Learning to Collectively Link Entities

Ashish Kulkarni
IIT Bombay
kulashish@gmail.com

Kanika Agarwal
IIT Bombay
kanika1712@gmail.com

Pararth Shah
IIT Bombay
pararthshah717@gmail.com

Sunny Raj Rathod
IIT Bombay
sunnyrajrathod@gmail.com

Ganesh Ramakrishnan
IIT Bombay
ganesh@cse.iitb.ac.in

ABSTRACT

Recently Kulkarni *et al.* [20] proposed an approach for collective disambiguation of entity mentions occurring in natural language text. Their model achieves disambiguation by efficiently computing exact MAP inference in a binary labeled Markov Random Field. Here, we build on their disambiguation model and propose an approach to jointly learn the node and edge parameters of such a model. We use a max margin framework, which is efficiently implemented using projected subgradient, for collective learning. We leverage this in an online and interactive annotation system which incrementally trains the model as data gets curated progressively. We demonstrate the usefulness of our system by manually completing annotations for a subset of the Wikipedia collection. We have made this data publicly available. Evaluation shows that learning helps and our system performs better than several other systems including that of Kulkarni *et al.*

Categories and Subject Descriptors

G.3 [**PROBABILITY AND STATISTICS**]: Markov processes

Keywords

entity disambiguation, associative markov network, learning

1. INTRODUCTION

Entity linking (EL) is the task of identifying and linking mentions embedded in unstructured text to their referent entity in a catalog like Wikipedia. This has been shown to benefit several systems, including search [4, 18], text classification [5], and other tasks. An entity linking system [21] typically consists of a spotter that first identifies short token segments ("spots") as potential mentions of entities from its catalog. Many entities may qualify for a mention, *e.g.*, "python" has over 15 senses in Wikipedia including *Python (genus)* and *Python (programming language)*. In the second stage, a disambiguator assigns zero or more entities to selected mentions, based on mention-entity coherence, as well as entity-entity similarity. Collectively, these two stages comprise an *annotator*.

Some of the recent work [33, 24] shows that several mentions may have no associated sense in the catalog. This is referred to as the no-attachment (NA) problem (or NIL in the TAC-KBP challenge [25]). The other relatively lesser addressed challenge is that of multiple attachments [20], where, a mention might link to more than one entities from the catalog. This might often be a result of insufficient context and has been acknowledged by some of the recent entity disambiguation challenges[1].

As we describe in the next section, lot of earlier work on entity disambiguation focused on per-mention disambiguation. There has been several recent research on collective disambiguation of all mentions leveraging features derived from both mention-local context and global entity-entity relatedness. However, the complexity of these models often makes it computationally unfeasible to jointly learn their feature weights. We leverage the disambiguation model of Kulkarni *et al.* [20] and propose an efficient approach to jointly learn the local and global feature weights.

2. PRIOR WORK

Entity disambiguation: Earlier works [10, 3, 26] on entity annotation focused on per-mention disambiguation. This involves selecting the best entity to assign to a mention, independent of the assignments to other mentions in the document. Wikify! [26] for instance, uses context overlap for disambiguation and combines it with a classifier model that exploits local and topical features. Cucerzan [9] introduced the notion of agreement on categories of entities in addition to the local context overlap, in which the entity context comprised out-links from and in-links to their corresponding Wikipedia documents. Milne *et al.* [27] formulated a "relatedness" measure of similarity between two entities from Wikipedia, based on their in-link overlap. Relatedness, in conjunction with the prior probability of occurrence of an entity, was then used to train a classifier model. Han *et al.* [16] leveraged the semantic information in Wikipedia to build a large-scale semantic network and developed a similarity measure to be used for disambiguation. Kulkarni *et al.* [21] were the first to propose a general collective disambiguation approach, giving formulations for trade-off between mention-entity compatibility and coherence between entities. Several graph-based approaches [17, 11] followed

[1]http://web-ngram.research.microsoft.com/ERD2014/

CODS '16, March 13-16, 2016, Pune, India

DOI: http://dx.doi.org/10.1145/2888451.2888454

that cast the disambiguation problem as a problem of dense subgraph selection from a graph of mentions and candidate entities, making use of collective signals.

Most of these systems seem to prefer tagging conservatively. Some of them [9, 17] restrict their tagging to named entities, while others use a subset of entities from a background taxonomy such as TAP [10] or Wikipedia [27, 26]. Others [19, 1] have proposed LDA-based generative models but focus only on person names. Some of the more recent systems [21, 15] do perform aggressive spotting, aided by the anchor dictionary of Wikipedia entities and study the recall-precision tradeoff.

Kulkarni *et al.* [20] propose a joint disambiguation model based on a Markov network of entities as nodes and edges for their relatedness. Disambiguation is achieved by performing a MAP inference on this graph and it naturally handles the NA and multiple attachment cases. Unlike other approaches [14, 28], their graph models candidate entities with binary labels, instead of mentions with multiple labels. A suitable assumption on cliques and their potentials makes efficient computation of exact inference possible. However, it is not clear as to how the node and edge feature weights are set.

Joint learning: To the best of our knowledge, none of the graph-based models above have attempted to jointly learn the node and edge feature weights. While there is prior work [31, 32] that applied graphical models to the problem of information extraction and coreference resolution, exact inference and estimation is intractable in these models. Similar approaches have also been applied to the problem of entity disambiguation [21, 14, 28], but hardly anyone has attempted to jointly learn the feature weights. Taskar *et al.* [29] proposed an approach to learn associative Markov networks (AMNs). They provide an approximate quadratic program for the problem of learning a margin maximizing Markov network and show that it is guaranteed to return optimal parameters for AMNs with binary-valued variables. Our learning approach is based on this work but we are perhaps the first ones to apply it in the text annotation domain. Further, we also extend this learning approach and propose an interactive active learning framework.

