Resolving Surface Forms to Wikipedia Topics

Yiping Zhou Lan Nie Omid Rouhani-Kalleh Flavian Vasile Scott Gaffney
Yahoo! Labs at Sunnyvale
{zhouy,lannie,omid,flavian,gaffney}@yahoo-inc.com

Abstract

Ambiguity of entity mentions and concept references is a challenge to mining text beyond surface-level keywords. We describe an effective method of disambiguating surface forms and resolving them to Wikipedia entities and concepts. Our method employs an extensive set of features mined from Wikipedia and other large data sources, and combines the features using a machine learning approach with automatically generated training data. Based on a manually labeled evaluation set containing over 1000 news articles, our resolution model has 85% precision and 87.8% recall. The performance is significantly better than three baselines based on traditional context similarities or sense commonness measurements. Our method can be applied to other languages and scales well to new entities and concepts.

1 Introduction

Ambiguity in natural language is prevalent and, as such, it can be a difficult challenge for information retrieval systems and other text mining applications. For example, a search for "*Ford*" in Yahoo! News retrieves about 40 thousand articles containing *Ford* referring to a company (Ford Motors), an athlete (Tommy Ford), a place (Ford City), etc. Due to reference ambiguity, even if we knew the user was only interested in the company, they would still have to contend with articles referring to the other concepts as well.

In this paper we focus on the problem of resolving references of named-entities and concepts in natural language through their textual *surface forms*. Specifically, we present a method of resolving surface forms in general text documents to Wikipedia entries. The tasks of resolution and disambiguation are nearly identical; we make the distinction that resolution specifically applies when a known set of referent concepts are given a priori. Our approach differs from others in multiple aspects including the following.

1) We employ a rich set of disambiguation features leveraging mining results from large-scale data sources. We calculate context-sensitive features by extensively mining the categories, links and contents of the entire Wikipedia corpus. Additionally we make use of context-independent data mined from various data sources including Web user-behavioral data and Wikipedia. Our features also capture the one-to-one relationship between a surface form and its referent.

2) We use machine learning methods to train resolution models with a large automatically labeled training set. Both ranking-based and classification-based resolution approaches are explored.

3) Our method disambiguates both entities and word senses. It scales well to new entities and concepts, and it can be easily applied to other languages.

We propose an extensive set of metrics to evaluate not only overall resolution performance but also out-of-Wikipedia prediction. Our systems for English language are evaluated using real-world test sets and compared with a number of baselines. Evaluation results show that our systems consistently and significantly outperform others across all test sets.

The paper is organized as follows. We first describe related research in Section 2, followed by an introduction of Wikipedia in Section 3. We then introduce our learning method in Section 4 and our features in Section 5. We show our experimental results in Section 6, and finally close with a discussion of future work.

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1335–1343,
Beijing, August 2010

2 Related Work

Named entity disambiguation research can be divided into two categories: some works (Bagga and Baldwin, 1998; Mann and Yarowsky, 2003; Pedersen et al., 2005; Fleischman and Hovy, 2004; Ravin and Kazi, 1999) aim to cluster ambiguous surface forms to different groups, with each representing a unique entity; others (Cucerzan, 2007; Bunescu and Paşca, 2006; Han and Zhao, 2009; Milne and Witten, 2008a; Milne and Witten, 2008b) resolve a surface form to an entity or concept extracted from existing knowledge bases. Our work falls into the second category.

Looking specifically at resolution, Bunescu and Pasca (2006) built a taxonomy SVM kernel to enrich a surface form's representation with words from Wikipedia articles in the same category. Cucerzan (2007) employed context vectors consisting of phrases and categories extracted from Wikipedia. The system also attempted to disambiguate all surface forms in a context simultaneously, with the constraint that their resolved entities should be globally consistent on the category level as much as possible. Milne and Witten (2008a, 2008b) proposed to use Wikipedia's link structure to capture the *relatedness* between Wikipedia entities so that a surface form is resolved to an entity based on its relatedness to the surface form's surrounding entities. Besides relatedness, they also define a *commonness* feature that captures how common it is that a surface form links to a particular entity in general. Han and Zhao (2009) defined a novel alignment strategy to calculate similarity between surface forms based on semantic relatedness in the context.

