Adding High-Precision Links to Wikipedia

Thanapon Noraset **Chandra Bhagavatula** **Doug Downey**
Department of Electrical Engineering & Computer Science
Northwestern University
Evanston, IL 60208
{nor|csbhagav}@u.northwestern.edu, ddowney@eecs.northwestern.edu

Abstract

Wikipedia's link structure is a valuable resource for natural language processing tasks, but only a fraction of the concepts mentioned in each article are annotated with hyperlinks. In this paper, we study how to augment Wikipedia with additional high-precision links. We present *3W*, a system that identifies concept mentions in Wikipedia text, and links each mention to its referent page. *3W* leverages rich semantic information present in Wikipedia to achieve high precision. Our experiments demonstrate that *3W* can add an average of seven new links to each Wikipedia article, at a precision of 0.98.

1 Introduction

Wikipedia forms a valuable resource for many Natural Language Processing and Information Extraction tasks, such as Entity Linking (Cucerzan, 2007; Han and Zhao, 2009), Ontology Construction (Wu and Weld, 2008; Syed et al., 2008) and Knowledge Base Population (Hoffart et al., 2013; Lehmann et al., 2013). Wikipedia's links provide disambiguated semantic information. For example, when a system processes the text *"Chicago was received with critical acclaim"* from an article, the system does not need to infer the referent entity of *"Chicago"* if the word is already hyperlinked to the Wikipedia page of the Oscar-winning film. Unfortunately, in Wikipedia only a fraction of the phrases that can be linked are in fact annotated with a hyperlink. This is due to Wikipedia's conventions of only linking to each concept once, and only when the links have a certain level of utility for human readers.[1] We see this as an opportunity to improve Wikipedia as a resource for NLP systems. Our experiments estimate that as of September 2013, there were an average of 30 references to Wikipedia concepts left unlinked within each of English Wikipedia's four million pages.

In this paper, our goal is to augment Wikipedia with additional high-precision links, in order to provide a new resource for systems that use Wikipedia's link structure as a foundation. Identifying references to concepts (called *mentions*) in text and linking them to Wikipedia is a task known as *Wikification*. Wikification for general text has been addressed in a wide variety of recent work (Mihalcea and Csomai, 2007; Milne and Witten, 2008b; McNamee and Dang, 2009; Ratinov et al., 2011). The major challenge of this task is to resolve the ambiguity of phrases, and recent work makes use of various kinds of information found in the document to tackle the challenge. In contrast to this body of work, here we focus on the special case of Wikifying *Wikipedia articles*, instead of general documents. This gives us an advantage over general-text systems due to Wikipedia's rich content and existing link structure.

We introduce *3W*, a system that identifies mentions within Wikipedia and links each to its referent concept. We show how a Wikipedia-specific Semantic Relatedness measure that leverages the link structure of Wikipedia (Milne and Witten, 2008b) allows *3W* to be radically more precise at high levels of yield when compared to baseline Wikifiers that target general text. Our experiment shows that *3W* can add on average seven new links per article at precision of 0.98, adding approximately 28 million new links to 4 million articles across English Wikipedia.[2]

[1] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking)

[2] http://websail.cs.northwestern.edu/projects/3W

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 651–656,
October 25-29, 2014, Doha, Qatar.

2 Problem Definition

In this section, we define our link extraction task. A link l is a pair of a surface form s_l and a concept t_l. A *surface form* is a span of tokens in an article, and the *concept* is a Wikipedia article referred to by the surface form. For existing hyperlinks, the surface form corresponds to the anchor text and the concept is the link target. For example, a hyperlink *[[Chicago City | Chicago]]* has surface form *"Chicago City"* and referent concept *Chicago*.[3] Given documents $\mathcal{D} = \{d_1, ..., d_{|\mathcal{D}|}\}$ and a set of links $\mathcal{L} = \{l_1, .., l_{|\mathcal{L}|}\} \in \mathcal{D}$, our goal is to generate a set of high-precision links $\mathcal{L}^*$ for $\mathcal{D}$, distinct from $\mathcal{L}$. In this paper, the document set $\mathcal{D}$ consists of articles from English Wikipedia, and $\mathcal{L}$ is the set of existing links on Wikipedia.

The task can be divided into 3 steps. The first step is to *extract* a set of potential mentions $\mathcal{M} = \{m_1, .., m_{|\mathcal{M}|}\}$ where m is, similar to l, a pair of surface form s_m and a set of candidate concepts $\mathcal{C}(m) = \{t_1, ..., t_{|\mathcal{C}(m)|}\}$. For m having $|\mathcal{C}(m)| > 1$, we need to *disambiguate* it by selecting only one target concept $t_m \in \mathcal{C}(m)$. Since the correct concept may not exist in $\mathcal{C}(m)$ and the previous step could output an incorrect concept, the final step is to decide whether to *link* and include m in $\mathcal{L}^*$. We describe the details of these steps in the following section.