2.1 Our Contributions

We leverage the disambiguation model of Kulkarni *et al.* [20] and propose an efficient approach to jointly learn the node and edge feature weights. We also develop an interactive active learning framework that progressively improves the model as more training data becomes available. We implemented our approach in an online annotation system and used it to semi-automatically curate labeled data[2]. Our trained model performs better than several other systems including that of Kulkarni *et al.*

3. PRELIMINARIES

We borrowed the features and the disambiguation model from the work described in Kulkarni *et al.* and present it in brief here. We first start with the problem definition.

3.1 Problem Definition

The primary goal of document annotation is to link entity mentions in an input document to entities in a catalog. Mentions (or "spots") are contiguous token sequences in a text, *e.g. Bush*, that can potentially link to an entity, *e.g. George Bush* in the catalog. Let $\mathcal{M}_d$ be the set of all mentions in a document d and $\mathscr{E}$ be the set of all entities in the catalog. Then, the entity linking problem is to find for each mention $m \in \mathcal{M}_d$, the set of entities $\hat{E}_m \subset \mathscr{E} \cup \{NA\}$ that it can link to.

As a first step, the input text d is processed by a "spotter" to identify the set $\mathcal{M}_d$ of mentions and the set of candidates $E_m \subset \mathscr{E}$, $\forall m \in \mathcal{M}_d$. $e_m \in E_m$ is called a candidate entity for spot m. The set $E_d = \cup_{m \in \mathcal{M}_d} E_m$ forms the candidate entities set for document d. This is then followed by a "disambiguation" phase that obtains from E_m, the set $\hat{E}_m$ of entities that the mention m can actually link to. When none of the entities in E_m are valid, then $\hat{E}_m = \{NA\}$. Alternatively, more than one entities from E_m might get promoted to $\hat{E}_m$. Assuming *one sense per discourse* [13], an entity in the candidate set of more than one mentions, links (or does not link) to all those mentions.

3.2 Entity Catalog

A catalog is a structured knowledge base comprising categories with entities under them, along with their attributes and relations. Wikipedia has seen an extensive organic growth and covers entities spanning a vast set of domains. We report experimental results using the Wikipedia dump from May 2011, with approximately 4.4 million entities. For evaluation on ERD, we used as catalog, the snapshot of Freebase as provided in the challenge.

3.3 Spotter

We processed the Wikipedia dump and indexed it in separate fields storing page title, synopsis, frequent words, outlinks, full text *etc.* Given a document d, we first use the Stanford POS tagger to obtain a set $\mathcal{T}_d$ of tokens (and their spans) consisting of nouns, adjectives and verbs occurring in the document text. Tokens appearing in close spans are consolidated into a phrase if the phrase is an anchor text for an entity in Wikipedia. The set of tokens and phrases obtained after consolidation, is the set $\mathcal{M}_d$ of all potential mentions in the document. For each mention $m \in \mathcal{M}_d$, we then fire a fielded query against the catalog store to obtain the set E_m of candidate entities for the mention. In our experiments we only retain the top k (empirically set to 8) entities from the result set. In addition, we also used the Wikipedia Miner toolkit[3] to retrieve candidate senses for these tokens and include them in the set E_m.

3.4 Features

We used three types of features - (1) Popularity-based features of an entity: Prior Sense Probability [26], In-Link Count, Out-Link Count; (2) Mention-Entity compatibility features [21]; (3) Entity-Entity relatedness features: Category-based Similarity [9], In-link based Similarity [27], Out-link based Similarity, Contextual Similarity.

3.5 The Disambiguation Model

Having identified the set of candidate entities for each mention, a disambiguator attempts to link each mention to zero or more entities. The label of a candidate is a collective result of the interplay of local mention-entity and global entity-entity relatedness signals.

[2]http://tinyurl.com/entitydisamb-data

[3]http://sourceforge.net/projects/wikipedia-miner/

A *Markov Random Field* (MRF) is an undirected graphical model that captures local correlations between random variables [29].

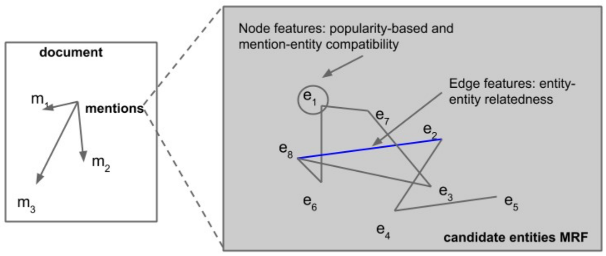

Figure 1: Candidate entities MRF model

A node is instantiated in the MRF graph for each possible entity mapping of each mention instance in a document. Edges capture entity-entity relatedness. Let $\mathbf{x}_i$ be the node feature vector of candidate i and $\mathbf{x}_{ij}$ be the edge feature vector of the edge joining candidates i and j. Each candidate corresponds to a random variable that takes a binary label, $y_i \in \{0,1\}$, based on whether or not it correctly disambiguates the underlying mention. Let C be the set of all cliques in the MRF and each clique $c \in C$ be associated with a clique potential $\phi_c(.)$. Cliques are restricted to nodes and edges and their potentials are parameterized by log-linear functions of feature vectors; *i.e.*, $log\ \phi_c(.) = \mathbf{w}_c \cdot \mathbf{x}_c$, where, $\mathbf{x}_c$ is the feature vector of a clique and $\mathbf{w}_c$, the corresponding weight vector. The potentials are assumed to be submodular [29], that is, they are associated with only those assignments, where, variables in a clique have the same label (associative). Moreover, these potentials are all greater than one and the rest are set to one. Thus, $log\ \phi_c(.) = 0$ for non-associative cliques and therefore we define node and edge weights only for associative cliques. Let $\mathbf{w}_0$ and $\mathbf{w}_1$ be the node feature weights influencing node labels 0 and 1 respectively and $\mathbf{w}_{00}$ and $\mathbf{w}_{11}$ be the associative edge weights influencing the connected nodes to take the same label. The probability of a complete graph labeling y is given by $P(y) = \frac{1}{Z}\prod_{c\in C}\phi_c(y_c)$, where Z is the partition function. Disambiguation is achieved by doing MAP inference on this graph.