Milne and Witten's work is most related to what we propose here in that we also employ features similar to their relatedness and commonness features. However, we add to this a much richer set of features which are extracted from Web-scale data sources beyond Wikipedia, and we develop a machine learning approach to automatically blend our features using completely automatically generated training data.

3 Wikipedia

Wikipedia has more than 200 language editions, and the English edition has more than 3 million articles as of March 2009. Newsworthy events are often added to Wikipedia within days of occurrence; Wikipedia has bi-weekly snapshots available for download.

Each article in Wikipedia is uniquely identified by its title which is usually the most common surface form of an entity or concept. Each article includes body text, outgoing links and categories. Here is a sample sentence in the article titled "Aristotle" in wikitext format. "*Together with Plato and [[Socrates]] (Plato's teacher), Aristotle is one of the most important founding figures in [[Western philosophy]].*" Near the end of the article, there are category links such as "*[[Category:Ancient Greek mathematicians]]*". The double brackets annotate outgoing links to other Wikipedia articles with the specified titles. The category names are created by authors. Articles and category names have many-to-many relationships.

In addition to normal articles, Wikipedia also has special types of articles such as redirect articles and disambiguation articles. A redirect article's title is an alternative surface form for a Wikipedia entry. A disambiguation article lists links to similarly named articles, and usually its title is a commonly used surface form for multiple entities and concepts.

4 Method of Learning

Our goal is to resolve surface forms to entities or concepts described in Wikipedia. To this end, we first need a recognizer to detect surface forms to be resolved. Then we need a resolver to map a surface form to the most probable entry in Wikipedia (or to *out-of-wiki*) based on the context.

Recognizer: We first create a set of Wikipedia (article) entries $E = \{e_1, e_2, ...\}$ to which we want to resolve surface forms. Each entry's surface forms are mined from multiple data sources. Then we use simple string match to recognize surface forms from text documents.

Among all Wikipedia entries, we exclude those with low importance. In our experiments, we removed the entries that would not interest general Web users, such as stop words and punctuations. Second, we collect surface forms for entries in E using Wikipedia and Web search query click logs based on the following assumptions:

- Each Wikipedia article title is a surface form for the entry. Redirect titles are taken as alternative surface forms for the target entry.
- The anchor text of a link from one article to another is taken as an alternative surface form for the linked-to entry.
- Web search engine queries resulting in user clicks on a Wikipedia article are taken as alternative surface forms for the entry.

As a result, we get a number of surface forms for each entry e_i. If we let s_{ij} denote the j-th surface form for entry i, then we can represent our entry dictionary as $EntSfDict = \{<e_1, (s_{11}, s_{12}, ...)>, <e_2, (s_{21}, s_{22}, ...)>, ...\}$.

Resolver: We first build a labeled training set automatically, and then use supervised learning methods to learn models to resolve among Wikipedia entries. In the rest of this section we describe the resolver in details.

4.1 Automatically Labeled Data

To learn accurate models, supervised learning methods require training data with both large quantity and high quality, which often takes lots of human labeling effort. However, in Wikipedia, links provide a *supervised* mapping from surface forms to article entries. We use these links to automatically generate training data. If a link's anchor text is a surface form in *EntSfDict*, we extract the anchor text as surface form *s* and the link's destination article as Wikipedia entry *e*, then add the pair *(s, e)* with a positive judgment to our labeled example set. Continuing, we use *EntSfDict* to find other Wikipedia entries for which *s* is a surface form and create negative examples for these and add them to our labeled example set. If *e* does not exist in *EntSfDict* (for example, if the link points to a Wikipedia article about a stop word), then a negative training example is created for every Wikipedia entry to which *s* may resolve. We use *oow* (out-of-wiki) to denote this case.

