Plato: A Selective Context Model for Entity Resolution

Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, Fernando Pereira
Google Inc., Mountain View CA 94043, USA
{nevena, asubram, ringgaard, pereira}@google.com

Abstract

We present Plato, a probabilistic model for entity resolution that includes a novel approach for handling noisy or uninformative features, and supplements labeled training data derived from Wikipedia with a very large unlabeled text corpus. Training and inference in the proposed model can easily be distributed across many servers, allowing it to scale to over 10^7 entities. We evaluate Plato on three standard datasets for entity resolution. Our approach achieves the best results to-date on TAC KBP 2011 and is highly competitive on both the CoNLL 2003 and TAC KBP 2012 datasets.

1 Introduction

Given a document collection and a knowledge base (KB) of entities, *entity resolution*, also known as *entity disambiguation* or *entity linking*, is the process of mapping each *entity mention* in a document to the corresponding entity record in the KB (Bunescu and Pasca, 2006; Cucerzan, 2007; Dredze et al., 2010; Hachey et al., 2013).

Entity resolution is challenging because referring expressions are often ambiguous on their own and can only be disambiguated by their surrounding context. Consider the name `Newcastle`; it can refer to the city of Newcastle upon Tyne in UK, to the football (soccer for US readers) club Newcastle United F.C., to a popular beverage (Newcastle Brown Ale), or to several other entities. The ambiguity can only be resolved with appropriate context. Another complicating factor is that no KB is complete, and so a name in a document may refer to an entity that is missing from the KB. This problem is commonly called NIL *detection*.

In this paper we present a probabilistic model for entity resolution. Our system, hereafter referred to as Plato, is designed to be resilient to irrelevant features and can be seen as a *selective* extension of the naïve Bayes model. Specifically, we assume that most of the context features of a mention are irrelevant to its disambiguation. This contrasts with the naïve Bayes assumption that all features are generated from a class-conditional distribution and are thus all relevant to the class assignment. Our empirical results support this modeling choice. We train Plato in a semi-supervised manner, starting with labeled data derived from Wikipedia, and continuing with a very large unlabeled corpus of Web documents. The use of unlabeled data enables us to obtain a better estimate of feature distributions and discover new features that are not present in the (labeled) training data. Plato scales up easily to very large KBs with millions of entities and includes NIL detection as a natural by-product of inference. We named our system after the Greek philosopher because the system's inference of real underlying entities from imperfect evidence reminds us of Plato's Theory of Forms.

Previous entity resolution studies (Milne and Witten, 2008; Cucerzan, 2007; Ratinov et al., 2011; Hoffart et al., 2011; Hachey et al., 2013) have typically relied on three main components: a *mention model*, a *context model*, and a *coherency model*. The mention model, perhaps the most important component (Hachey et al., 2013), estimates the prior belief that a particular phrase refers to a particular entity

Transactions of the Association for Computational Linguistics, vol. 3, pp. 503–515, 2015. Action Editor: Noah Smith.
Submission batch: 4/2015; Revision batch: 6/2015; Published 9/2015.

in the KB. In addition to providing a prior, the mention model also helps efficient inference by giving zero probability to entities that are never referred to by a particular name. The context model helps disambiguate the entity using the textual context of the mention. This includes both features extracted from the immediate context (such as the enclosing sentence) and from the overall discourse (such as the most salient noun phrases in the document). Finally, the coherency model encourages all referring expressions in a document to resolve to entities that are related to each other in the KB. For example, a mention of Sunderland A.F.C. (a rival football club to Newcastle United F.C.) may reduce the uncertainty about the mention `Newcastle`. Since a coherency model introduces dependencies between the resolutions of all the mentions in a document, it is seen as a *global* model, while mention and context models are usually referred to as *local* (Ratinov et al., 2011). Coherency models typically have an increased inference cost, as they require access to the relevant entity relationships in the KB.

Plato does *not* include a full coherency component. Instead, mentions in a given document are sorted into coreference clusters by a simple within-document coreference algorithm similar to that of Haghighi and Klein (2009). Each coreference cluster is then resolved to the KB independently of the resolution of the other clusters in the document. The context features for each mention cluster in our model include the names of other referring phrases in the document. Since many referring phrases are unambiguous, our hypothesis is that such context can capture much of the discourse coherence usually represented by a coherency model. Plato detects and links both nominal and named mentions, but following previous work, we evaluate it only on the resolution of gold *named entity* mentions to either KB or NIL.

We train Plato with expectation-maximization (EM), which is easily parallelizable and thus can scale up to very large KBs and unlabeled training corpora. Indeed, our efficient distributed implementation allows the system to scale up to KBs with over 10^7 entities. Plato achieves highly competitive results on several benchmarks: CoNLL 2003, TAC 2012 KBP, and TAC 2011 KBP. Most importantly, this performance is "out-of-the-box": we did *not* use any of the corresponding training sets, labeled or not, to train the model or tune hyperparameters.

2 Definitions and Notation

We are given a document collection where all the candidate entity mentions have been identified. Each mention is characterized by its phrase, and by the document context. The context is abstracted as a (multi)set of features that includes phrases related to the mention by linear order, syntax, or within-document coreference and phrases extracted from the whole enclosing document. Context features for a mention depend only on the document text, and not on the results of the entity resolution for other mentions. Therefore, we can treat each mention (more strictly speaking, each coreference cluster) independently. Finally, we are given the set $\mathcal{E}$ of KB entities.