$$\begin{aligned} L(y) &= \arg\max_{y\in\mathcal{Y}} \sum_{i\in\mathcal{N}} log\ \phi_i(y_i) \ + \sum_{ij\in\mathcal{E}} log\ \phi_{ij}(y_{ij}) \\ &= \arg\min_{y\in\mathcal{Y}} -(\sum_{i\in\mathcal{N}} \mathbf{w}_0 \cdot \mathbf{x}_i(1-y_i) + \mathbf{w}_1 \cdot \mathbf{x}_i y_i \\ &\quad + \sum_{ij\in\mathcal{E}} \mathbf{w}_{00} \cdot \mathbf{x}_{ij}(1-y_{ij}) + \mathbf{w}_{11} \cdot \mathbf{x}_{ij} y_{ij}) \end{aligned} \tag{1}$$

$\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges in the MRF, $y_i \in \{0,1\}$ and $y_{ij} = y_i y_j$. For an MRF with binary labeled nodes and associative edge potentials, MAP inference can be computed exactly in polynomial time, by finding the min cut of an augmented flow graph [2].

Construction of flow graph: The MRF is augmented by adding two special terminal nodes *source, s* and *sink, t* that correspond to the two labels 0/1. For each node i, we add terminal edges $s \rightarrow i$ with weight $\langle \mathbf{w}_0, \mathbf{x}_i \rangle$ and $i \rightarrow t$ with weight $\langle \mathbf{w}_1, \mathbf{x}_i \rangle$. For each neighborhood edge $i \rightarrow j$, we assign weight $\langle (\mathbf{w}_{00} + \mathbf{w}_{11}), \mathbf{x}_{ij} \rangle$. We also add weight $\langle \mathbf{w}_{00}, \mathbf{x}_{ij} \rangle$ to the edge $s \rightarrow i$ and $\langle \mathbf{w}_{11}, \mathbf{x}_{ij} \rangle$ to the edge $j \rightarrow t$. MAP inference in original MRF corresponds to the s/t min cut on this augmented graph, with nodes on the s side of the cut getting a label of 0, and the nodes on the t side being assigned a label of 1.

4. LEARNING FEATURE WEIGHTS

Algorithm 1: MRF Learning algorithm

Data: Training set $\{X, \hat{q}\}$, MRF graph g, Slack penalty C, Iterations T, Step size α_t
Result: Weight vector w
$w \leftarrow 0$
$t \leftarrow 1$
$N_n \leftarrow$ number of nodes in g
$f_{opt} \leftarrow \infty$
$w_{opt} \leftarrow 0$
while $t \leq T$ **do**
  $g \leftarrow$ construct flow network from g
  $\tilde{q} \leftarrow$ s/t mincut of g
  $\nabla_w \xi(w) \leftarrow 2w + C(\hat{q} - \tilde{q})^T\ X$
  $w \leftarrow w - \alpha_t\ \nabla_w \xi(w)$
  Project w onto the positive orthant
  Compute function value f
  if $f < f_{opt}$ **then**
    $f_{opt} \leftarrow f$
    $w_{opt} \leftarrow w$
  end
  $t \leftarrow t + 1$
end
return w_{opt}

The submodularity restriction and binary labels, make efficient implementation of learning possible. We jointly learn both node and edge feature weights following the general max-margin framework described in [29, 30]. Consider a graph with $\mathcal{N}$ nodes and $\mathcal{E}$ edges constructed as described above. Following Taskar *et al.*, the learning problem can be formulated in terms of the cut vector, such that, we minimize the norm of the weight vector subject to the constraint that the desired labeling scores better than an arbitrary labeling by an amount that scales with the Hamming distance between the desired and incorrect labelings.

$$\min_{\mathbf{w}\geq 0} \|\mathbf{w}\|^2 \tag{2}$$

subject to

$$\min_{\mathbf{q}\in\ Q} \sum_{i,j\in\mathcal{E}} \mathbf{w} \cdot \mathbf{x}_{ij}(q_{ij} - \hat{q}_{ij}) - (N_n - \hat{\mathbf{q}}_n^T \cdot \mathbf{q}_n) \geq 0$$

Here, Q is the set of all valid cuts and $q_{ij} \in \{0,1\}$ indicates if edge $i \rightarrow j$ is cut ($q_{ij} = 1$). $\mathbf{q}_n$ is the cut vector for terminal edges with components q_{si} and q_{it}, where, $q_{si} = 1$ implies that i is labeled 1. $\hat{\mathbf{q}}$ is the cut vector corresponding to the desired labeling. The first component of the constraint captures the difference in cost of the min cut induced by the weights $\mathbf{w}$ and that of the desired labeling. The other component corresponds to the number of labeling disagreements, N_n being the number of nodes in the graph (excluding s and t).