Instead of article level coreference resolution, we only match partial names with full names based on the observation that surface forms for named entities are usually capitalized word sequences in English language and a named entity is often mentioned by a long surface form followed by mentions of short forms in the same article. For each pair *(s, e)* in the labeled example set, if *s* is a partial name of a full name *s'* occurring earlier in the same document, we replace *(s, e)* with *(s', e)* in the labeled example set.

Using this methodology we created 2.4 million labeled examples from only 1% of English Wikipedia articles. The abundance of data made it possible for us to experiment on the impact of training set size on model accuracy.

4.2 Learning Algorithms

In our experiments we explored both Gradient Boosted Decision Trees (GBDT) and Gradient Boosted Ranking (GBRank) to learn resolution models. They both can easily combine features of different scale and with missing values. Other supervised learning methods are to be explored in the future.

GBDT: We use the stochastic variant of GBDTs (Friedman, 2001) to learn a binary logistic regression model with the judgments as the target. GBDTs compute a function approximation by performing a numerical optimization in the function space. It is done in multiple stages, with each stage modeling residuals from the model of the last stage using a small decision tree. A brief summary is given in Algorithm 1. In the stochastic version of GBDT, one sub-samples the training data instead of using the entire training set to compute the loss function.

Algorithm 1 GBDTs

```
Input: training data $\{(x_i, y_i)\}_{i=1}^N$, loss function
$L[y, f(x)]$, the number of nodes for each tree $J$
, the number of trees $M$.
1:    Initialize $f(x)=f^0$
2:    For $m = 1$ to $M$
2.1:    For $i = 1$ to $N$, compute the negative
        gradient by taking the derivative of the
        loss with respect to $f(x)$ and substitute
        with $y_i$ and $f_i^{m-1}(x_i)$.
2.2:    Fit a $J$-node regression tree to the
        components of the negative gradient.
2.3:    Find the within-node updates $a_j^m$ for $j$
        $= 1$ to $J$ by performing $J$ univariate optimizations of the node contributions to
        the estimated loss.
2.4:    Do the update $f_i^m(x_i) = f_i^{m-1}(x_i) + r \times a_j^m$,
        where $j$ is the node that $x_i$ belongs to, $r$
        is learning rate.
3:    End for
4:    Return $f^M$
```

In our setting, the loss function is a negative binomial log-likelihood, x_i is the feature vector for a surface-form and Wikipedia-entry pair *(s_i, e_i)*, and y_i is +1 for positive judgments and -1 is for negative judgments.

GBRank: From a given surface form's judgments we can infer that the correct Wikipedia entry is preferred over other entries. This allows us to derive pair-wise preference judgments from absolute judgments and train a model to rank all the Wikipedia candidate entries for each surface form. Let $S = \{(x_i, x_i') \mid l(x_i) \ge l(x_i'), i = 1, ..., N\}$ be the set of preference judgments, where x_i and x_i' are the feature vectors for two pairs of surface-forms and Wikipedia-entry, $l(x_i)$ and $l(x_i')$ are their absolute judgments respectively. GBRank (Zheng et al., 2007) tries to learn a function h such that $h(x_i) \ge h(x_i')$ for $(x_i, x_i') \in S$. A sketch of the algorithm is given in Algorithm 2.