When discussing probabilistic models, we use uppercase for random variables, lowercase for the values they take, and boldface for vectors. We use w_m to represent the phrase of mention m. The context of mention m is represented either as a binary feature presence vector $\mathbf{b}_m \in \{0,1\}^{|\mathcal{F}|}$, or as a feature count vector $\mathbf{c}_m \in \mathbb{N}^{|\mathcal{F}|}$. The random variable $E_m \in \mathcal{E}$ ranges over the possible candidate entities for mention m. We provide more details on how the candidates are obtained in Section 8.

3 Naïve Bayes Model

We start with a naïve Bayes model that will serve as a source of intuition and an evaluation baseline. In this model, the phrase and context of a mention are conditionally independent given the entity:

$$p(e|\mathbf{c}, w) \propto p(e)p(w|e)p(\mathbf{c}|e). \quad (1)$$

If we assume that the set of feature counts is drawn from the multinomial distribution then

$$p(\mathbf{c}|e, \boldsymbol{\theta}_e) = \frac{(\sum_k c_k)!}{c_1! \ldots c_{|\mathcal{F}|}!} \prod_k {\theta_{e,k}}^{c_k},$$

where $\theta_{e,k}$ is the probability of drawing feature k given that the entity is e. The posterior probability of an entity given the context feature counts $\mathbf{c}$ and the mention phrase w is

$$p(e|\mathbf{c}, w) \propto p(e)p(w|e) \prod_k {\theta_{e,k}}^{c_k}.$$

At first sight, the naïve Bayes model seems well suited for the entity resolution problem. It is very simple to implement. Given labeled data, the maximum likelihood (ML) estimate of the parameters $\boldsymbol{\theta}_e$ can be obtained from data counts. Unlabeled training data can be incorporated using the EM algorithm, which lends itself to easy parallelization.

We implemented a naïve Bayes model and used it for resolving entities in the CoNLL corpus (Hoffart et al., 2011) and the TAC KBP corpora (Ji et al., 2011; Mayfield et al., 2012), as discussed in more detail in Section 9. We found that the performance of the model was only slightly better than using only the mention phrase. We hypothesize that for the more difficult cases in the test set, many context features of a mention are not informative in identifying the entity, contrary to the model assumption that *all* context features are drawn from an entity-conditional distribution. Consider the following example:

> *While Solon may have slipped slightly this year in Cleveland magazine's ranking of best suburbs, it climbed higher on a more prestigious list. On Monday, the city placed 23rd on Money magazine's annual list of best places to live in the United States.*

There are five US locations named `Solon` in Wikipedia (in addition to the pre-Socratic Athenian statesman). In the above, `Solon` refers to a suburb of Cleveland, Ohio. The only context feature that helps us discriminate between the different possible disambiguations is `Cleveland`; the remaining features (such as `Money magazine`, `United States`) could easily appear in the context of the other candidates. Thus, combining the evidence from all features may blur the distinction between these entities. Our *selective context* model aims to address this issue.

4 Selective context model

Our selective context model assumes that most features that appear in the context of a mention are not discriminative for entity disambiguation. In particular, we make a simplifying modeling assumption that exactly one context feature is relevant for disambiguation, and the remaining features are drawn from a background distribution. Let K be the random variable corresponding to the index of the relevant feature for a mention of entity e. The model can be written as

$$p(k, e, w, \mathbf{b}) = p(w)p(e|w)p(k|e)\prod_j p(b_j|k)$$

$$p(b_j|k) = \begin{cases} b_j & \text{if } k = j \\ \beta_j^{b_j}(1-\beta_j)^{1-b_j} & \text{if } k \neq j. \end{cases}$$

Here β_j parameterizes the background probability of observing a feature that is not relevant. We impose the constraint that the relevant k^{th} feature must be on, and hence $p(b_k|K=k) = b_k$. We treat mentions (or mention clusters) as independent and identically distributed; the complete model is shown in Figure 1.

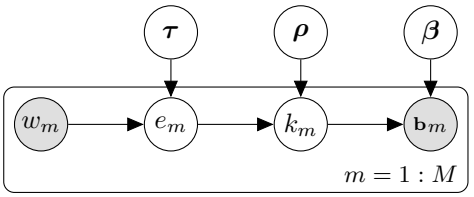

Figure 1: Selective context model. The mention phrase w_m provides a prior distribution over possible entities for mention m. The latent variable k_m selects a relevant context feature b_{m,k_m} that fires for entity e_m; the remaining features are drawn from a background distribution $p(b_j|\beta_j)$. The entity is represented as a latent variable here, but it is observed for the labeled training mentions.

Given a test mention $(w', \mathbf{b}')$, we can compute the entity posterior by marginalizing out the latent variable k:

$$p(e|\mathbf{b}', w') \propto p(e|w')\sum_k p(k|e)b'_k \prod_{j\neq k} p(b'_j|K \neq j)$$

$$\propto p(e|w')\sum_k b'_k \frac{p(k|e)}{\beta_k}. \quad (2)$$

Thus the entity posterior is a product of the name score $(p(e|w'))$ and context score $(\sum_k b'_k p(k|e)/\beta_k)$. This is intuitively appealing: if we assume that $\beta_k \approx p(k)$ then $p(k|e)/\beta_k$ is similar to the pointwise mutual information between feature k and entity e.[1] Thus the context score is the sum of the scores for all features present in the context.

[1] Pointwise mutual information is defined as $\text{pmi}(x;y) = \log p(y|x)/p(y)$.