3 System Overview

In this section, we describe in detail how *3W* adds high-precision links to Wikipedia.

3.1 Mention Extraction

In this step, we are given a document d, and the goal is to output a set of mentions $\mathcal{M}$. Our system finds a set of potential surface forms, s_m, by finding substrings in d that match the surface form of some links in $\mathcal{L}$. For example, from the phrase *"map of the United States on the wall"*, we can match 4 potential surface forms: *"map"*, *"United States"*, *"map of the United States"*, and *"wall"*. Notice that some of them are overlapping. The system selects a non-overlapping subset of the surface forms that maximizes the following score function:

$$Score(\mathcal{M}) = \sum_{m \in \mathcal{M}} \frac{T(s_m) PL(s_m)}{|\mathcal{C}(m)|} \quad (1)$$

[3] http://en.wikipedia.org/wiki/Chicago

where $PL(s_m)$ is the probability that the text s_m is linked (that is, the fraction of the occurrences of the string s_m in the corpus that are hyperlinked), $T(s_m)$ is the number of tokens in s_m, and $|\mathcal{C}(m)|$ is the number of candidate concepts. Intuitively, we prefer a longer surface form that is frequently linked and has a specific meaning. Furthermore, we eliminate common surface forms (i.e. *"wall"*) by requiring that $PL(s_m)$ exceed a threshold. In the previous example, we are left with only *"map of the United States"*.

Because Wikipedia's concepts are largely noun phrases, *3W* only looks for surface forms from top-level noun phrases generated by the Stanford Parser (Socher et al., 2013). In addition, each name entity (NE) (Finkel et al., 2005) is treated as an atomic token, meaning that multi-word NEs such as *"California Institute of the Arts"* will not be broken into multiple surface forms.

Finally, the system pairs the result surface forms with a set of candidate concepts, $\mathcal{C}(m)$, and outputs a set of mentions. $\mathcal{C}(m)$ consists of those concepts previously linked to the surface form in $\mathcal{L}$. For instance, the surface form *"map of the United States"* has been linked to three distinct concepts in English Wikipedia.

3.2 Disambiguation

Given a set of mentions $\mathcal{M}$ from the previous step, The next step is to select a concept $t \in \mathcal{C}(m)$ for each $m \in \mathcal{M}$. We take the common approach of ranking the candidate concepts. *3W* uses a machine learning model to perform pair-wise ranking of $t \in \mathcal{C}(m)$ and select the top-ranked candidate concept. We refer to *3W*'s disambiguation component as the *ranker*. The ranker requires a feature vector for each candidate concept of a mention. The rest of this section describes the features utilized by the ranker. The first two feature groups are commonly used in Wikification systems. The third feature group is specifically designed for mentions in Wikipedia articles.

3.2.1 Prior Probability Features

The conditional probability of a concept t given mention surface s_m, $P(t|s_m)$, is a common feature used for disambiguation. It forms a very strong Wikification baseline ($\sim$ 86% in micro-accuracy). This probability can be estimated using Wikipedia links ($\mathcal{L}$). In addition, we use the *external* partition of the

Google "Cross-Lingual Dictionary" described in (Spitkovsky and Chang, 2012) to get the estimates for the probability from links outside Wikipedia.

3.2.2 Lexical Features

To make use of text around a mention m, we create bag-of-word vectors of the mention's source document $d(m)$, and of a set of words surrounding the mention, referred to as the *context* $c(m)$. To compare with a concept, we also create bag-of-word vectors of candidate concept's document $d(t)$ and candidate concept's context $c(t)$. We then compute cosine similarities between the mention's vectors for $d(m)$ and $c(m)$, with the concept candidate vectors for $d(t)$ and $c(t)$ as in the Illinois Wikifier (Ratinov et al., 2011). In addition to similarities computed over the top-200 words (utilized in the Illinois Wikifier), we also compute similarity features over vectors of all words.