Algorithm 2 GBRank

```
1:    Initialize $h=h_0$
2:    For $k=1$ to $K$
2.1:      Use $h_{k-1}$ as an approximation of $h$ and
          compute
            $S^+ = \{(x_i, x_i') \in S \mid h_{k-1}(x_i) \ge h_{k-1}(x_i') + \tau\}$
            $S^- = \{(x_i, x_i') \in S \mid h_{k-1}(x_i) < h_{k-1}(x_i') + \tau\}$
          where $\tau = \alpha(l(x_i) - l(x_i'))$
2.2:      Fit a regression function $g_k$ using
          GBDT and the incorrectly predicted
          examples
            $\{(x_i, h_{k-1}(x_i') + \tau), (x_i', h_{k-1}(x_i) - \tau) \mid (x_i, x_i') \in S^-\}$
2.3:      Do the update
            $h_k(x) = (kh_{k-1}(x) + \eta g_k(x))/(k+1)$, where $\eta$ is
          learning rate.
3:    End for
4:    Return $h_K$
```

We use a tuning set independent from the training set to select the optimal parameters for GBDT and GBRank. This includes the number of trees M, the number of nodes J, the learning rate r, and the sampling rate for GBDT; and for GBRank we select K, α and η.

The feature importance measurement given by GBDT and GBRank is computed by keeping track of the reduction in the loss function at each feature variable split and then computing the total reduction of loss along each explanatory feature variable. We use it to analyze feature effectiveness.

4.3 Prediction

After applying a resolution model on the given test data, we obtain a score for each surface-form and Wikipedia-entry pair *(s, e)*. Among all the pairs containing s, we find the pair with the highest score, denoted by *(s, $\tilde{e}$)*.

It's very common that a surface form refers to an entity or concept not defined in Wikipedia. So it's important to correctly predict whether the given surface form cannot be mapped to any Wikipedia entry in *EntSfDict*.

We apply a threshold to the scores from resolution models. If the score for *(s, $\tilde{e}$)* is lower than the threshold, then the prediction is *oow* (see Section 4.1), otherwise $\tilde{e}$ is predicted to be the entry referred by s. We select thresholds based on F1 (see Section 6.2) on a tuning set that is independent from our training set and test set.

5 Features

For each surface-form and Wikipedia-entry pair *(s, e)*, we create a feature vector including features capturing the context surrounding s and features independent of the context. They are context-dependent and context-independent features respectively. Various data sources are mined to extract these features, including Wikipedia articles, Web search query-click logs, and Web-user browsing logs. In addition, *(s, e)* is compared to all pairs containing s based on above features and the derived features are called differentiation features.

5.1 Context-dependent Features

These features measure whether the given surface form s resolving to the given Wikipedia entry e would make the given document more coherent. They are based on 1) the vector representation of e, and 2) the vector representation of the context of s in a document d.

Representation of *e*: By thoroughly mining Wikipedia and other large data sources we extract contextual clues for each Wikipedia entry e and formulate its representation in the following ways.

1) *Background representation*. The overall background description of e is given in the corresponding Wikipedia article, denoted as A_e. Naturally, a bag of terms and surface forms in A_e can represent e. So we represent e by a back-

ground word vector E_{bw} and a background surface form vector E_{bs}, in which each element is the occurrence count of a word or a surface form in A_e's first paragraph.

2) *Co-occurrence representation.* The terms and surface forms frequently co-occurring with *e* capture its contextual characteristics. We first identify all the Wikipedia articles linking to A_e. Then, for each link pointing to A_e we extract the surrounding words and surface forms within a window centered on the anchor text. The window size is set to 10 words in our experiment. Finally, we select the words and surface forms with the top co-occurrence frequency, and represent *e* by a co-occurring word vector E_{cw} and a co-occurring surface form vector E_{cs}, in which each element is the co-occurrence frequency of a selected word or surface form.