Finally, it is important to note that our modeling goal here is not parameter sparsity, but rather capturing the sparsity of *useful* disambiguating features that occur in the context of each entity mention. In fact, the model parameters are not sparse in practice.

5 Learning and Inference

We parameterize the model as follows:

- vectors $\boldsymbol{\tau}_w$ parameterize the conditional probability of an entity given the phrase w, with $\tau_{w,e} = p(E = e|W = w)$
- vectors $\boldsymbol{\rho}_e$ parameterize the probability of relevant features for entity e, with $\rho_{e,k} = p(K = k|E = e)$
- scalars $\beta_j = p(B_j = 1|K \neq j)$ parameterize the background feature distribution.

We estimate the maximum likelihood parameters from both labeled and unlabeled data. The latent variables in our model are the relevant feature indices k_m for all mentions m, as well as the entities e_m for the unlabeled mentions. We approximate the posterior distribution over latent variables as a product of marginals, and use the following auxiliary distributions:

- $q_m(e)$ is the probability that mention m resolves to entity e, set to the ground truth for the labeled mentions.
- $s_m(k)$ is the auxiliary distribution over the relevant feature index for mention m.

We describe two approaches to estimating the parameters: (1) standard EM algorithm, where we infer all latent variables, and (2) a memory-efficient alternative. While both these approaches can be implemented using a distributed processing framework such as map-reduce (Dean and Ghemawat, 2008), the latter where we only infer the missing entity labels scales better than the standard EM approach. Simulations on synthetic data suggest that the two algorithms have similar performance (see Section 5.3) .

5.1 Standard EM algorithm

The EM algorithm for the model performs coordinate ascent in the following lower bound on the likelihood of observed data:

$$\mathcal{L} = \sum_m \sum_e q_m(e) \Big(\sum_w [w_m = w] \ln \tau_{e,w} + \sum_k s_m(k) \big(\ln(b_{m,k}\rho_{k,e}) - \ln \beta_k \big) + \sum_j b_{m,j} \ln \beta_j + (1 - b_{m,j}) \ln(1 - \beta_j) \Big) + \mathcal{H}(q) + \mathcal{H}(s) + const$$

where $[\cdot]$ is the Iverson bracket, and $\mathcal{H}(\cdot)$ is the entropy function. It can be shown that the iterative updates are given by:

E-step:

$$q_m(e) \propto \tau_{w,e} \exp\left(\sum_k s_m(k) \ln \rho_{e,k}/\beta_k \right)$$

$$s_m(k) = \frac{b_{m,k}}{\beta_k} \exp\left(\sum_e q_m(e) \ln \rho_{e,k} \right)$$

M-step:

$$\tau_{w,e} \propto \sum_m q_m(e)[w_m = w]$$

$$\rho_{e,k} \propto \sum_m q_m(e) s_m(k)$$

$$\beta_j = \frac{\sum_m (b_{m,j} - s_m(j))}{\sum_m (1 - s_m(j))}$$

5.2 Memory-efficient EM algorithm

One practical drawback to using the full EM algorithm is that maintaining the auxiliary distributions $q_m(e)$ and $s_m(k)$ requires a very large amount of memory, since they scale with the data. In this section we propose a simpler memory-efficient alternative, where we only update $q_m(e)$. We perform the E-step according to the entity posterior equation (2). In the M-step, rather than updating parameters $\{\beta_j\}$, we use empirical marginals. To update parameters $\{\boldsymbol{\rho}_e\}$, we approximate $s_m(k)$ by a fixed uniform distribution over features that fired, $s'_m(j) = b_{m,j} / \sum_k b_{m,k}$. The update for $\boldsymbol{\tau}$ remains the same as before. The memory-efficient EM algorithm is:

E-step:

$$q_m(e) \propto \tau_{w,e} \sum_k b_{m,k} \frac{\rho_{e,k}}{\beta_k} \quad (3)$$

M-step:

$$\tau_{w,e} \propto \sum_m q_m(e)[w_m = w] \quad (4)$$

$$\rho_{e,k} \propto \sum_m q_m(e) s'_m(k) \quad (5)$$

$$\beta_j = \frac{1}{M} \sum_m b_{m,j} \quad (6)$$

where M is the number of mentions. Note that these updates can be efficiently implemented in a map-reduce framework, where the map (E-step) computes the distribution $q_m(e)$, and the reduce (M-step) updates the parameters. This is the learning algorithm we use for all our experiments.

5.3 Simulation

We compared the performance of the two selective context EM algorithms (standard EM and its memory efficient variant) and the EM algorithm for naïve Bayes on synthetic data generated from our model. We left out the mention prior and only evaluated the context part. We assumed that there are $|\mathcal{E}| = 10$ equiprobable entities, $M = 2,000$ mentions, and $|\mathcal{F}| = 200$ possible features. Each entity was assigned a set of 5-10 randomly selected relevant features (these feature sets were allowed to overlap). For each mention, we drew one relevant feature according to $\boldsymbol{\rho}_e$, and a number of other features according to $\{\beta_j\}$. We sampled parameters $\{\boldsymbol{\rho}_e\}$ from a symmetric Dirichlet distribution, and parameters $\{\beta_j\}$ from a uniform distribution in $[0, \sigma]$, where σ roughly controlled the number of noisy features introduced. We generated synthetic datasets with $\sigma \in \{0.02, 0.03, 0.05, 0.07, 0.1, 0.15, 0.22, 0.33\}$. We then removed labels from half of the mentions and ran the three inference algorithms. To compare the results, we computed micro-averaged precision and recall over the unlabeled mentions, since these are the metrics we are ultimately interested in. The results are shown in Fig. 2, where each curve corresponds to a dataset. It is evident that the performance naïve Bayes is hindered by spurious features, even though each mention gets at least one discriminative feature. The two selective context EM algorithms are more robust to noise; this is as expected, since they are based on the true data-generating model. Note that the performance of the memory-efficient version is similar to that of the standard EM algorithm.