By rearranging terms, we obtain

$$\min_{\mathbf{w}\geq 0} \|\mathbf{w}\|^2 \tag{3}$$

$$\text{subject to } \min_{\mathbf{q} \in Q} \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij} + \hat{q}_{ij}(\delta_{is} + \delta_{jt}))q_{ij} \geq N_n + \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij})\hat{q}_{ij}$$

Here, δ_{ij} is the Kronecker delta. Now, consider the inequality constraint $\min_{\mathbf{q} \in Q} \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij} + \hat{q}_{ij}(\delta_{is} + \delta_{jt}))q_{ij} \geq N_n + \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij})\hat{q}_{ij}$. The left-hand-side of this inequality is equivalent to adding a capacity $\hat{q}_{ij}(\delta_{is} + \delta_{jt})$ to all cut terminal edges. Since each node participates in one terminal edge, this is equivalent to adding a total capacity of at-most N_n to the current flow graph. Therefore, $\min_{\mathbf{q} \in Q} \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij} + \hat{q}_{ij}(\delta_{is} + \delta_{jt}))q_{ij} \leq N_n + \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij})q_{ij} \leq N_n + \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij})\hat{q}_{ij}$. It follows that the left-hand-side and right-hand-side of the inequality in the constraint must be equal [30]. Moving the constraint to the objective, we get,

$$\min_{\mathbf{w} \geq 0} \|\mathbf{w}\|^2 + \mathcal{C}(N_n + \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij})\hat{q}_{ij} - \min_{\mathbf{q} \in Q} \sum_{i,j \in \mathcal{E}} (\mathbf{w}^T \cdot \mathbf{x}_{ij} + \hat{q}_{ij}(\delta_{is} + \delta_{jt}))q_{ij})$$

Summing over all the documents in the training set, we get the final objective,

$$\min_{\mathbf{w} \geq 0} \|\mathbf{w}\|^2 + \sum_{d \in D} (\mathcal{C} \ (\mathcal{N}_d + \sum_{i,j \in \mathcal{E}_d} (\mathbf{w}^T \cdot \mathbf{x}_{ij})\hat{q}_{ij} - \min_{\mathbf{q} \in Q} \sum_{i,j \in \mathcal{E}_d} (\mathbf{w}^T \cdot \mathbf{x}_{ij} + \hat{q}_{ij}(\delta_{is} + \delta_{jt}))q_{ij})) \tag{4}$$

Here, $\mathbf{w} = [\mathbf{w}_0^T \ \mathbf{w}_1^T \ \mathbf{w}_{00}^T \ \mathbf{w}_{11}^T]^T$, $\mathcal{N}_d$ is the number of nodes (excluding s and t) and $\mathcal{E}_d$ is the set of edges in the candidate entity MRF graph for a document $d \in D$, the set of all training documents, s and t are special source and sink nodes, respectively. The term $\mathcal{N}_d - \hat{q}_{ij}(\delta_{is} + \delta_{jt}))q_{ij}$ gives the number of misclassified nodes and $\sum_{i,j \in \mathcal{E}_d} \mathbf{w}^T \cdot \mathbf{x}_{ij}\hat{q}_{ij} - \mathbf{w}^T \cdot \mathbf{x}_{ij} q_{ij}$ is the total capacity of incorrectly cut edges in the flow graph. $\mathcal{C}$ is the penalty associated with the incorrect labeling. We solved the formulation (4) using the subgradient descent method as described in Algorithm 1.

4.1 Handling Unbalanced Training Data

The training data has many more entities labeled 0 as compared to those labeled 1. In our datasets, we observed a skew of about $3:1$. This results in a bias towards the over-represented class in the learning algorithm and the accuracy of the non-dominant class suffers. We addressed this problem by assigning separate misclassification penalties $\mathcal{C}_0$ and $\mathcal{C}_1$ for label 0 and 1 disagreements respectively in equation 4, where, disagreements are defined as below.

DEFINITION 1. *Let $l_i \in \{0,1\}$ and $\hat{l}_i \in \{0,1\}$ be the predicted and actual labels of node i. We say that a node i has label 0 disagreement if $l_i \neq \hat{l}_i = 0$. Similarly it has label 1 disagreement if $l_i \neq \hat{l}_i = 1$.*

PROPOSITION 1. *For an edge $i \rightarrow j$ with $q_{ij} \neq \hat{q}_{ij}$, exactly one of the nodes agrees on the label* i.e. $l_i = \hat{l}_i$ *(or $l_j = \hat{l}_j$) and the other node disagrees on the label* i.e. $l_j \neq \hat{l}_j$ *($l_i \neq \hat{l}_i$).*

PROOF. Case 1: Let $q_{ij} \neq \hat{q}_{ij} = 0$. This implies that the edge is not cut in the actual labeling and therefore $\hat{l}_i = \hat{l}_j$. However, $q_{ij} = 1$ implies that $l_i \neq l_j$. It follows that either $l_i = \hat{l}_i$ or $l_j = \hat{l}_j$.
Case 2: Let $q_{ij} \neq \hat{q}_{ij} = 1$. Following a similar argument as that for case 1 above, we have that $\hat{l}_i \neq \hat{l}_j$ and $l_i = l_j$. Again, it follows that either $l_i = \hat{l}_i$ or $l_j = \hat{l}_j$. □

DEFINITION 2. *An edge $i \rightarrow j$ with $q_{ij} \neq \hat{q}_{ij}$, is said to have a label 0 disagreement if either $l_i \neq \hat{l}_i = 0$ or $l_j \neq \hat{l}_j = 0$. It is said to have a label 1 disagreement if either $l_i \neq \hat{l}_i = 1$ or $l_j \neq \hat{l}_j = 1$.*

