents of a string independent of its context.

Given a source of information for our prominence prior, the second challenge we face is the question of *how* to combine it with contextual evidence. We solve this problem by com-

PROBDISAMBIG( $s, D, \tau$ ):

1.  $e^* = \arg \max_{e \in E} P(D|s \rightarrow e) P(s \rightarrow e)$
2. **if**  $P(s \rightarrow e^*|D) > \tau$ , **return**  $e^*$
3. **else return** *NoEntity*

Figure 2: Pseudocode for a named entity disambiguation algorithm based on the model defined in Section 3.1. The inputs are:  $s$ , a named entity string;  $D$ , the document containing  $s$ ; and  $\tau$ , the minimum value that the posterior probability  $P(s \rightarrow e|D)$  must obtain for  $e^*$  to be returned.

binning contextual and prior information via Bayes’ theorem.

In the following sections, we describe in detail GROUNDERS’ novel model of named entity ambiguity, our source of contextual evidence for disambiguation, and the Wikipedia-based prominence prior outlined above.

### 3.1 A Novel Model for Named Entity Disambiguation

Suppose that we observe a named entity string  $s$  in a document  $D$ . Given a database  $E$  of candidate entities, we define the *named entity disambiguation problem* as the task of maximizing  $P(s \rightarrow e^*|D)$ , i.e. finding the entity  $e \in E$  that  $s$  most likely refers to in the context given by  $D$ . We can express this as the problem of maximizing the probability  $P(s \rightarrow e|D)$  over all entities  $e \in E$ , where the notation  $s \rightarrow e$  represents the event that  $s$  refers to  $e$ . Rewriting this using Bayes’ theorem, we can restate named entity disambiguation as finding the entity  $e^*$  in the following optimization problem:

$$e^* = \arg \max_{e \in E} P(D|s \rightarrow e)P(s \rightarrow e). \quad (1)$$

We make the simplifying assumption that the normalizing constant  $P(D)$  in Bayes’ theorem is uniform in order to threshold the value of  $P(s \rightarrow e|D)$  to control precision and recall.

The two factors on the right hand side of Equation 1 correspond to the two sources of information that we include in GROUNDERS. The first factor  $P(D|s \rightarrow e)$  represents the likelihood of seeing the document  $D$  given that we know it contains a reference to  $e$ . We can interpret the role of this in the optimization as the source of *contextual evidence* that the document  $D$  gives us. The second factor  $P(s \rightarrow e)$  is the prior probability of the string  $s$  referring to  $e$ . This is the *prominence prior* that was introduced in the previous section and can be thought of as a measure of the ambiguity of  $s$ .

This view of named entity disambiguation has three clear benefits. First, it corresponds to our intuition about how human readers disambiguate a string  $s$ . If we believe that  $s$  refers to  $e$  with high probability, then it would take a lot of contextual evidence to convince us otherwise. On the other hand, if  $s$  is ambiguous, then the role of context becomes more important for our decision.

The second benefit of GROUNDERS’ model is that it gives us a measure of the uncertainty of the most likely entity  $e^*$

in Equation 1. This is useful for controlling the precision and recall of a system that uses the model. For example, we could choose a threshold parameter  $\tau \in [0, 1]$  and instruct the computer to ground only the strings  $s$  such that  $P(s \rightarrow e^*|D) > \tau$ .

Lastly, a third benefit of our model is that it provides a very general framework for incorporating different types of information into the disambiguation process. By changing the details of the contextual and prior model components, we can account for new types of evidence. In this sense, our model describes a family of algorithms for named entity disambiguation. The pseudocode for an algorithm based on this model is shown in Figure 2.

### 3.2 GROUNDERS Implementation

In the following sections, we describe our implementation of the GROUNDERS system, which consists of three parts: the entity database  $E$ , the context model component  $P(D|s \rightarrow e)$ , and the prominence prior model component  $P(s \rightarrow e)$ .