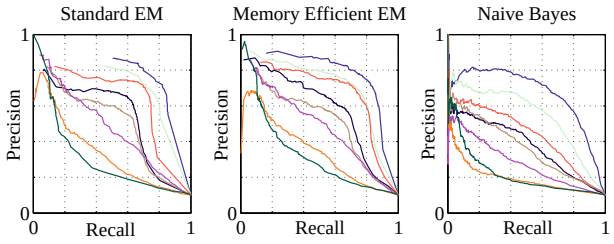

Figure 2: Micro-averaged precision-recall curves for the two EM algorithms for the selective context model (see Section 5), and the EM algorithm for the naïve Bayes model. Each color corresponds to a dataset with a different level of noise. It can be seen that in comparison to the naïve Bayes model, the selective context models are more resilient to spurious features. In addition the two learning approaches – standard EM and its memory efficient variant have similar performance.

5.4 Parallel Implementation

Semi-supervised learning has the important benefit of including many informative context features that are not present in the contexts of labeled training mentions. However, this comes at the cost of a very large model that in some cases may not fit in a single computer's memory. Fortunately, our inference algorithm is easily parallelizable. We partition the model across multiple servers in a distributed client-server architecture. This does not change any of the model parameters but rather partitions them so that they can be loaded into memory on multiple servers. Each entity is assigned to a server according to a heuristic algorithm described below, and model parameters for the entity are stored on the assigned server.

Clients process documents to find mentions (if they have not been provided), their contexts, and the context feature vectors for each mention. This process does not require access to the model. Each client stores a lookup table that maps mention phrases to the servers containing entities for which the mention phrase probability is nonzero. To re-

solve a mention, the lookup table is consulted to identify the servers that could resolve the mention. All mentions in a document (or a suitable batch of documents) can be sent in parallel to the selected servers. Thus, the time to resolve all mentions in a document is proportional to the maximum response time of an entity server, rather than to the sum of per-mention response times. Further, queries to the same entity server are batched together.

Once an entity server receives a query for a mention m (consisting of the phrase w_m and context features $\mathbf{b}_m$), it looks up the candidate entities for w_m, retrieves model parameters, and returns the entities $e_1, \ldots, e_n$ with the n highest $p(e|\mathbf{b}_m, w_m)$ (Equation 2). These lists are then merged across all the servers responsive to w_m to yield the n top-scoring entities for m.

We assign entities to servers by creating a bipartite entity-name graph and applying a greedy graph clustering algorithm that penalizes large clusters as they have a negative impact on load-balancing, while at the same time ensuring that most names will only be in one or a few clusters. Each cluster is then assigned to a server. Phrases such as `Washington` that evoke many entities may be distributed across multiple servers. Plato can also be made more responsive and resilient to server failure by replicating entity models across multiple servers.

6 Related Work

Entity resolution is a key step in many language-processing tasks such as text classification (Gabrilovich and Markovitch, 2007), information extraction (Lin et al., 2012) and grounded semantic parsing (Kwiatkowski et al., 2011). It can also help upstream tasks such as part-of-speech tagging, parsing, and coreference resolution (Finin et al., 2009; Mayfield et al., 2009) as it provides a link to world knowledge such as entity types, aliases, and gender.

Early work (Bunescu and Pasca, 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007; Milne and Witten, 2008; Nguyen and Cao, 2008) on the entity resolution problem focused on linking named entities to the relevant Wikipedia pages (also known as *Wikification*). Most of these systems resolved mentions by defining a similarity score between mention contexts and Wikipedia page contents. Mihalcea and Csomai (2007) and Han and Sun (2011) used naïve Bayes models similar to our baseline. Dredze et al. (2010) and Ratinov et al. (2011) used a ranking support vector machine (SVM), trained to put the correct entity above the remaining ones. More recently He et al. (2013) used stacked autoencoders to learn a context score.

While Bunescu and Pasca (2006) and Mihalcea and Csomai (2007) used local models only, others (Cucerzan, 2007; Milne and Witten, 2008; Kulkarni et al., 2009; Ferragina and Scaiella, 2010; Han and Zhao, 2009; Ratinov et al., 2011; Han et al., 2011; Hoffart et al., 2011; He et al., 2013; Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015) used a coherency model in conjunction with a local model. One popular approach to coherency has been to use variants of the PageRank algorithm (Page et al., 1999) to re-score candidates (He et al., 2013; Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015). Pershina et al. (2015) achieve the highest accuracy to date on the CoNLL 2003 dataset using a version of Personalized PageRank.

Chisholm and Hachey (2015) demonstrate the benefits of supplementing curated Wikipedia data with a large noisy Web corpus. Their model, relying on simple distance scores and an SVM ranker, achieves highly competitive results when trained on both Wikipedia and the Wikilinks corpus (Orr et al., 2013). While all of the systems described thus far rely on supervised learning, Plato is semi-supervised, and our experimental results (Section 9) confirm that incorporating unlabeled Web data can lead to significant performance gains.