4.2 Active Learning

Our online annotation system presents an opportunity to continuously update the model as more labeled data becomes available. The commonly used *passive learning* approach involves manual annotation of randomly and independently sampled data. Due to the time and cost associated with this process, often there is not enough training data to meet certain level of performance. *Active learning* [22] aims to minimize the labeling effort, by requesting labels for the most informative samples, so as to achieve a desired level of accuracy. While there are several approaches to querying examples for labeling [23], we follow a pragmatic approach, that can be characterized as *least certain querying* method. The method samples examples with the smallest difference between two highest probability classes. Our binary labeled MRF model labels a node, based on the collective effect of the node potential and the edge potentials on the edges connecting the node to its neighbors. We define certainty $C(i)$ at a node n_i as

$$C(i) = \Big| \Big(w_0 \cdot x_i + \sum_{(ij) \in \mathcal{E}: j \in N(i)} w_{00} \cdot x_{ij} \Big) - \Big(w_1 \cdot x_i + \sum_{(ij) \in \mathcal{E}: j \in N(i)} w_{11} \cdot x_{ij} \Big) \Big| \tag{5}$$

where $N(i)$ is the set of all neighboring nodes of the node i. The certainty score $C(d)$ for a document d is then computed as the average certainty score across all nodes in that document.

$$C(d) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} C(i) \tag{6}$$

The active learning algorithm then queries for a document with the lowest $C(d)$ and presents the document for labeling. It might be possible to further reduce the labeling effort by requesting labels for only top k entities in the selected document, where the entities are ordered in increasing values of their $C(i)$.

5. EXPERIMENTS AND RESULTS

5.1 Data Sets

Table 1 shows details of the datasets used. We use $Wiki_{cur}$ (created by Kulkarni *et al.* [20]) for training our model and present cross-validation results. We also evaluate on several other datasets from the entity linking literature. Kulkarni *et al.* [21] had created a dataset ($IITB_{part}$) based on aggressive spotting but assuming single attachment. We, therefore, used our annotation system to manually complete annotations (to create $IITB_{cur}$) for the documents in this dataset. Three volunteers put in a combined effort of close to 300 hours to curate 57 documents in the $IITB_{cur}$ dataset. Each document was reviewed by at least one other volunteer. For a given document, our system displays the spots along with their candidate entity sets. Volunteers were instructed to:

1. link all correct entities (multiple attachments) to a mention;
2. add an entity manually if it is not already in the candidate set;
3. link disambiguation pages sparingly and only if they begin with a definition that is relevant in the context of the mention;
4. leave the mention untagged (NA) only if none of the above qualify and
5. manually add mentions (and their entities) whenever they were missed by our spotter.

The number of mentions is indicative of our aggressive spotting (Refer 3.3). It is encouraging to note that close to 15% of these mentions have multiple attachments[4] and around 30% have no attachment. In spite of our best efforts, data curation continues to be an extremely challenging task. We discuss some of these challenges in a later section (Refer to Section 6).

5.2 Evaluation Measures

We follow the *fuzzy evaluation measure* [7] that accounts for slight syntactic and semantic variations in the match of a predicted and true annotation, where an annotation a is defined as the mention-entity pair $\langle m, e \rangle$. Using their notion of *weak annotation match* $M_w(a_1, a_2)$[5], we use as performance metrics, *Recall*, *Precision* and *F1* micro-averaged over all documents in a dataset. After factoring out spotter errors, we also separately report the accuracy of our disambiguation model alone (Referred to as "disambiguator only"). This accuracy can be easily computed by comparing the predicted and true labels of candidate entities also present in the ground truth (*i.e.* $E_d \cap \hat{E}_d$ for a document d).

5.3 Experiments with only Node Features

5.3.1 Is there merit in data curation?

The data curation process presents an opportunity for continuous training where our inference model periodically evolves, as more and more data gets curated. Optionally, in the absence of any curated data to start with (at time $t = t_0$ when our model is yet untrained), one could use a Logistic Regression model, trained on a large uncurated dataset, to warm-start the data curation process. As data gets curated and our model is trained, we switch to our trained model at time $t = t_k$.

We trained binary label LR models using 10000 randomly sampled Wikipedia documents, replacing an original Wikipedia document with its curated version from $Wiki_{cur}$, one at a time. Figure 2 plots the training accuracies of these models for an increasing number of curated documents. The improvement in accuracy could be explained by the reduction in false negatives achieved by virtue of aggressive tagging and multiple attachments in the curated dataset. Based on this observation, we claim (and verify it in section 5.4.3) that our MRF model too would benefit from data curation. At the same time, the use of an LR model for warm-starting an online annotation system as ours is strongly recommended.

5.3.2 How does our model perform?

Thereafter, we trained our candidate entity MRF model on $Wiki_{cur}$ dataset using the node features alone. We report two-fold cross-validation results on $Wiki_{cur}$ and test results on $IITB_{part}$ (Refer to table 2). These serve as a baseline for our collective approach.

5.4 Collective Disambiguation

Next, we trained our model using node features and one or more edge features. Iterations T were fixed at 600, $\mathcal{C}$ was tuned as described below, and step size (at iteration t), $\alpha_t = K/\sqrt{t}$, where K was empirically set to 0.01.

5.4.1 Effect of $\mathcal{C}$ on accuracy

The $\mathcal{C}$ parameter in equation 4 acts as a regularizer and is indicative of the tolerance of disagreement between predicted and true labels. It was tuned on the training fold during two-fold cross-validation on $Wiki_{cur}$. Also, to account for the skew in label 0 and 1 instances in the dataset, we penalized label 0 and label 1 disagreements separately, using $\mathcal{C}_0$ and $\mathcal{C}_1$, respectively. A higher $\mathcal{C}_1$ for instance, improves the label 1 recall while adversely impacting the precision. It is this recall-precision tradeoff for varying values of $\mathcal{C}_0$, $\mathcal{C}_1$, that we capture in Figure 4. We chose the best $\mathcal{C}_0$ and $\mathcal{C}_1$ from these for all our experiments.

