A Discriminative Latent Variable Model for Clustering of Streaming Data with Application to Coreference Resolution

Rajhans Samdani RSAMDAN2@ILLINOIS.EDU
Kai-Wei Chang KCHANG10@ILLINOIS.EDU
Dan Roth DANR@ILLINOIS.EDU

Abstract

We present a latent variable structured prediction model, called the Latent Left-linking Model (L3M), for discriminative supervised clustering of items that follow a streaming order. L^3M admits efficient inference and we present a learning framework for L^3M that smoothly interpolates between latent structural SVMs and hidden variable CRFs. We present a fast stochastic gradient-based learning technique for L^3M. We apply L^3M to coreference resolution, which is a well known clustering task in Natural Language Processing, and experimentally show that L^3M outperforms several existing structured prediction-based techniques for coreference as well as several state-of-the-art, albeit ad hoc, approaches.

1. Introduction

Many applications require clustering of items appearing as a data stream, e.g. weather monitoring, financial transactions, network intrusion detection (Guha et al., 2003), and email spam detection (Haider et al., 2007). In this paper, we focus on discriminative supervised learning for data stream clustering with features defined on pairs of items. This setting is different and more general than supervised metric learning techniques (Xing et al., 2002) and k-centers style approaches that have been widely studied in the data mining literature (e.g. Guha et al. (2003)).

We present a novel and principled discriminative model for clustering streaming items which we call a Latent Left-Linking Model (L^3M). L^3M is a feature-based probabilistic structured prediction model where each item can *link* to a previous item with a certain probability, creating a *left-link*. L^3M expresses the probability of an item connecting to a previously formed cluster as a sum of the probablities of multiple left-links connecting that item to the items inside that cluster. We use a temperature-like parameter in L^3M (Samdani et al., 2012a;b) which allows us to tune the entropy of the resulting probability distribution.

We show that L^3M admits efficient inference, which is quadratic in the number of items. For learning in L^3M, we present a latent variable based objective function that generalizes and interpolates between hidden variable conditional random fields (HCRF) (Quattoni et al., 2007) and latent structural support vector machines (LSSVM) (Yu & Joachims, 2009) using a temperature parameter (Schwing et al., 2012). We present a fast stochastic gradient technique for learning that can update the model within the marginal inference routine, without having to wait for inference to finish. Our stochastic gradient strategy, despite being hard to theoretically characterize, provides great empirical performance; we show that tuning the temperature parameter also leads to significant gains in performance.

In this paper, we focus on coreference resolution as an application for clustering of streaming data. Coreference resolution is a challenging task, requiring a human or a system to identify denotative noun phrases called *mentions* and cluster those mentions together that refer to the same underlying entity. In other words, coreference resolution is the task of identification and clustering of mentions where two mentions share the same cluster if and only if they refer to the same entity. For example, in the following sentence, mentions with same subscript numbers are *coreferent*:

[Former Governor of Arkansas]$_1$, [Bill Clinton]$_1$, who was recently elected as the [President of the U.S.A.]$_1$, has been invited by the [Russian President]$_2$, [Vladimir Putin]$_2$, to visit [Russia]$_3$. [President Clinton]$_1$ said that [he]$_1$ looks forward to strengthening the relations between [Washington]$_4$ and [Moscow]$_5$.

We argue that the right way to view coreference clustering is as a streaming data clustering problem, where the mentions can be thought of as streaming items. This is motivated by the linguistic intuition that humans are likely to resolve coreference for a given mention based on antecedent mentions which are on its left (in a left-to-right

Appearing in *Proceedings of the 30^{th} International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013.

writing manner.) While this insight in itself is not new and has been used before successfully (Soon et al., 2001; Ng & Cardie, 2002; Bengtson & Roth, 2008; Chang et al., 2012), L^{3}M is the first attempt at formalizing this approach to coreference as a probabilistic structured prediction problem. Furthermore, L^{3}M is a strict generalization of previous left-linking approaches to coreference, as they connect each mention to at most one antecedent mention. In our experiments, L^{3}M outperforms several competing algorithms on benchmark datasets.

2. Notation and Pairwise Classifier

Notation: For a given data stream d, let m_d be the total number of items[1] in d, e.g. in coreference, d could be a document and m_d could be the number of mentions in d. We refer to items using their indices which range from 1 to m_d. A clustering $\mathcal{C}$ for a data stream d is represented as a set of disjoint sets partitioning the set $\{1, \ldots, m_d\}$. Alternatively, we also represent $\mathcal{C}$ as a binary function with $\mathcal{C}(i,j) = 1$ if items i and j are co-clustered, otherwise $\mathcal{C}(i,j) = 0$. During training, we are given a collection of data streams D, where for each data stream $d \in D$, $\mathcal{C}_d$ refers to the annotated ground truth clustering.