Several recent entity resolution systems (Kataria et al., 2011; Han and Sun, 2012; Houlsby and Ciaramita, 2014) have been based on topic modeling. Han and Sun (2012) associate each document with a single topic; the topic generates entities, and the entities generate context words (non-referrent phrases). Houlsby and Ciaramita (2014) consider each entity to be a topic, and the words generated by the entity include both non-referent and referent phrases in the document, similarly to our context features. While topic modeling approaches are capable of exploiting both labeled and unlabeled data, inference, typically based on sampling, can be extremely slow when the number of entities is very

large. In contrast, our inference algorithm is simple to implement, parallelizable, scalable, and easily extended to the semi-supervised setting.

Jin et al. (2014) argue that many textual features within Web documents are irrelevant to the entity resolution task. Their method iteratively selects the most discriminative features for each document by trying to minimize the distance to some entity in the KB, and it outperforms a naïve Bayes model that includes all features. In contrast, we incorporate assumptions about feature relevance into our probabilistic model, and do not require access to the KB during learning or inference.

7 Data

Knowledge Base We use Freebase[2] (Bollacker et al., 2008) as our KB. Freebase data is harvested from many sources, including Wikipedia, AMG, and IMDB. As of this writing, it contains more than 21 million topics and 70 million properties. For a large majority of topics that appear both in Freebase and Wikipedia (Freebase covers more than 95% of Wikipedia), Freebase maintains a link to the Wikipedia page of that topic. While it is feasible to train our models using all of Freebase, for the experiments in this paper we restrict ourselves to the set of entities that appear in both Freebase and Wikipedia, as this is the standard setup used for our evaluation corpora.

Labeled Data We use all pages in Wikipedia that contain a corresponding topic in Freebase as labeled data. For a given Wikipedia page, we treat the target Wikipedia page of the link as the entity, and the anchor text as a mention of that entity (Milne and Witten, 2008; Han and Sun, 2011). We ignore disambiguation pages.

Unlabeled Data We collected a Web corpus of 50 million pages from a source similar to the Clueweb09 corpus (Hallan and Hoy, 2009) for use as unlabeled data.

Evaluation Data and Setup We used three evaluation corpora: (a) CoNLL 2003[3] (Hoffart et al., 2011), (b) TAC 2011 KBP (Ji et al., 2011), and (c) TAC 2012 KBP (Mayfield et al., 2012).

[2] www.freebase.com.

[3] www.mpi-inf.mpg.de/yago-naga/aida.

The CoNLL dataset contains 1,393 articles with about 34K mentions (Hachey et al., 2013). For the purpose of comparison with previous work, we evaluate Plato on the 231 `test-b` documents with 4,483 linkable gold mentions. Performance on those mentions is measured by micro-averaged precision@1, that is, accuracy averaged across mentions. We did *not* use CoNLL `train` or `test-a` documents for training or parameter tuning.

The TAC KBP competitions use a subset of a 2008 Wikipedia dump as the reference knowledge base. TAC 2012 evaluation data contains 2,226 gold mentions, of which 1,177 are linkable to the reference KB, and TAC 2011 data contains 2,250 mentions of which 1,123 are linkable to the same KB. Thus the TAC KBP competition evaluation includes NIL entities; participants are required to cluster NIL mentions across documents so that all mentions of each unknown entity are assigned a unique identifier. In addition to the official evaluation metric $B^{3+}F_1$, we also report Plato's in-KB accuracy as well as overall accuracy, where all NIL mentions are considered to be in one cluster. Once again as in the case of CoNLL, we did *not* train or tune Plato on any TAC KBP training data.

For each of the test corpora, we evaluate on the exact same set of mentions as previous work that we compare against in Tables 2 and 3. However, our setup differs from existing systems in two important ways. First, we train on Wikipedia and unlabeled Web data only, and do *not* use TAC or CoNLL training or development datasets. Second, candidate generation for each mention is based on our estimated mention prior (see Section 8 for details) and thus may differ from previous approaches. Plato's candidate recall is shown on Table 1; this is an upper bound on the accuracy of our approach.

8 Experimental Setup

Mention Prior We initialized the mention phrase parameters $\{\boldsymbol{\tau}_e\}$ from links in Wikipedia by counting how many times a given phrase w is used to refer to entity e, and normalizing appropriately (Han and Sun, 2012). We used the following sources to obtain (w, e) pairs for the above estimates: (a) w is the title of e's page after removing parenthesized disambiguation terms; (b) w is the title of a Wikipedia

Dataset	Candidate recall
CoNLL 2003	91.7
TAC 2011	84.8
TAC 2012	83.2

Table 1: Candidate generation recall on the three evaluation datasets: the percentage of linkable gold mentions for which the correct entity was in the set of candidates generated by Plato. This is an upper bound on our in-KB accuracy.

redirect page linking to e's page; (c) w is a Freebase alias (property `/common/topic/alias`) for Freebase topic e; (d) w is the title of a disambiguation page that links to e as a possible disambiguation. For all the aliases obtained from the above sources, we used the Wikilinks corpus (Orr et al., 2013) as an additional source of counts. In addition to the above sources, we also used anchors of Wikipedia pages as aliases if they occurred more than 500 times. Unlabeled data was used to reestimate the parameters $\boldsymbol{\tau}_e$ using Equation 4; however, we did not introduce any new aliases.