5.4.2 Effect of Edge Features

Table 3 shows the effect of different edge features in a collective setting. The model seems to benefit the most from the inlink and outlink relatedness features, while context overlap-based features seem to be noisy. This is understandable as context overlap-based signals are useful only for topically coherent entities, which might not hold true for an aggressively tagged corpus like ours [21].

5.4.3 Does training help?

We sampled 50 documents from the $Wiki_{cur}$ dataset, 5 at a time and used them for training, applying both passive (PL) and active learning (AL). The F_1 measure evaluated on an independent test set of 30 documents is shown in the plot (Refer to figure 3). The F_1 on training set seems to fluctuate, more so for *Train-AL*, as has been observed by others [6]. The performance of an active learner depends not only on training on instances that the model is least certain about, but, also on the informative features contained in them. The F_1 on test set does show a steady improvement and we expect this to improve further as more and more curated documents become available for training.

5.4.4 Comparison with collective approaches

We compared our system against several other collective annotation approaches: AIDA [17], Wikify! [26], TagMe [12], Wikipedia Miner [27] and Illionis Wikifier [28] on three datasets *viz.* $IITB_{part}$, AQUAINT (Wikipedia Miner) and MSNBC [9]. Our system consistently beats all these systems on all the three datasets (Table 4). Some of the other collective annotation systems like Cucerzan (F_1 : .45), CSAW [21] (F_1 :

[4]includes synonyms

[5]which is true iff mentions m_1 and m_2 overlap in the input text and entities e_1 and e_2 are synonyms

Table 1: Dataset statistics (Mean and Standard Deviation)

Dataset	Size	#Mentions		#Mentions as % of #words in a document		#NA		#Multiple attachments		#Overlapping mentions	
		M	SD	M	SD	M	SD	M	SD	M	SD
$Wiki_{cur}$	106	61.16	20.22	22.62	3.9	12.9	7.32	10.82	8.54	3.32	3.18
$IITB_{part}$	103	191.58	100.26	31.12	12.15	74.28	41.41	0	0	0	0
$IITB_{cur}$	57	196.02	82.68	31.5	5.9	63.89	37.54	25.5	18.43	3.07	2.85

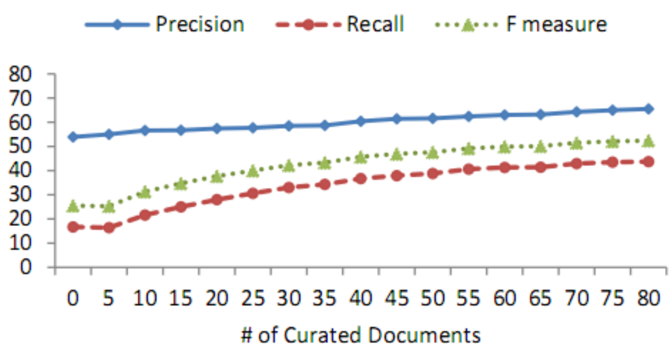

Figure 2: Effect of data curation

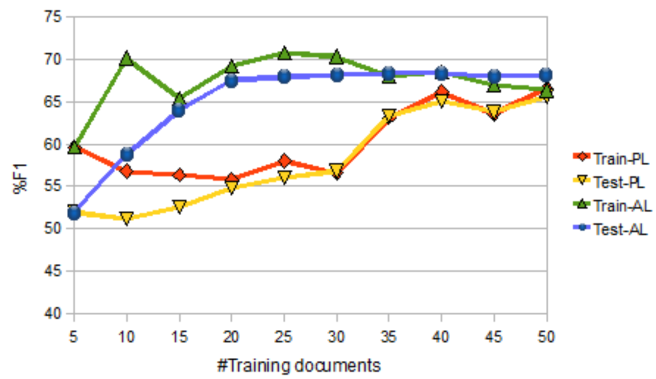

Figure 3: Effect of training

Table 2: Non-collective results (only node features) on $Wiki_{cur}$ set and $IITB_{part}$ datasets

Dataset	Disambiguator only			Weak annotation match		
	P	R	F	P	R	F
$Wiki_{cur}$	.82	.67	.74	.82	.56	.67
$IITB_{part}$	.82	.66	.73	.82	.50	.62

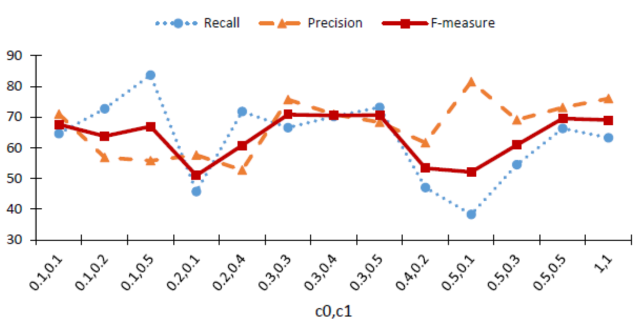

Figure 4: Effect of varying $\mathcal{C}$: $\mathcal{C}_0$ and $\mathcal{C}_1$

.69), [15] (F_1 : .73), and [14] (F_1 : .8) have used CSAW's evaluation measure to evaluate on $IITB_{part}$. We achieved an F_1 of 0.6 using the same measure. The relatively lower F_1 on this dataset could be attributed to inconsistencies between the ground truth and our knowledge base. During our manual annotation of $IITB_{part}$, we came across over 8000 annotations that were either added or removed[6] to create the $IITB_{cur}$ dataset.