3) *Relatedness representation*. We analyzed the relatedness between Wikipedia entries from different data sources using various measurements, and we computed over 20 types of relatedness scores in our experiments. In the following we discuss three types as examples. The first type is computed based on the overlap between two Wikipedia entries' categories. The second type is mined from Wikipedia inter-article links. (In our experiments, two Wikipedia entries are considered to be related if the two articles are mutually linked to each other or co-cited by many Wikipedia articles.) The third type is mined from Web-user browsing data based on the assumption that two Wikipedia articles co-occurring in the same browsing session are related. We used approximately one year of Yahoo! user data in our experiments. A number of different metrics are used to measure the relatedness. For example, we apply the algorithm of Google distance (Milne and Witten, 2008b) on Wikipedia links to calculate the Wikipedia link-based relatedness, and use mutual information for the browsing-session-based relatedness. In summary, we represent *e* by a related entry vector E_r for each type of relatedness, in which each element is the relatedness score between *e* and a related entry.

Representation of *s*: We represent a surface form's context as a vector, then calculate a context-dependent feature for a pair <*s,e*> by a similarity function *Sim* from two vectors. Here are examples of context representation.

1) *s* is represented by a word vector S_w and a surface form vector S_s, in which each element is the occurrence count of a word or a surface form surrounding *s.* We calculate each vector's similarity with the background and co-occurrence representation of *e*, and it results in $Sim(S_w, E_{bw})$, $Sim(S_w, E_{cw})$, $Sim(S_s, E_{bs})$ and $Sim(S_s, E_{cs})$.

2) *s* is represented by a Wikipedia entry vector S_e, in which each element is a Wikipedia entry to which a surrounding surface form *s* could resolve. We calculate its similarity with the relatedness representation of *e*, and it results in $Sim(S_e, E_r)$.

In the above description, similarity is calculated by dot product or in a summation-of-maximum fashion. In our experiments we extracted surrounding words and surface forms for *s* from the whole document or from the text window of 55 tokens centered on *s*, which resulted in 2 sets of features. We created around 50 context-dependent features in total.

5.2 Context-independent Features

These features are extracted from data beyond the document containing s. Here are examples.

- During the process of building the dictionary *EntSfDict* as described in Section 4, we count how often *s* maps to *e* and estimate the probability of *s* mapping to *e* for each data source. These are the commonness features.
- The number of Wikipedia entries that *s* could map to is a feature about the ambiguity of *s*.
- The string similarity between *s* and the title of A_e is used as a feature. In our experiments string similarity was based on word overlap.

5.3 Differentiation Features

Among all surface-form and Wikipedia-entry pairs that contain *s*, at most one pair gets the positive judgment. Based on this observation we created differentiation features to represent how *(s, e)* is compared to other pairs for *s*. They are derived from the context-dependent and context-independent features described above. For example, we compute the difference between the string similarity for *(s, e)* and the maximum string similarity for all pairs containing *s*. The derived feature value would be zero if *(s, e)* has larger string similarity than other pairs containing *s*.

6 Experimental Results

In our experiments we used the Wikipedia snapshot for March 6th, 2009. Our dictionary *EntSfDict* contains 3.5 million Wikipedia entries and 6.5 million surface forms.

A training set was created from randomly selected Wikipedia articles using the process described in Section 4.1. We varied the number of Wikipedia articles from 500 to 40,000, but the performance did not increase much after 5000. The experimental results reported in this paper are based on the training set generated from 5000 articles. It contains around 1.4 million training examples. There are approximately 300,000 surface forms, out of which 28,000 are the *oow* case.

Around 400 features were created in total, and 200 of them were selected by GBDT and GBRank to be used in our resolution models.

6.1 Evaluation Datasets

Three datasets from different data sources are used in evaluation.

1) *Wikipedia hold-out set*. Using the same process for generating training data and excluding the surface forms appearing in the training data, we built the hold-out set from approximately 15,000 Wikipedia articles, containing around 600,000 labeled instances. There are 400,000 surface forms, out of which 46,000 do not resolve to any Wikipedia entry.