Context Features To extract context features, all documents were processed as follows. The free text in each document was POS-tagged and dependency-parsed using a parser that is a reimplementation of the MaltParser (Nivre et al., 2007) with a linear kernel SVM. When trained on Sections 02-21 of the Wall Street Journal (WSJ) portion of the Penn Treebank (Marcus et al., 1993), our parser achieves an unlabeled attachment score (UAS) of 88.24 and a labeled attachment score (LAS) of 84.69 on WSJ Section 22. Named mentions (such as `Barack Obama`) and common noun phrases (such as `the president`) were identified using a simple rule-based tagger with rules over POS tag sequences and dependency parses. We then used a within-document coreference resolver comparable to that of Haghighi and Klein (2009) to cluster all identified mentions.

As context features in our model, we used the phrases of all mentions in each coreference cluster in the document. We did not differentiate between phrases corresponding to the same coreference cluster as the query string, and phrases in other clusters. Adding other types of local features, such as words and phrases near the mentions in a cluster and dependency paths involving mentions in a cluster, did not lead to significant performance improvements in either our proposed model or naïve Bayes, and so we did not include them. We initialized the context parameters $\{\boldsymbol{\rho}_k\}$ using only labeled data, and re-estimated them using unlabeled data, as detailed in Equation 5.

Inference To determine the set of candidate entities for each coreference cluster, we use the mention with the longest phrase in the cluster. This phrase is used to generate the candidates which are then scored using Equation 2. We copy the label of the highest-scoring entity to all mentions in the cluster. Note that clusters will often include proper names mentions, referential common noun phrases, and referential pronouns. Thus Plato detects and links both nominal and named mentions. However, following most existing work, we only evaluate the resolution of gold *named entity* mentions to either KB or NIL. While in the case of CoNLL all gold mentions can be resolved to the KB, in TAC NIL is a valid label.

NIL Detection and Clustering As noted earlier, not all mentions correspond to an entity in the KB, even if they share a name or alias with a known entity. The problem is further complicated by the fact that it is hard to estimate the total number of entities in the world with a particular name. Plato decides whether to resolve a mention m (i.e., the cluster that m is a member off) to an entity in KB or to NIL based on context alone. Let

$$a_m^* = \max_e \sum_k b_{m,k} \frac{\rho_{e,k}}{\beta_k}.$$

If $a_m^* < \alpha$, we resolve i to NIL. We found that setting $\alpha = 1e^{-5}$ works well in practice. We use the above rule both during EM-based learning and at inference time.

The TAC KBP evaluation requires participants to perform cross-document clustering of NIL mentions, such that each unknown entity has a distinct NIL id. We employ a very simple clustering strategy, similar to Cucerzan (2012):

- Since our KB is much bigger than 2008 Wikipedia, Plato sometimes resolves mentions

Model	CoNLL 2003 test-b micro-accuracy
Hoffart et al. (2013)	82.5
Sil and Yates (2013)	84.2
Nguyen et al. (2014)	84.8
Houlsby and Ciaramita (2014)	84.9
He et al. (2013) + Han et al. (2011)	85.6
Chisholm and Hachey (2015) no coherency	86.1
Chisholm and Hachey (2015) with coherency	**88.7**
Mention prior	74.1
Naïve Bayes	76.1
Supervised selective context	79.7
Plato (semi-supervised selective context)	86.4

Table 2: Mention-averaged accuracy on the CoNLL 2003 dataset in our experiments and previous best work. The results of the best system are shown in bold-face.

to entities that are in our KB but not in the TAC KB. We assign the same NIL label to all such mentions of the same entity.

- We assign a unique NIL identifier to each coreference cluster that is resolved to NIL by Plato.

Naïve Bayes Model We estimated the parameters of the naïve Bayes model from labeled data only, using a symmetric Dirichlet prior to prevent zero probabilities. While the naïve Bayes model can also be extended to include unlabeled data using the EM algorithm (Nigam et al., 2000), we did not pursue this beyond preliminary experiments, since the initial results were mediocre and re-estimation on unlabeled data is known to be very sensitive to initialization.

9 Results

Table 2 summarizes entity resolution results on the CoNLL 2003 corpus. This evaluation only considers linkable mentions, and we compare different systems in terms of mention-averaged accuracy. As external baselines for CoNLL test-b, we show the results of (Nguyen et al., 2014; Sil and Yates, 2013; Houlsby and Ciaramita, 2014; He et al., 2013; Chisholm and Hachey, 2015). To the best of our knowledge, these are the top reported results for this dataset. We also note that the systems of Alhelbawy and Gaizauskas (2014) and Pershina et al. (2015) are highly competitive on CoNLL; however we do not include their results in Table 2 due to differences to the standard evaluation settings (while we evaluate on test-b as is standard practice they evaluate on the entire dataset).

Table 3 shows the evaluation results on the two TAC KBP corpora; these evaluations also consider NIL entities. Our baseline for TAC 2012 is the system of Cucerzan (2012), which achieved the highest accuracy and $B^{3+}F_1$ score in the 2012 competition. Our baseline for TAC 2011 is Cucerzan (2011), which achieved the highest overall accuracy and second-highest $B^{3+}F_1$ score in the 2011 competition. Dalton and Dietz (2013) also report high accuracy on TAC KBP data; however, their results are computed on non-standard training/evaluation data splits, and hence not directly comparable.

In both tables, we include the results of all of our experiments: (a) mention prior baseline, (b) supervised naïve Bayes (see Section 3 for details); (c) supervised selective context model; and (d) Plato, the semi-supervised selective context model.