3.2.3 Wikipedia-specific Features

Because the links in an article are often related to one another, the existing links in a document form valuable clues for disambiguating mentions in the document. For each concept candidate $t \in \mathcal{C}(m)$, we compute a Semantic Relatedness (SR) measure between t and each concept from existing links in the source document. Our SR measure is based on the proportion of shared inlinks, as introduced by Milne and Witten (2008b). However, because Milne and Witten were focused on general text, they computed SR only between t and the *unambiguous* mentions (i.e. those m with $|\mathcal{C}(m)| = 1$) identified in the document. In our work, $d(m)$ is a Wikipedia article which is rich in existing links to Wikipedia concepts, and we can compute SR with all of them, resulting in a valuable feature for disambiguation as illustrated in our experiments. We use the SR implementation of Hecht et al. (2012). It is a modified version of Milne and Witten's measure that emphasizes links in Wikipedia article's overview. In addition, we add boolean features indicating whether s_m or t has already been linked in a document.

3.2.4 Reranking

The millions of existing Wikipedia links in $\mathcal{L}$ form a valuable source of training examples for our ranker. However, simply training on the links in $\mathcal{L}$ may result in poor performance, because those links exhibit systematic differences from the mentions in $\mathcal{M}$ that the ranker will be applied to. The reason is that our mention extractor attempts to populate $\mathcal{M}$ with *all* mentions, whereas $\mathcal{L}$ which contains only the specific subset of mentions that meet the hyperlinking conventions of Wikipedia, As a result, the features for $\mathcal{M}$ are distributed differently from those in $\mathcal{L}$, and a model trained on $\mathcal{L}$ may not might not perform well on $\mathcal{M}$. Our strategy is to leverage $\mathcal{L}$ to train an initial ranker, and then hand-label a small set of mentions from $\mathcal{M}$ to train a second-stage *re-ranker* that takes the ranking output of the initial ranker as a feature.

3.3 Linker

Our linker is a binary classifier that decides whether to include (link) each mention in $\mathcal{M}$ to the final output $\mathcal{L}^*$. Previous work has typically used a linker to determine so-called NIL mentions, where the referred-to concept is not in the target knowledge base (e.g., in the TAC KBP competition, half of the given mentions are NIL (Ji and Grishman, 2011)). The purpose of our linker is slightly different, because we also use a linker to control the precision of our output. We use a *probabilistic linker* that predicts a confidence estimate that the mention with its top-ranked candidate is correct. Our linker uses the same features as the ranker and an additional set of confidence signals: the number of times the top candidate concept appears in $\mathcal{L}$, and the score difference between the top-ranked candidate and the second-ranked candidate.

4 Experiments and Result

In this section, we provide an evaluation of our system and its subcomponents.

4.1 Experiment Setup

We trained our initial ranker models from 100,000 randomly selected existing links ($\mathcal{L}$). These links were excluded when building feature values (i.e. the prior probability, or Semantic Relatedness).

We formed an evaluation set of new links by applying our mention extractor to 2,000 randomly selected articles, and then manually labeling 1,900 of the mentions with either the correct concept or "no correct concept." We trained and tested our system on the evaluation set, using 10-fold cross validation. For each fold, we partitioned data

Model	Acc	Prec	Recall	F1
Prior	0.876	0.891	0.850	0.870
OnlyWikiLink −Wiki	0.896	0.905	0.871	0.888
OnlyWikiLink	0.944	0.950	0.920	0.935

Table 1: 10-fold cross validation performance of the initial rankers by Accuracy (excluded $\emptyset$-candidate mentions), BOT Precision, BOT Recall, BOT F1 on the 100,000 existing links.

Model	Acc	Yield	%Yield
Baseline	0.828	5	0.33%
OnlyWikiLink	0.705	150	9.80%
3W−Wiki	0.868	253	16.54%
3W	0.877	365	23.86%

Table 2: 10-fold cross validation performance of the system over 1,900 labeled mentions. Acc is disambiguation accuracy of *solvable* mentions. Yield is the number of output new mentions at precision ≥ 0.98, and %Yield is the percentage of Yield over the *solvable* mentions (recall).

into 3 parts. We used 760 mentions for training the final ranker. The linker was trained with 950 mentions and we tested our system using the other 190 mentions. Previous work has used various ML approaches for ranking, such as SVMs (Dredze et al., 2010). We found logistic regression produces similar accuracy to SVMs, but is faster for our feature set. For the linker, we use an SVM with probabilistic output (Wu et al., 2004; Chang and Lin, 2011) to estimate a confidence score for each output link.

4.2 Result

We first evaluate *3W*'s mention extraction. From the selected $2,000$ articles, the system extracted $59,454$ mentions ($\sim$30/article), in addition to the original $54,309$ links ($\sim$27/article). From the $1,900$ hand-labeled mentions, $1,530$ (80.5%) were *solvable* in that *3W* candidate set contained the correct target.