2) *MSNBC News test set*. This entity disambiguation data set was introduced by Cucerzan (2007). It contains 200 news articles collected from ten MSNBC news categories as of January 2, 2007. Surface forms were manually identified and mapped to Wikipedia entities. The data set contains 756 surface forms. Only 589 of them are contained in our dictionary *EntSfDict*, mainly because *EntSfDict* excludes surface forms of out-of-Wikipedia entities and concepts. Since the evaluation task is focused on resolution performance rather than recognition, we exclude the missing surface forms from the labeled example set. The final dataset contains 4,151 labeled instances. There are 589 surface forms and 40 of them do not resolve to any Wikipedia entry.

3) *Yahoo! News set*. One limitation of the MSNBC test set is the small size. We built a much larger data set by randomly sampling around 1,000 news articles from Yahoo! News over 2008 and had them manually annotated. The experts first identified *person*, *location* and *organization* names, then mapped each name to a Wikipedia article if the article is about the entity referred to by the name. We didn't include more general concepts in this data set to make the manual effort easier. This data set contains around 100,000 labeled instances. The data set includes 15,387 surface forms and 3,532 of them cannot be resolved to any Wikipedia entity. We randomly split the data set to 2 parts of equal size. One part is used to tune parameters of GBDT and GBRank and select thresholds based on F1 value. The evaluation results presented in this paper is based on the remaining part of the Yahoo! News set.

6.2 Metrics

The possible outcomes from comparing a resolution system's prediction with ground truth can be categorized into the following types.

- True Positive (TP), the predicted *e* was correctly referred to by *s*.
- True Negative (TN), *s* was correctly predicted as resolving to *oow*.
- Mismatch (MM), the predicted *e* was not correctly referred to by *s* and should have been *e'* from *EntSfDict*.
- False Positive (FP), the predicted *e* was not correctly referred to by *s* and should have been *oow*.
- False Negative (FN), the predicted *oow* is not correct and should have been *e'* from *EntSfDict*.

Similar to the widely used metrics for classification systems, we use following metrics to evaluate disambiguation performance.

$$precision = \frac{TP}{TP + FP + MM} \qquad recall = \frac{TP}{TP + FN + MM}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \qquad accuracy = \frac{TP + TN}{TP + FP + TN + FN + MM}$$

In the Yahoo! News test set, 23.5% of the surface forms do not resolve to any Wikipedia entries, and in the other two test sets the percentages of *oow* are between 10% and 20%. This demonstrates it is necessary in real-world applications to explicitly measure *oow* prediction. We propose following metrics.

$$precision\_oow = \frac{TN}{TN + FN} \qquad recall\_oow = \frac{TN}{TN + FP}$$

$$F1\_oow = \frac{2 \times precision\_oow \times recall\_oow}{precision\_oow + recall\_oow}$$

6.3 Evaluation Results

With our training set we trained one resolution model using GBDT (named as *WikiRes-c*) and another resolution model using GBRank (named as *WikiRes-r*). The models were evaluated along with the following systems.

1) *Baseline-r*: each surface form *s* is randomly mapped to *oow* or a candidate entry for *s* in *EntSfDict*.

2) *Baseline-p*: each surface form *s* is mapped to the candidate entry *e* for *s* with the highest commonness score. The commonness score is linear combination of the probability of *s* being mapped to *e* estimated from different data sources. The commonness score is among the features used in *WikiRes-c* and *WikiRes-r*.

3) *Baseline-m*: we implemented the approach brought by Cucerzan (2007) based on our best understanding. Since we use a different version of Wikipedia and a different entity recognition approach, the evaluation result differs from the result presented in their paper. But we believe our implementation follows the algorithm described in their paper.

In Table 1 we present the performance for each system on the Yahoo! News test set and the MSNBC test set. The performance of *WikiRes-c* and *WikiRes-r* are computed after we apply the thresholds selected on the tuning set described in Section 6.1. In the upper half of Table 1, the three baselines use the thresholds that lead to the best F1 on the Yahoo! News test set. In the lower half of Table 1, the three baselines use the thresholds that lead to the best F1 on the MSNBC test set.