The mention prior alone does surprisingly well on this task, but well below the previous best results, as might be expected. Supervised naïve Bayes performs better, but does not offer much improvement over the mention prior. The supervised selective context model performs much better than naïve Bayes, even though it is trained on exactly the same data. These results support our hypothesis that the performance of naïve Bayes suffers in the presence of irrelevant features. Finally, Plato outperforms the

Data	Model	In-KB accuracy	Overall accuracy	$B^{3+}F_1$
TAC 2011	Cucerzan (2011)	-	**86.8**	**84.1**
	Mention prior	67.9	78.7	75.8
	Naïve Bayes	69.3	79.6	76.5
	Supervised selective context	74.5	84.1	81.1
	Plato (semi-supervised selective context)	**79.3**	**86.5**	**84.0**
TAC 2012	Cucerzan (2012) Run 1	72.0	76.2	72.1
	Cucerzan (2012) Run 3	71.2	**76.6**	**73.0**
	Mention prior	47.0	59.1	52.5
	Naïve Bayes	50.6	61.3	55.1
	Supervised selective context	68.7	74.3	69.7
	Plato (semi-supervised selective context)	**74.2**	**76.6**	71.2

Table 3: TAC KBP evaluation results for our model and previous highest-accuracy systems. The best results are shown in bold-face; this includes the highest-accuracy system and systems whose performance was not statistically significantly different, according to a two-tailed t-test with $p = 0.05$.

other models in accuracy by a substantial margin, suggesting that incorporating unlabeled data helps the model generalize through feature discovery.

In comparison with previous work, Plato is highly competitive with existing results on all three datasets. On TAC 2012 data, Plato achieves the highest in-KB accuracy, and the same overall accuracy as the previous best system (Cucerzan, 2012). On TAC 2011 data, Plato once again achieves the highest in-KB accuracy, and has similar overall accuracy as the best system of (Cucerzan, 2011) (86.5 compared to 86.8).

The highest reported accuracy on CoNLL (test-b) is obtained by Chisholm and Hachey (2015), who use both Wikipedia and Wiklinks to train their model. Their results support the case for incorporating large amounts of noisy training data into entity linking systems. Without coherency, their model performs slightly worse than Plato, suggesting that coherency is a good direction for improving Plato. Recently, Pershina et al. (2015) have reported accuracy 91.8 on the entire CoNLL dataset (train, test-a, test-b) using a variant of Personalized PageRank. This is higher than Plato candidate recall (upper bound on our accuracy); as their publication is very recent, we have not had the chance to evaluate Plato on their candidates. Current Plato accuracy on the whole CoNLL dataset is 86.5.

Since many of the systems described here were trained on different datasets and features, it is hard to provide a deeper comparison of their properties. However, we reiterate that Plato's performance is out-of-the-box: it was *not* trained on CoNLL or TAC data, and we used the *exact* same model for all evaluations in Tables 2 and 3

Finally, we illustrate the favorable feature discovery properties of the semi-supervised approach with the following example:

> *George Harrison said that partnering with Roush Fenway Racing further demonstrates how Wii is bringing gaming to the masses.*

Here the mention `George Harrison` refers to the former senior vice-president (SVP) of Nintendo, and not the Beatle George Harrison (Freebase id `/m/03bnv`). The mention prior has a strong preference for the Beatles: $p(\text{Beatles}|\texttt{George Harrison}) = 0.92$ while $p(\text{SVP}|\texttt{George Harrison}) = 0.02$. In addition, none of the sentence context features occur in Wikipedia. Since the supervised model is trained only on Wikipedia, the mention prior component dominates, and the system incorrectly infers that `George Harrison` refers to the Beatle. However, once we retrain with unlabeled data, the model learns several new relevant features for the correct entity, including `Rouse Fenway Racing` and

Wii, and gaming. As a result we now get

$$p(\text{SVP}|\texttt{George Harrison}, \mathbf{b}) = 0.74$$
$$p(\text{Beatles}|\texttt{George Harrison}, \mathbf{b}) = 0.25,$$

which leads us to the correct inference for the person mentioned in the passage.

10 Conclusions and Future work

We have presented Plato, a simple and scalable entity resolution system that leverages unlabeled data to produce state-of-the-art results.

The main gains in our approach come from combining a novel *selective* context model with a large corpus of unlabeled Web documents. We have demonstrated that a model in which most features are considered noisy is superior to a model in which *all* features depend on the entity. However, in some circumstances such a model may fail to exploit all useful features. An obvious direction for future work is extending the framework to cases where a small subset of features can be relevant, for example using binary per-feature indicator variables. A more subtle direction for context modeling could involve distinguishing between salient entities, for which most features (mentions in that cluster) in a document are likely to be informative, and non-salient entities with few informative features.

Plato does not include a cross-entity coherency model; while such models are intuitively appealing, they depend on cross-entity links that are often missing for rare entities, and may require computationally demanding joint inference in a probabilistic model. We capture discourse coherency only by adding referring phrases in the document to the context features of each mention cluster (as strings, not entities). Very recent results by Chisholm and Hachey (2015) and Pershina et al. (2015) suggest that simple coherence-based rescoring can significantly boost performance, and so this is another potential direction for improving Plato.

While semi-supervised training leads to major accuracy gains for our method, it also creates very large models. We are able to serve those models with a simple distributed architecture, but it would be worth exploring inference methods that could reduce model size while not compromising accuracy. One possibility involves improving inference to select a small set of relevant features for each mention, rather than averaging over all features.