[6]due to erroneous annotations or newer Wikipedia dump

Table 3: Effect of edge features: two-fold cross validation on $Wiki_{cur}$. Edge features that showed improvement over node features are shown in bold.

Edge feature	Disambiguator only			Weak annotation match		
	P	R	F	P	R	F
Category	.72	.74	.73	.72	.63	**.67**
Outlink (O)	.84	.67	**.74**	.84	.57	**.68**
Inlink (I)	.80	.73	**.76**	.80	.62	**.70**
Frequent (F)	.84	.64	.73	.84	.54	.66
Synopsis	.69	.61	.65	.69	.52	.59
Syn. V/Adj.	.69	.67	.68	.69	.57	.62
Full text	.85	.63	.73	.85	.54	.66
All features	.44	.50	.47	.44	.42	.43
I+O	.85	.67	**.74**	.85	.56	**.68**
I+O+F	.79	.74	**.76**	.79	.63	**.70**

We evaluated our system on the ERD dataset and achieved R:.62, P:.66, F_1 : .64. We believe that our system benefits from model training, thereby performing better than that of [20] (F_1 : .61). While some of the other systems [8] at ERD performed better, this could be attributed to their choice of features. Our system offers an end-end annotation framework that is interactive and jointly trains feature weights.

5.5 Results on $IITB_{cur}$

Section 6 shows some examples of incomplete annotations in the $IITB_{part}$ dataset. It is precisely such cases that we tried to correct during data preparation. Finally, we report the accuracy of our model on the $IITB_{cur}$ dataset - P : 77.4%, R : 54.3%, F_1 : 63.8%.

Table 4: Comparison with publicly available systems (as reported by Cornolti *et al.* [7]) on three datasets

	$IITB_{part}$			AQUAINT			MSNBC		
Annotator	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**
AIDA	.07	.66	.04	.21	.35	.15	.47	.75	.35
Wikify!	.37	.55	.28	.34	.29	.42	.41	.34	.51
TagMe	.44	.45	.42	.51	.46	.57	.52	.48	.55
Wikipedia Miner	.52	.57	.48	.47	.38	.63	.46	.55	.36
Illionis Wikifier	.44	.58	.36	.34	.29	.42	.41	.34	.51
Our Model (Node+I)	**.67**	.76	.60	.78	.81	.74	**.67**	.68	.66
Our Model (Node+I+O+F)	.65	.69	**.61**	**.79**	.82	**.75**	.66	.63	**.69**

5.6 Performance Evaluation

While our model allows for efficient inference and learning, graph construction itself is an expensive operation. For a document with $|E_d|$ candidate entities, the graph construction complexity is $O\left(|E_d|^2\right)$. For documents in the $Wiki_{cur}$ set with 190 candidate entities on an average, the average graph construction time was about 57 seconds. For the relatively larger documents in the $IITB_{cur}$ dataset, the average graph construction time was around 1.5 minutes. The performance could be improved by (a) pre-computing the entity-entity features for all entities in the knowledge base (b) dividing input document into chunks and performing graph construction and inference in parallel.

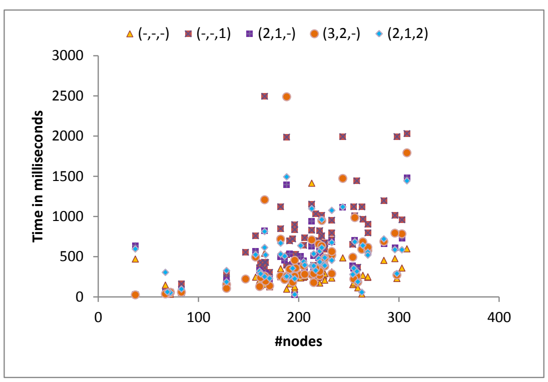

Figure 5: Running time for inference on $Wiki_{cur}$

The running time for inference (Figure 5) shows a slightly quadratic behavior in the number of candidate entities $|E_d|$ of a document. Inference on most documents runs in under 0.5 seconds. On the relatively sparser *Inlink+Outlink* graphs (Refer to Figure 6), training is much faster than the more dense *Category* graphs. The faster training happens without trading off much on accuracy as can be seen in Table 3. For our experiments, the model was retrained at time t using all the available training data. While this might be acceptable for offline training, online systems might benefit from faster incremental training approaches.

6. CHALLENGES WITH DATA CURATION

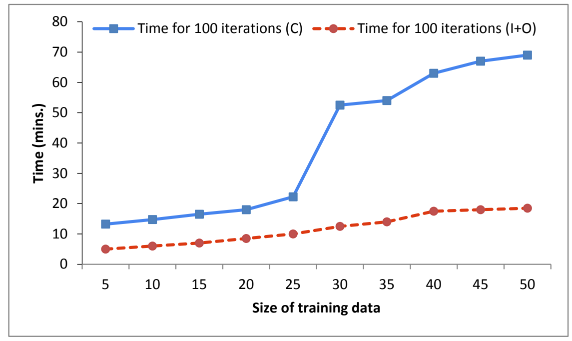

Figure 6: Scalability of training

Data curation is a tedious and challenging task. Its inherent ambiguity often introduces annotator bias leading to either incomplete or ambiguous annotations in the curated data.

1. There might be cases when two or more entities are correct as attachments for a mention. *E.g.*, mention 'Barack Obama' can be tagged as *Barack Obama* or *President of United States*, and both might seem correct in the context that it appeared. A 'one entity per mention' assumption makes it impossible to honor such cases.