#### Wikipedia as an Entity Database

Following the work of Bunescu and Pasca [2006] and Cucerzan [2007], we can treat Wikipedia as an entity database, where each article corresponds to an entity. Wikipedia is well-suited to act as an entity database for named entity disambiguation on the Web: it is domain-independent and contains millions of articles, so many of the named entities mentioned on the Web have a Wikipedia article. Let  $E$  represent the set of all entities in Wikipedia. Define  $Article(e)$  to be the article text of an entity  $e \in E$ . We use a basic filter to remove articles that do not describe entities (e.g., list and category pages).

#### Cosine Similarity Context Model

Our implementation of the context model component  $P(D|s \rightarrow e)$  is based on the assumption that if  $D$  contains a reference to  $e$ , then the words used in  $D$  will tend to overlap with the words used in the Wikipedia article  $Article(e)$ . To measure this overlap, we treat  $D$  and  $Article(e)$  as tf-idf vectors and compute the cosine similarity of  $D$  and  $Article(e)$ :

$$P(D|s \rightarrow e) = \cos(D, Article(e)). \quad (2)$$

We computed the idf scores for each term using a Lucene index of Wikipedia as a corpus (see the following section for more information), which defines the idf score of a term  $t$  as

$$idf(t) = 1 + \log \left( \frac{N + 1}{n_t} \right),$$

where  $n_t$  is the number of Wikipedia articles containing the term  $t$  and  $N$  is the total number of Wikipedia articles. We note that our model of context gives a simple measurement of similarity between the context and the Wikipedia articles and is not an actual probability measure.

In order to compare the behavior of the GROUNDERS system to one that ignores the prominence prior information, we define the COSINE-SIM algorithm, which simply returns the entity  $s$  that maximizes the cosine similarity in equation (2):

$$COSINE-SIM(s, D) = \arg \max_{e \in E} \cos(D, Article(e)).$$

The COSINE-SIM algorithm is similar to the baseline method used in [Bunescu and Pasca, 2006], which also uses tf-idf cosine similarity to compare contexts to Wikipedia articles.

### Search Engine Prominence Prior

Given a string  $s$  and an entity  $e \in E$ , how do we come up with a good prior probability  $P(s \rightarrow e)$ ? In order to answer this, consider the following types of evidence that would suggest that  $s$  refers to  $e$ :

1.  $e$  is known to be referred to as  $s$  or a string similar to  $s$
2.  $e$  is referred to with high frequency relative to other entities in  $E$  that can be referred to as  $s$

The first type of information is a necessary condition for the event  $s \rightarrow e$  to have a non-zero probability: there must be some evidence that  $e$  can be referred to as  $s$  or else  $P(s \rightarrow e)$  should be 0. The second type of information is based on the intuition that if  $e$  is more prominent than another entity  $e'$ , then  $P(s \rightarrow e)$  should be greater than  $P(s \rightarrow e')$ .

In the GROUNDER system, we use an existing Lucene-based Wikipedia search engine<sup>1</sup> to calculate these two types of information. This software is currently being used as the public search engine for the English-language Wikipedia. Given a query string  $s$ , the search engine returns a list of scored entities. The search score of an entity  $e$  with respect to the query string is given by

$$\begin{aligned} \text{search-score}(e, s) &= \text{query-match}(e, s) \\ &\times \left( 1 + \log \left( 1 + \frac{\text{in-deg}(e)}{\alpha} \right) \right). \end{aligned}$$

The value  $\text{search-score}(e, s)$  can be interpreted as how likely it is that  $s$  refers to  $e$  in the absence of context. The first factor  $\text{query-match}(e, s)$  is a non-negative score representing how well the query  $s$  matches the article name of  $e$ , the titles of any redirect pages to  $e$ , and the words of the article of  $e$ . The second factor boosts the score of  $e$  proportional to the number of incoming links on Wikipedia, written as  $\text{in-deg}(e)$ . This can be interpreted as a measure of prominence of  $e$  in the Wikipedia hyperlink network. We used the search engine’s default value of  $\alpha = 15$  in our experiments. We define  $\text{search-score}(e, s)$  to be 0 for any  $e$  that is not in the set of articles returned when searching for  $s$ . We also normalize the search scores such that  $\sum_{e' \in E} \text{search-score}(e', s) = 1$ .