References

Ayman Alhelbawy and Robert Gaizauskas. 2014. Graph ranking for collective named entity disambiguation. In *Proc. 52nd Annual Meeting of the Association for Computational Linguistics*, ACL 14, pages 75–80.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250. ACM.

Razvan C. Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proc. 11th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 06.

Andrew Chisholm and Ben Hachey. 2015. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156.

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

Silviu Cucerzan. 2011. TAC entity linking by performing full-document entity extraction and disambiguation. In *In Proc. of the Text Analysis Conference*, TAC 11.

Silviu Cucerzan. 2012. The MSR system for entity linking at TAC 2012. In *In Proc. of the Text Analysis Conference*, TAC 12.

Jeffrey Dalton and Laura Dietz. 2013. A neighborhood relevance model for entity linking. In *Proc. of the 10th Conference on Open Research Areas in Information Retrieval*.

Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, January.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proc. of the 23rd International Conference on Computational Linguistics*, COLING 10, pages 277–285.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In *Proc. of the 19th ACM International Conference on Information Knowledge and Management*, CIKM 10, pages 1625–1628. ACM.

Tim Finin, Zareen Syed, James Mayfield, Paul McNamee, and Christine Piatko. 2009. Using Wikitology for cross-document entity coreference resolution. In *Proc. of the AAAI Spring Symposium on Learning by Reading and Learning to Read*. AAAI Press.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Harnessing the expertise of 70,000 human editors: Knowledge-based feature generation for text categorization. *Journal of Machine Learning Research*, 8:2297–2345, December.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with Wikipedia. *Artificial Intelligence*, 194(0):130 – 150.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proc.of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP 09. ACL.

Jamie Hallan and Mark Hoy. 2009. Clueweb09 data set. http://lemurproject.org/clueweb09/.

Xianpei Han and Le Sun. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1 of *ACLHLT 11*. ACL.

Xianpei Han and Le Sun. 2012. An entity-topic model for entity linking. In *EMNLP-CoNLL*, pages 105–115.

Xianpei Han and Jun Zhao. 2009. Named entity disambiguation by leveraging Wikipedia semantic knowledge. In *Proc. of the 18th Conference on Information and Knowledge Management*, CIKM 09, pages 215–224. ACM.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in Web text: a graph-based method. In *In Proc. 34th ACM SIGIR conference on research and development in information retrieval*, pages 765–774. ACM.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL 13, pages 30–34.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP11. ACL.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2013. AIDA: accurate online disambiguation of named entities in text and tables. http://www.mpi-inf.mpg.de/yago-naga/aida/index.html.

Neil Houlsby and Massimiliano Ciaramita. 2014. A scalable Gibbs sampler for probabilistic entity linking. In *Advances in Information Retrieval*, pages 335–346. Springer.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the TAC 2011 knowledge base population track. In *Proc. of the 4th Text Analysis Conference*, TAC 11.

Yuzhe Jin, Emre Kiciman, Kuansan Wang, and Ricky Loynd. 2014. Entity linking at the tail: sparse signals, unknown entities, and phrase models. In *Proc. of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 453–462, New York, NY, USA. ACM.

Saurabh S. Kataria, Krishnan S. Kumar, Rajeev R. Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu. 2011. Entity disambiguation with hierarchical topic models. In *Proc. of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1037–1045. ACM.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of Wikipedia entities in web text. In *Proc. of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 457–466. ACM.

Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523.

Thomas Lin, Mausam, and Oren Etzioni. 2012. Entity linking at web scale. In *Proc. of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

James Mayfield, David Alexander, Bonnie Dorr, Jason Eisner, Tamer Elsayed, Tim Finin, Clay Fink, Marjorie Freedman, Nikesh Garera, Paul McNamee, Saif Mohammad, Douglas Oard, Christine Piatko, Asad Sayeed, Zareen Syed, Ralph Weischedel, Tan Xu, and David Yarowsky. 2009. Cross-document coreference resolution: A key technology for learning by reading. In *AAAI 2009 Spring Symposium on Learning by Reading and Learning to Read*.

James Mayfield, Javier Artiles, and Hoa Trang Dang. 2012. Overview of the TAC 2012 knowledge base population track. In *Proc. of the 5th Text Analysis Conference*, TAC 12.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proc. of the 16th ACM Conference on Information and Knowledge Management*, CIKM 07, pages 233–242.

David N. Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proc. of the 17th ACM Conference on Information and Knowledge Management*, CIKM 07, pages 509–518.

Hien T. Nguyen and Tru H. Cao. 2008. Named entity disambiguation on an ontology enriched by Wikipedia. In *Proc. 2008 IEEE International Conference on Research, Innovation, and Vision for the Future in Computing and Communication Technologies*, RIVF 08, pages 247–254. IEEE.

Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. 2014. AIDA-light: High-throughput named-entity disambiguation. In *Proc. of the Linked Data on the Web Workshop*.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, May.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

Dave Orr, Amar Subramanya, and Fernando Pereira. 2013. Learning from big data: 40 million entities in context. http://googleresearch.blogspot.com/2013/03/learning-from-big-data-40-million.html.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank citation ranking: bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford InfoLab, November.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized Page Rank for named entity disambiguation. In *Proc. 2015 Annual Conference of the North American Chapter of the ACL*, NAACL HLT 14, pages 238–243.

Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACLHLT 11, pages 1375–1384. ACL.

Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In *Proc. of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM 13, pages 2369–2374. ACM.