Among the three baselines, *Baseline-r* has the lowest performance. *Baseline-m* uses a few context-sensitive features and *Baseline-p* uses a context-independent feature. These two types of features are both useful, but *Baseline-p* shows better performance, probably because the surface forms in our test sets are dominated by common senses. In our resolution models, these features are combined together with many other features calculated from different large-scale data sources and on different granularity levels. As shown in Table 1, both of our resolution solutions substantially outperform other systems. Furthermore, *WikiRes-c* and *WikiRes-r* have similar performance.

	Precision	**Recall**	**F1**	**Accuracy**	**p-value**
Yahoo! News Test Set					
Baseline-r	47.023	60.831	53.043	47.023	0
Baseline-p	73.869	88.157	80.383	73.175	5.2e-78
Baseline-m	62.240	80.517	70.208	62.240	1.3e-160
WikiRes-r	83.406	**88.858**	86.046	80.717	0.012
WikiRes-c	**85.038**	87.831	**86.412**	**81.463**	---
MSNBC Test Set					
Baseline-r	60.272	64.545	62.335	60.272	8.9e-19
Baseline-p	82.292	86.182	84.192	82.003	0.306
Baseline-m	78.947	84.545	81.651	78.947	0.05
WikiRes-r	**88.785**	**86.364**	**87.558**	**84.550**	0.102
WikiRes-c	88.658	85.273	86.932	83.192	---

Table 1. Performance on the Yahoo! News Test Set and the MSNBC Test set

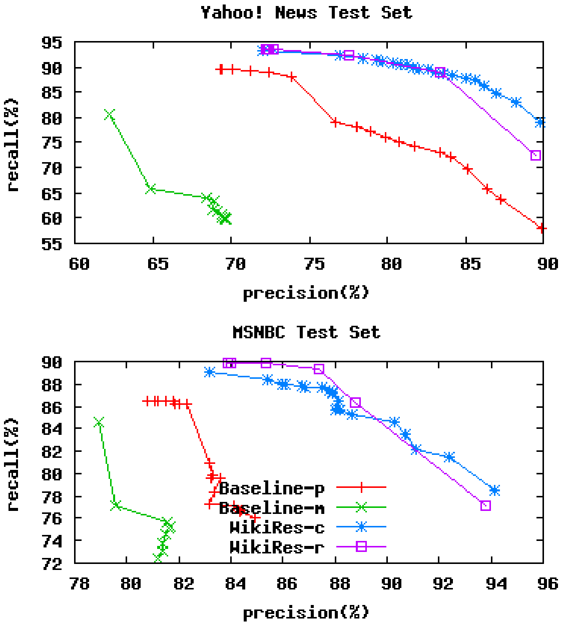

Figure 1. Precision-recall on the Yahoo! News Test Set and the MSNBC Test Set

We compared *WikiRes-c* with each competitor and from the statistical significance test results in the last column of Table 1 we see that on the Yahoo! News test set *WikiRes-c* significantly outperforms others. The p-values for the MSNBC test set are much higher than for the Yahoo! News test set because the MSNBC test set is much smaller.

Attempting to address this point, we see that the F1 values of *WikiRes* on the MSNBC test set and on the Yahoo! News test set only differs by a couple percentage points, although, these test sets were created independently. This suggests the objectivity of our method for creating the Yahoo! News test set and provides a way to measure resolution model performance on what would occur in a general news corpus in a statistically significant manner.

In Figure 1 we present the precision-recall curves on the Yahoo! News and the MSNBC test sets. We see that our resolution models are substantially better than the other two baselines at any particular precision or recall value on both test sets. *Baseline-r* is not included in the comparison since it does not have the tradeoff between precision and recall. We find the precision-recall curve of *WikiRes-r* is very similar to *WikiRes-c* at the lower precision area, but its recall is much lower than other systems after precision reaches around 90%. So, in Figure 1 the curves of *WikiRes-r* are truncated at the high precision area.