2. Human annotators often limit their attention to the candidate entities retrieved by the spotter and very rarely search the catalog for any missed candidates. This results in a lot of missed annotations and often many mentions getting no attachments (NA).

3. Annotators also seem biased towards entity names that match with the mention text. However, this is often not true. *E.g.* a mention of 'cone snail' disambiguates to *Conidae* and *Conus*.

4. Wikipedia contains many disambiguation pages that often show up in the candidate set for a mention. Tagging a mention with a disambiguation page seems to beat the very purpose of a disambiguation system. Ideally, the mention should be annotated with one of the entities on the disambiguation page or NA if none of them is semantically right.

Table 5 shows some of these cases from the $IITB_{part}$ dataset. It is cases like these that we attempted to correct in coming up with the curated $IITB_{cur}$ dataset.

7. CONCLUSION

We presented an approach to jointly train the node and edge features of a collective disambiguation model for the purpose of entity linking. Our system leverages active learning to bring down labeling effort. Experiments show that the model benefits from training and improves with the availability of more labeled data. We consistently performed

Table 5: Examples of predictions on the $IITB_{part}$ highlighting the challenges in data curation

Ground Mention	Ground Entity	Predicted Entity	Remarks
lifestyle	Lifestyle	NA	Disambiguation page attachment
harsh reality	NA	Reality	Incomplete data
effort	NA	Energy	Incomplete data
self discipline	Discipline	self → Self, discipline → Discipline	Overlapping mentions
god	God (male deity)	God	Multiple correct entities
intellect	Intelligence	Intellect	Multiple correct entities

better than many other systems on various datasets. It also scales reasonably well and with suggested tweaks can be used for large scale document annotation.

8. ACKNOWLEDGMENTS

This research was supported by the Intranet Search project from IRCC at IIT Bombay.

9. REFERENCES

[1] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SIAM International Conference on Data Mining*, 2006.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[3] R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16, 2006.

[4] S. Chakrabarti, K. Puniyani, and S. Das. Optimizing scoring functions and indexes for proximity search in type-annotated corpora. In *Proceedings of the 15th international conference on World Wide Web*, pages 717–726, New York, NY, USA, 2006.

[5] M.-W. Chang, L. Ratinov, D. Roth, and V. Srikumar. Importance of semantic representation: Dataless classification. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 830–835. AAAI Press, 2008.

[6] J. Chen, A. Schein, L. Ungar, and M. Palmer. An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 120–127, Stroudsburg, PA, USA, 2006.

[7] M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 249–260, Republic and Canton of Geneva, Switzerland, 2013.

[8] M. Cornolti, P. Ferragina, M. Ciaramita, H. Schütze, and S. Rüd. The smaph system for query entity recognition and disambiguation. In *Proceedings of the First International Workshop on Entity Recognition and Disambiguation*, ERD '14, pages 25–30, New York, NY, USA, 2014. ACM.

[9] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, volume 6, pages 708–716, 2007.

[10] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, pages 178–186, New York, NY, USA, 2003.

[11] A. Fahrni and M. Strube. Jointly disambiguating and clustering concepts and entities with markov logic. In *COLING*, pages 815–832, 2012.

[12] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628, New York, NY, USA, 2010.

[13] W. A. Gale, K. W. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237, Stroudsburg, PA, USA, 1992.

[14] X. Han and L. Sun. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 105–115, Stroudsburg, PA, USA, 2012.

[15] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM, 2011.

[16] X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 215–224, New York, NY, USA, 2009. ACM.

[17] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Stroudsburg, PA, USA, 2011.

[18] G. Kasneci, F. M. Suchanek, G. Ifrim, S. Elbassuoni, M. Ramanath, and G. Weikum. Naga: harvesting, searching and ranking knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1285–1288, New York, NY, USA, 2008.

[19] S. S. Kataria, K. S. Kumar, R. R. Rastogi, P. Sen, and S. H. Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge*

discovery and data mining, pages 1037–1045, New York, NY, USA, 2011.

[20] A. Kulkarni, K. Agarwal, P. Shah, S. R. Rathod, and G. Ramakrishnan. System for collective entity disambiguation. In *Proceedings of the First International Workshop on Entity Recognition and Disambiguation*, ERD '14, pages 111–118, New York, NY, USA, 2014. ACM.

[21] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.

[22] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *In Proceedings of the 11th International Conference on Machine Learning (ICML)*, pages 148–156, 1994.

[23] M. Li and I. K. Sethi. Confidence-based active learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(8):1251–1261, Aug. 2006.

[24] T. Lin, Mausam, and O. Etzioni. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 84–88, Stroudsburg, PA, USA, 2012.

[25] P. McNamee. Overview of the tac 2009 knowledge base population track. 2009.

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

[27] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[28] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 1375–1384, Stroudsburg, PA, USA, 2011.

[29] B. Taskar, V. Chatalbashev, and D. Koller. Learning associative markov networks. In *Proceedings of the twenty-first international conference on Machine learning*, page 102. ACM, 2004.

[30] P. Vernaza, B. Taskar, and D. Lee. Online, self-supervised terrain classification via discriminatively trained submodular markov random fields. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2750–2757. IEEE, 2008.

[31] B. Wellner, A. McCallum, F. Peng, and M. Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 593–601, Arlington, Virginia, United States, 2004.

[32] M. L. Wick, A. Culotta, K. Rohanimanesh, and A. McCallum. An entity based model for coreference resolution. In *SDM*, pages 365–376, 2009.

[33] Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and S. Gaffney. Resolving surface forms to wikipedia topics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1335–1343, Stroudsburg, PA, USA, 2010.