Pairwise classifier: We use a pairwise scoring function that indicates the compatibility of a pair of items. These pairwise scores are used as basic building block for clustering, which we will pose as a structured prediction problem. In particular, for any two items i and j, we can produce a pairwise compatibility score w_{ij} using a scoring component that uses extracted features $\phi(i,j)$ as

$$w_{ij} = \mathbf{w} \cdot \phi(i,j) \ . \tag{1}$$

The extracted features contain different features indicative of the (in)compatilbility of items i and j. For instance, in coreference, these features could be lexical overlap, mutual distance, gender match, etc. This pairwise approach is by far the most common, straightforward, and successful approach to the modeling of coreference (Bengtson & Roth, 2008; Stoyanov et al., 2009; Ng, 2010; Fernandes et al., 2012) and was also shown to be useful for clustering of streaming spam emails by Haider et al. (2007).

3. Probabilistic Latent Left-linking Model

In this section, we describe our Latent Left-Linking Model (L^{3}M.) The idea of Latent Left-Linking Model is inspired by a popular inference approach to coreference clustering which we call the *Best-Left-Link* approach. In the Best-Left-Link strategy, each mention, i.e. item, i is connected to the best antecedent mention j with $j < i$ (i.e. a mention occurring to the left assuming a left-to-right reading order.) The "best" antecedent mention is the one with the highest pairwise score w_{ij}; furthermore, if w_{ij} is below some threshold, say 0, then i is not connected to any antecedent mention. The final clustering is a transitive closure of these "best" links. The intuition for this strategy is placed in how humans read and decipher coreference links. While this approach has been empirically successful for coreference clustering (Soon et al., 2001; Ng & Cardie, 2002; Bengtson & Roth, 2008), this paper, to the best of our knowledge, is the first attempt at formalizing this approach as a structured prediction problem generally applicable to data stream clustering, generalizing the inference problem, and presenting principled learning techniques.

3.1. Latent Left-Linking Model:

In the Best-Left-Link approach, each item connects to the "best" antecedent. However, a machine learning system based on a pairwise classifier may not be able to make the right decision by looking at just one best item. To see this, consider the following coreference clustering example from the introduction:

[Former Governor of Arkansas]$_1$, [Bill Clinton]$^i_1$, who was recently elected as the [President of the U.S.A.]$^j_1$, has been invited by the [Russian President]$^k_2$, [Vladimir Putin]$_2$, to visit [Russia]$_3$. [President Clinton]$^l_1$ said that [he]$_1$ looks forward to strengthening the relations between [Washington]$_4$ and [Moscow]$_5$.

Let us say that we are trying to resolve the membership of mention l ('President Clinton') and that all the previous mentions have been correctly clustered. It is possible that the Best-Left-Link strategy might prefer to link mention l to mention k (which is incorrect) over mentions i and j (which are correct) as all the mentions i, j, and k have similar lexical overlap with mention l, but k is closer[2]. However, by looking at both links, from l to j and from l to i (and combining the scores of these links), it is possible for a pairwise classifier-based system to rule out the link from l to k. We formalize and generalize this idea in L^{3}M.

In order to simplify the notation and the description, we create a dummy item with index 0, which is to the left of all the items and has $\phi(i,0) = \emptyset$ and $w_{i0} = 0$ for all items i. Furthermore, for a clustering $\mathcal{C}$, if an item i is not co-clustered with any previously occurring item, then we assume $\mathcal{C}(i,0) = 1$, so that $\sum_{0 \le j < i} \mathcal{C}(i,j) \ge 1$.

[1] We use m_d as the total number items only for notational convenience and because some applications like coreference have a finite number of items. As such m_d can be very large, possibly infinite, and our clustering and learning algorithm will work just as well.

[2] We observe in our experiments that the distance feature does indeed get a very high weight.