In our experiments, we were interested in testing the effects of this measure of prominence on performance, so we introduce a smoothed version of the search score given by

$$\text{search-score}(e, s|\lambda) = \frac{\lambda}{N_{s,e}} + (1 - \lambda) \cdot \text{search-score}(e, s) \quad (3)$$

where  $N_{s,e}$  is the number of entities  $e$  such that  $\text{search-score}(e, s) \neq 0$ . We can think of this as a mixture of a uniform score over all candidate entities with the search score. A value of  $\lambda = 0$  corresponds to the full search score and a value of  $\lambda = 1$  corresponds to a uniform score. We can now define the prominence prior as

$$P(s \rightarrow e|\lambda) = \text{search-score}(e, s|\lambda).$$

<sup>1</sup>Available at <http://www.mediawiki.org/wiki/Extension:Lucene-search>.

```

GROUNDER( $s, D, \lambda, \tau$ ):
1. for  $e \in E$ :
    $score[e] = \cos(D, Article(e)) \cdot \text{search-score}(e, s|\lambda)$ 
3.  $e^* = \arg \max_{e \in E} score[e]$ 
4. if  $score[e] > \tau$  return  $e^*$ 
5. else return  $NoEnt$ 

```

Figure 3: Pseudocode for GROUNDER. The inputs are:  $s$ , a named entity string;  $D$ , the document containing  $s$ ;  $\lambda$ , the prior smoothing parameter; and  $\tau$ , the minimum value that  $P(s \rightarrow e|D, \lambda)$  must obtain for  $e^*$  to be returned.

To compare the GROUNDER system to one that ignores contextual evidence, we introduce the PRIOR algorithm, which simply returns the entity that maximizes the prior probability in (3), given a smoothing parameter  $\lambda$ :

$$PRIOR(s, \lambda) = \arg \max_{e \in E} P(s \rightarrow e|\lambda).$$

### The GROUNDER Algorithm

Now that we have defined the necessary components of the model described in Section 3.1, we can combine them and formally define GROUNDER. The GROUNDER assigns each entity  $e \in E$  a score  $score[e]$ , which is the product of its local context probability and the prior prominence probability. Figure 3 shows the pseudocode of GROUNDER, which includes a threshold parameter  $\tau \in [0, 1]$  to control precision and recall.

## 4 Experimental Results

This section evaluates GROUNDER’s performance in grounding strings drawn from a “random” set of Web pages. Section 4.1 describes our method of creating a dataset and Section 4.2 uses this dataset to characterize the problem space for grounding arbitrary proper nouns to Wikipedia concepts. Sections 4.3 and 4.4 present our experimental results.

### 4.1 Creating a Web-based Dataset

We began with a corpus of approximately 500 million Web pages of arbitrary topics and genres, including blogs, news articles, and online stores. Associated with this Web corpus is a set of tuples extracted by the TEXTRUNNER Open IE system [Banko *et al.*, 2007; Banko and Etzioni, 2008]. These tuples are extractions in the form  $(arg_1, pred, arg_2)$  where  $arg_1$ ,  $pred$ , and  $arg_2$  are text strings and  $pred$  expresses a relation between  $arg_1$  and  $arg_2$  (e.g., (“Clinton”, “born in”, “Hope, AR”)).

We collected a set of 500 tuples and their Web page of origin, uniformly sampling from a collection that have a proper noun as  $arg_1$  and were manually verified to be correct extractions. We took the  $arg_1$  values as our set of strings  $s$  and the associated Web page as the document  $D$  associated with  $s$ .