In Table 2 we compare the performance of out-of-Wikipedia prediction. The comparison is done on the Yahoo! News test set only, since there are only 40 surface forms of *oow* case in the MSNBC test set. Each system's threshold is the same as that used for the upper half of Table 1. The results show our models have substantially higher precision and recall than *Baseline-p* and *Baseline-m*. From the statistical significance test results in the last column, we can see that *WikiRes-c* significantly outperforms *Baseline-p* and *Baseline-m*. Also, our current approaches still have room to improve in the area of out-of-Wikipedia prediction.

We also evaluated our models on a Wikipedia hold-out set. The model performance is greater than that obtained from the previous two test sets because the hold-out set is more similar to the training data source itself. Again, our models perform better than others.

From the feature importance lists of our GBDT model and GBRank model, we find that the commonness features, the features based on Wikipedia entries' co-occurrence representation and the corresponding differentiation features are the most important.

	Precision	Recall	F1	p-value
Baseline-p	64.907	22.152	33.03	1.6e-20
Baseline-m	47.207	44.78	45.961	1.3e-34
WikiRes-r	**68.166**	52.994	59.630	0.084
WikiRes-c	67.303	**59.777**	**63.317**	---

Table 2. Performance of Out-of-Wikipedia Prediction on the Yahoo! News Test Set

7 Conclusions

We have described a method of learning to resolve surface forms to Wikipedia entries. Using this method we can enrich the unstructured documents with structured knowledge from Wikipedia, the largest knowledge base in existence. The enrichment makes it possible to represent a document as a machine-readable network of senses instead of just a bag of words. This can supply critical semantic information useful for next-generation information retrieval systems and other text mining applications.

Our resolution models use an extensive set of novel features and are leveraged by a machine learned approach that depends only on a purely automated training data generation facility. Our methodology can be applied to any other language that has Wikipedia and Web data available (after modifying the simple capitalization rules in Section 4.1). Our resolution models can be easily and quickly retrained with updated data when Wikipedia and the relevant Web data are changed.

For future work, it will be important to investigate other approaches to better predict *oow*. Adding global constraints on resolutions of the same term at multiple locations in the same document may also be important. Of course, developing new features (such as part-of-speech, named entity type, etc) and improving training data quality is always critical, especially for social content sources such as those from Twitter. Finally, directly demonstrating the degree of applicability to other languages is interesting when accounting for the fact that the quality of Wikipedia is variable across languages.

References

Bagga, Amit and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the Vector Space Model. *Proceedings of the 17th international conference on Computational linguistics*.

Bunescu, Razvan and Marius Paşca. 2006. Using Encyclopedic Knowledge for Named Entity Disambiguation. *Proceedings of the 11th Conference of the European Chapter of the Association of Computational Linguistics (EACL-2006)*.

Cucerzan, Silviu. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. *Pro-*

ceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.

Fleischman, Ben Michael and Eduard Hovy. 2004. Multi-Document Person Name Resolution. *Proceesing of the Association for Computational Linguistics*.

Friedman, J. H. 2001. Stochastic gradient boosting. *Computational Statistics and Data Analysis,* 38:367–378.

Han, Xianpei and Jun Zhao 2009. Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.*

Mann, S. Gidon and David Yarowsky. 2003. Unsupervised Personal Name Disambiguation. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*.

Milne, David and Ian H. Witten. 2008a. Learning to Link with Wikipedia. *In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM'2008)*.

Milne, David and Ian H. Witten. 2008b. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence*.

Pedersen, Ted, Amruta Purandare and Anagha Kulkarni. 2005. Name Discrimination by Clustering Similar Contexts. *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (2005)*.

Ravin, Y. and Z. Kazi. 1999. Is Hillary Rodham Clinton the President? In *Association for Computational Linguistics Workshop on Coreference and its Applications*.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics,* pages 189-196.

Zheng, Zhaohui, K. Chen, G. Sun, and H. Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 287-294.