As described in section 3.2.4, *3W* employs a 2-stage ranker. We first evaluate just the initial ranker, using 10-fold cross validation on 100,000 existing links. We show micro accuracy and bag-of-title (BOT) performance used by Milne and Witten (2008b) in Table 1. The ranker with all features (*OnlyWikiLink*) outperforms the ranker without Wikipedia-specific features (*OnlyWikiLink−Wiki*) by approximately five points in F1. This demonstrates that Wikipedia's rich semantic content is helpful for disambiguation.

Next, we evaluate our full system performance (disambiguation and linking) over the hand-labeled evaluation set. We experimented with different configurations of the rankers and linkers. Our *Baseline* system disambiguates a mention m by selecting the most common concept for the surface $s(m)$. *OnlyWikiLink* uses the ranker model trained on only Wikipedia links, ignoring the labeled mentions. *3W* is our system using all features described in section 3.2,

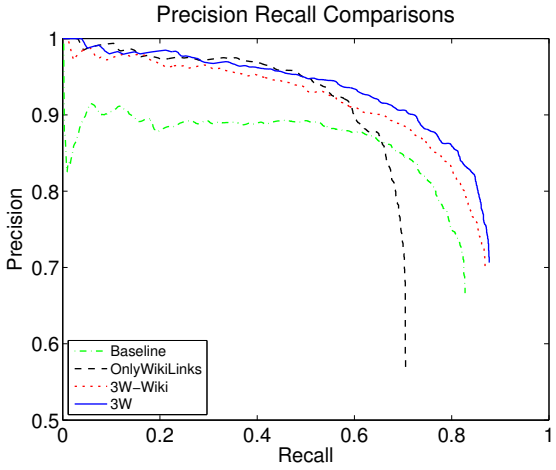

Figure 1: Plot between Precision and Recall of systems on 1,900 mentions from 10-fold cross validation.

and *3W−Wiki* is *3W* without Wikipedia-specific features. The last two configurations are trained using the labeled mentions.

Table 2 shows the disambiguation accuracy of each system over the solvable mentions. Our final system, *3W*, has the best disambiguation accuracy.

To evaluate the linking performance, we select the confidence threshold such that the system outputs mentions with precision of ≥ 0.98. The third column in Table 2 shows the *yield*, i.e. the number of mentions output at precision 0.98. *3W* outputs the largest number of new links (365). Nearly half (157) are new concepts that have not been linked in the source article. We find that the Rerank feature helps increase recall: without it, the yield of *3W* drops by 27%. Using %Yield, we estimate that *3W* will output $14,000$ new links for the selected $2,000$ articles ($\sim$7/article), and approximately 28 million new links across the 4 million articles of English Wikipedia.

Adjusting the confidence threshold allows the system to trade off precision and recall. Figure 1 shows a precision and recall curve. *3W* and *OnlyWikiLink* are comparable for

many high-precision points, but below 0.95 *OnlyWikiLink*'s precision drops quickly. Plots that finish at higher rightmost points in the graph indicate systems that achieve higher accuracy on the complete evaluation set.

5 Conclusions and Future Work

We presented *3W*, a system that adds high-precision links to Wikipedia. Whereas many Wikification systems focus on general text, *3W* is specialized toward Wikipedia articles. We showed that leveraging the link structure of Wikipedia provides advantages in disambiguation. In experiments, *3W* was shown to Wikipedia with $\sim$7 new links per article (an estimated 28m across 4 million Wikipedia articles) at high precision.

Acknowledgments

This work was supported in part by DARPA contract D11AP00268 and the Allen Institute for Artificial Intelligence.

References

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP-CoNLL 2007*, pages 708–716.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Xianpei Han and Jun Zhao. 2009. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 215–224. ACM.

Brent Hecht, Samuel H Carton, Mahmood Quaderi, Johannes Schöning, Martin Raubal, Darren Gergle, and Doug Downey. 2012. Explanatory semantic relatedness and explicit spatialization for exploratory search. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 415–424. ACM.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1148–1158. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. 2013. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

David Milne and Ian H Witten. 2008b. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *ACL*.

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*. Citeseer.

Valentin I. Spitkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for english wikipedia concepts. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Zareen Saba Syed, Tim Finin, and Anupam Joshi. 2008. Wikipedia as an ontology for describing documents. In *ICWSM*.

Fei Wu and Daniel S Weld. 2008. Automatically refining the wikipedia infobox ontology. In *Proceedings of the 17th international conference on World Wide Web*, pages 635–644. ACM.

Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. 2004. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5(975-1005):4.