To create a gold standard from this dataset, we manually identified the entity  $e \in E$  to which that  $s$  refers (and set

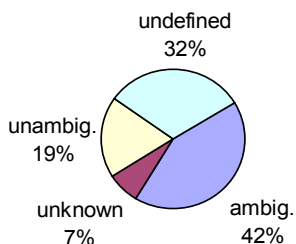


Figure 4: Nearly one third of proper nouns in our dataset have no Wikipedia page (undefined). Only 19% are the easy cases that unambiguously refer to a single Wikipedia page.

$e = NoEntity$  if Wikipedia did not contain an article for that entity). This gives us a set of 500 triples  $(s, D, e)$ , where  $s$  is proper noun string occurring on document  $D$ , and  $e$  is the entity to which that  $s$  actually refers.

## 4.2 Characterizing the Problem Space

This section addresses the following questions: What proportion of the strings  $s$  in our dataset are relatively easy to map to the correct  $e$ ? What proportion present a relatively difficult disambiguation problem? And what proportion are impossible to map to Wikipedia because a Wikipedia page for  $s$  does not exist?

To help answer these questions, we distinguish between four types of  $s$ : *undefined*, *unambiguous*, *ambiguous*, and *unknown*. Undefined strings are cases where  $s$  refers to an entity that does not have a Wikipedia page. The remaining three cases are defined in terms of the set of Wikipedia entities  $Ents(s)$  that can be referred to by  $s$ . A string  $s$  is unambiguous if  $Ents(s)$  contains exactly one Wikipedia page; it is ambiguous if  $|Ents(s)| > 1$ ; and it is unknown if there is a Wikipedia page  $e$  that corresponds to  $s$ , but  $s$  is not among the known ways to refer to  $e$ , i.e.  $|Ents(s)| = 0$ .

We compute  $Ents(s)$  from information that Wikipedia provides about the different ways to which an entity  $e \in E$  can be referred, following the methodology of Bunescu and Pasca [2006] and Cucerzan [2007]. This information comes from four different sources: article titles, redirect pages, disambiguation pages, and hyperlink anchor text. For each  $e \in E$ , we can define a set  $Names(e)$  containing the strings that are known to refer to  $e$  in Wikipedia. We can also define the set  $Ents(s) = \{e \in E : s \in Names(e)\}$ , which is the collection of entities that  $s$  can refer to in Wikipedia.

We can represent the relationship between strings and entities as a bipartite graph, where there is an edge between a string  $s$  and an entity  $e$  when  $s \in Names(e)$ . Figure 1 shows a simple example relating a set of entities  $\{Bill\_Clinton, Hillary\_Rodham\_Clinton\}$  to a set of strings  $\{\text{“Bill Clinton”, “Hillary Clinton”, “Roger Clinton”, “Clinton”}\}$ . In this toy example, the strings “Bill Clinton” and “Hillary Clinton” are unambiguous, the string “Clinton” is ambiguous, and the string “Roger Clinton” is undefined.

Let  $\mathcal{D}$  be the set of 500 triples  $(s, D, e)$ , where  $s$  is the first argument of the extraction,  $D$  is the text of the document containing the extraction, and  $e$  is the entity that  $s$  actually refers to. Figure 4 shows how the string-entity pairs  $(s, e)$

in  $\mathcal{D}$  are distributed relative to the information in Wikipedia. 32% of the  $s$  in our dataset are undefined – no method using the *Ents* and *Names* relations can possibly map them to a Wikipedia page. Only 19% constitute the easy case of strings that unambiguously map to a single Wikipedia page, such as “Hillary Clinton”.

The remainder are the hard cases. To make matters worse, we found that 31 of the 208 ambiguous  $s$  did not include the correct  $e$  in the set of candidate entities  $Ents(s)$ . Hence, an entity grounding algorithm that relies on the set of known references to  $s$  on Wikipedia, will fail on these cases as well as on the *unknown* cases. We need an entity grounding algorithm that is not limited by the incompleteness of  $Names(e)$ . The Grounder system avoids this problem by using a search engine over Wikipedia articles, allowing inexact matches on article titles and redirect pages.