Probabilistic Left Link: In our L^3M approach, we assume that each item can connect to an antecedent item on its left (i.e. occurring before it) with a certain probability. However, this left-linkage remains latent as the final clustering, and not the left-links, is the output variable of interest and is observed during training. Furthermore, we assume that the event that an item i links to antecedent mention j is independent of the event that any item $i', i' \neq i$, links to some mention j'. In particular, for a data stream d, each item $i \geq 1$ connects to an item j, $0 \leq j < i$, with probability $P(i \rightarrow j; w)$ given by

$$Pr[i \rightarrow j; d, \mathbf{w}] = \frac{e^{\frac{1}{\gamma}(\mathbf{w}\cdot\phi(i,j))}}{Z_i(\mathbf{w}, \gamma)} \quad , \tag{2}$$

where $Z_i(\mathbf{w}, \gamma) = \sum_{0 \leq k < i} e^{\frac{1}{\gamma}(\mathbf{w}\cdot\phi(i,k))}$ is a normalizing constant and $\gamma \in (0, 1]$ is a constant temperature parameter (Samdani et al., 2012a;b).

Clustering Probability with Latent-Left Links Let us assume that we cluster items in a streaming order and when looking at item i, we have already created a certain set of clusters. We assume that the dummy item 0 is in its own cluster which it does not share with any other item. Now, if $Pr[i, c; d, \mathbf{w}]$ is the probability that item i is assimilated in clustering c then $Pr[i, c; d, \mathbf{w}]$ is given by:

$$\begin{aligned} Pr[i, c; d, \mathbf{w}] &= \sum_{j \in c, 0 \leq j < i} Pr[i \rightarrow j; d, \mathbf{w}] \\ &= \sum_{j \in c, 0 \leq j < i} \frac{e^{\frac{1}{\gamma}(\mathbf{w}\cdot\phi(i,j))}}{Z_i(\mathbf{w}, \gamma)} \quad . \end{aligned} \tag{3}$$

Note that the probability of linking an item to a cluster takes into account all the items inside that cluster, mimicking the notion of an item-to-cluster link.

The case of $\gamma = 0$: As γ approaches zero, the probability $P[i \rightarrow j; d, w]$ approaches a Kronecker delta function that puts probability 1 on item $j = \arg\max_{0 \leq k < i} \mathbf{w} \cdot \phi(i, j)$, and 0 everywhere else. This the same as the Best-Left-Link approach which considers only the highest scoring antecedent item. Similarly, $Pr[i, c; d, \mathbf{w}]$ in Eq. 3 approaches a Kronecker delta function centered on the cluster containing the best antecedent. Thus, for the rest of this section, we abuse the notation and use the expressions in Eq. (2) and (3) for all $\gamma \in [0, 1]$, where for $\gamma = 0$, the probability distributions are assumed to be replaced by the appropriate Kronecker delta distribution. We will show how tuning the value of $\gamma \in [0, 1]$ can yield interesting learning and inference algorithms, and improve the prediction accuracy.

3.2. Inference

Inference or decoding is the task of creating a final clustering for a given data stream. Due the assumption that all items create left-links independently, once an item i is clustered, the clustering decision is not reconsidered later on. Combining this insight with Eq. (3) implies that we can do inference in L^3M in a greedy left-to-right fashion.

Algorithm 1 Inference algorithm for L^3M.

1: **Given:** Data stream d and weights $\mathbf{w}$
2: **Initialize:** Clustering $\mathcal{C} = \emptyset$
3: **for** $i = 1, \ldots, m_d$ **do**
4: $\quad bestscore \leftarrow 0, bestcluster \leftarrow \emptyset$
5: $\quad$ **for** $c \in \mathcal{C}$ **do**
6: $\quad\quad score \leftarrow \begin{cases} \sum_{j \in c} e^{\frac{1}{\gamma}(\mathbf{w}\cdot\phi(i,j))} & \gamma > 0 \\ \max_{j \in c} e^{(\mathbf{w}\cdot\phi(i,j))} & \gamma = 0 \end{cases}$,
7: $\quad\quad$ **if** $score > bestscore$ **then**
8: $\quad\quad\quad bestscore \leftarrow score, bestcluster \leftarrow b$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad$ **if** $bestscore > 1$ **then**
12: $\quad\quad \mathcal{C} \leftarrow \mathcal{C} \setminus \{bestcluster\} \cup \{bestcluster \cup \{i\}\}$
13: $\quad$ **else**
14: $\quad\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$
15: $\quad$ **end if**
16: **end for**
17: return $\mathcal{C}$

Alg. 1 presents the inference routine that returns the clustering in the form of a set of sets of co-clustered items. Note that this algorithm does not make use of the dummy item (item 0). For each item i, lines 5-10 detect the best existing cluster ($bestcluster$) to connect item i to and compute the score of connecting i to this cluster. Line 11 checks if this score is greater than a threshold of 1, which is the unnormalized score of letting i remain unconnected (or the implicit score of connecting i to item 0.) If the $bestscore$ is greater than 1, then i is connected to $bestcluster$, otherwise it starts its own cluster.