## 4.3 Evaluation of Grounder

This section compares the full Grounder algorithm with its two main components—PRIOR and COSINE-SIM. Our experiments utilize recall and precision metrics. To do so, we created a ranked list of results for each method on the dataset  $\mathcal{D}_W$ , the subset of  $\mathcal{D}$  such that  $e \in E$ , ordered by that method’s confidence score. As we vary a threshold  $\tau$ , we can define recall as the percentage of correctly grounded strings  $s$  with confidence greater than  $\tau$  divided by all possible correct groundings in  $\mathcal{D}_W$ . Precision is the percentage of correctly grounded  $s$  divided by the number of results with confidence greater than  $\tau$ .

Figure 5 shows a recall-precision curve for COSINE-SIM, which uses Cosine Similarity exclusively; for PRIOR, which uses the prominence prior computed from the search engine scores; and the full Grounder system. For this experiment we set  $\lambda = 0.0$ , which uses the unsmoothed prior.

The COSINE-SIM method suffers from lower recall and precision than either of the other methods. Combining the two knowledge sources gives the best result, with a recall-precision curve that is consistently higher than PRIOR alone. While cosine similarity is informative, it is unable to make many fine-grained distinctions. The Prior given by the Lucene search engine score completely ignores local context from the page  $D$ , but succeeds when  $s$  is either unambiguous (e.g. “Hillary Clinton”) or when an ambiguous  $s$  refers to the most prominent Wikipedia page. PRIOR can achieve precision 1.0 for 31% of our dataset  $\mathcal{D}_W$ .

The remaining cases are the hardest ones to disambiguate, where local context is necessary to find the correct entity. We examined the “hard cases” in which  $s$  does not refer to the dominant sense of  $s$ . In these cases, where PRIOR is always wrong, we found that COSINE-SIM is able to distinguish the correct entity about 40% of the time. This number is lower than the previously reported results because it corresponds to the subset of strings that do not refer to their most common sense.

## 4.4 Parameter Settings to Combine Knowledge Sources

The key to Grounder’s success is the appropriate combination of the information in its two components: PRIOR and

## Algorithm Performance

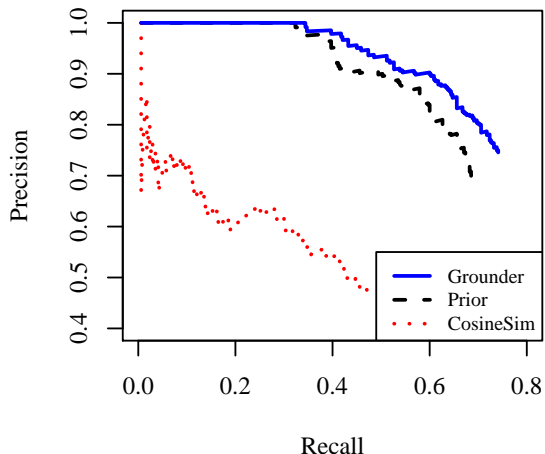


Figure 5: Combining both the search engine score (Prior) and the Cosine Similarity gives better performance than either knowledge source alone.

COSINE-SIM, which is controlled by the smoothing parameter  $\lambda$ . We carried out a sensitivity analysis to demonstrate that we chose a value for  $\lambda$  that is close to optimal.

In the analysis, we varied the parameter  $\lambda$  by increments of 0.2. Figure 6 shows the results for  $\lambda$  from 0.0 to 1.0. The lowest curve in Figure 6 is for  $\lambda = 1.0$ , which is identical to the COSINE-SIM algorithm. The results improve monotonically until  $\lambda = 0.0$  which gives equal weight to both the prior and contextual evidence scores. Thus, we believe we have a value of  $\lambda$  that is close to optimal.

## 5 Conclusions and Future Work

This paper presented our Grounder system, which disambiguates named entities mentioned in text by mapping them to Wikipedia pages. A novel feature of the Grounder system is that it combines local contextual evidence with the prior probability given by search engine scores. We show that this method scales well to arbitrary Web text. Grounder achieves precision of 1.0 at recall 0.34 and precision 0.90 at recall 0.60.

We did not include any type-specific or genre-specific knowledge in Grounder to ensure that it scales to arbitrary entities. A future direction of research is to incorporate more types of evidence into the contextual component of the probabilistic model. For example, including type-specific coreference resolution to create soft constraints on entity types might be useful.

Another interesting direction would be to explicitly model the joint distribution  $P(s_1 \rightarrow e_1, \dots, s_n \rightarrow e_n | D)$  of a set of ambiguous strings  $s_1, \dots, s_n$  on  $D$ . One could imagine that knowledge about the referent entity of  $s_i$  would provide information about the referent entities of the other strings in  $D$ .

Finally, we plan to embed Grounder as a module in a variety of textual inference systems including cross-document reference resolution systems, textual entailment systems.

## Performance with Smoothed Prior

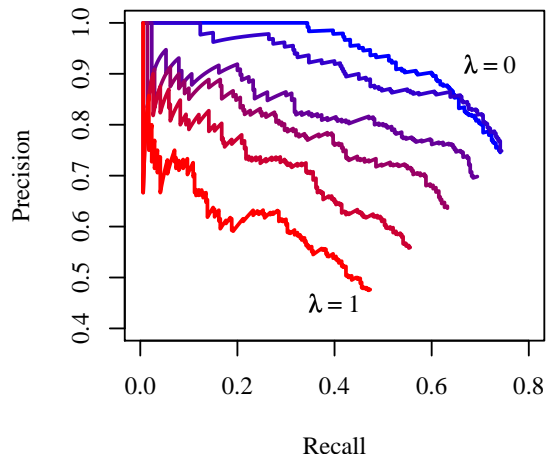


Figure 6: Performance improves as we vary  $\lambda$  from 1.0 to 0.0, at which point equal weight is given to the search engine score (Prior) and the Cosine Similarity.

## 6 Acknowledgements

We thank Stefan Schoenmackers, Mausam, and Dan Weld for comments on earlier drafts and Silviu Cucerzan for helpful discussion.

This research was supported in part by NSF grant IIS-0803481, ONR grant N00014-08-1-0431 as well as gifts from Google, and carried out at the University of Washington’s Turing Center.

## References

- [Banko and Etzioni, 2008] M. Banko and O. Etzioni. The tradeoffs between traditional and open relation extraction. In *Proceedings of ACL*, 2008.
- [Banko *et al.*, 2007] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the Web. In *Procs. of IJCAI*, 2007.
- [Bhattacharya and Getoor, 2006a] Indrajit Bhattacharya and Lise Getoor. Entity resolutions in graphs. In D. Cook and L. Holder, editors, *Mining Graph Data*. Wiley, 2006.
- [Bhattacharya and Getoor, 2006b] Indrajit Bhattacharya and Lise Getoor. A latent dirichlet model for unsupervised entity resolution. In *SIAM Conference on Data Mining (SDM)*, April 2006. Winner of the Best Paper Award.
- [Bunescu and Pasca, 2006] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*. The Association for Computer Linguistics, 2006.
- [Cucerzan, 2007] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP-CoNLL 2007*, pages 708–716, 2007.
- [Fellegi and Sunter, 1969] Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

- [Li *et al.*, 2004] Xin Li, Paul Morie, and Dan Roth. Robust reading: Identification and tracing of ambiguous names. In *HLT-NAACL*, pages 17–24, 2004.
- [Putnam, 1975] Hilary Putnam. The meaning of ‘meaning’. In K. Gunderson, editor, *Language, Mind, and Knowledge*. University of Minnesota Press, 1975.
- [Singla and Domingos, 2005] Parag Singla and Pedro Domingos. Collective object identification. In *IJCAI-2005*, pages 1636–1637, 2005.
- [Yates and Etzioni, 2007] A. Yates and O. Etzioni. Unsupervised resolution of objects and relations on the Web. In *Procs. of HLT*, 2007.