Note that for $\gamma = 0$, this inference is the same as the Best-Left-Link inference (Ng & Cardie, 2002; Bengtson & Roth, 2008; Chang et al., 2011), where each item is linked to a cluster solely based on a single pairwise link to the best-link item. Thus by tuning γ value we generalize the Best-Left-Link inference and allow other items to play a role in clustering. Also, note that the time complexity of L^3M inference, despite entertaining many left-links at the same time, is the same as that of Best-Left-Link inference i.e. $O(m_d^2)$.

3.3. Latent Variable Learning

Given a set of annotated training data streams D and annotated clustering $\mathcal{C}_d$ for each data stream $d \in D$, the task of learning is to estimate $\mathbf{w}$. We will use a likelihood-based approach to learning, and compute the probability

$Pr[\mathcal{C}_d; d, \mathbf{w}]$ of generating a clustering C_d, given $\mathbf{w}$.

Likelihood Computation: Due to our assumption that all the items link to the left independent of other items, we can write down $Pr[\mathcal{C}_d; d, \mathbf{w}]$ as the product of the probabilities of each item i connecting in a manner consistent with $\mathcal{C}_d$:

$$Pr[\mathcal{C}_d; d, \mathbf{w}] = \prod_{i=1}^{m_d} Pr[i, \mathcal{C}_d; d, \mathbf{w}] \ , \tag{4}$$

where $Pr[i, \mathcal{C}_d; d, \mathbf{w}]$ is the probability that item i, $i \geq 1$, connects to its left in a manner consistent with $\mathcal{C}_d$ i.e. this is the probability that i links to an antecedent item which is actually co-clustered with i in the clustering $\mathcal{C}_d$. Using Eq. (3), $Pr[i, \mathcal{C}_d; d, \mathbf{w}]$ is simply given by:

$$\begin{aligned} Pr[i, \mathcal{C}_d; d, \mathbf{w}] &= \sum_{0 \leq j < i} \frac{e^{\frac{1}{\gamma}(\mathbf{w} \cdot \phi(i,j))} \mathcal{C}_d(i,j)}{Z_i(\mathbf{w}, \gamma)} \\ &= \frac{Z_i(\mathcal{C}_d, \mathbf{w}, \gamma)}{Z_i(\mathbf{w}, \gamma)}, \end{aligned} \tag{5}$$

where $Z_i(\mathcal{C}_d, \mathbf{w}, \gamma) = \sum_{0 \leq j < i} e^{\frac{1}{\gamma}(\mathbf{w} \cdot \phi(i,j))} \mathcal{C}_d(i,j)$. Essentially $Z_i(\mathcal{C}_d, \cdot, \cdot)$ can be thought of as the unnormalized probability mass out of $Z_i(\cdot, \cdot)$ allocated to connecting as per clustering $\mathcal{C}_d$. Finally substituting (5) in (4), we get

$$Pr[\mathcal{C}_d; d, \mathbf{w}] = \prod_{i=1}^{m_d} \frac{Z_i(\mathcal{C}_d, \mathbf{w}, \gamma)}{Z_i(\mathbf{w}, \gamma)} \ . \tag{6}$$

Thus the log-likelihood of data D is given by

$$\begin{aligned} & \sum_{d \in D} \log Pr[\mathcal{C}_d; d, \mathbf{w}] \\ = & \sum_{d \in D} \sum_{i=1}^{m_d} \left(\log Z_i(\mathcal{C}_d, \mathbf{w}, \gamma) - \log Z_i(\mathbf{w}, \gamma) \right). \end{aligned} \tag{7}$$

Objective Function for Learning: We learn $\mathbf{w}$ by minimizing the regularized negative log-likelihood of the data, $LL(\mathbf{w})$, augmented with a softmax loss-based margin similar to Gimpel & Smith (2010):

$$\begin{aligned} LL(\mathbf{w}) &= \tfrac{\lambda}{2} \|\mathbf{w}\|^2 + \tfrac{1}{|D|} \sum_{d \in D} \sum_{i=1}^{m_d} \left(\log Z'_i(\mathbf{w}, \gamma) \right. \\ &\quad \left. - \log Z_i(\mathcal{C}_d, \mathbf{w}, \gamma) \right), \end{aligned} \tag{8}$$

where λ is regularization penalty and $Z'_i(\mathbf{w}, \gamma) = \sum_{0 \leq j < i} e^{\frac{1}{\gamma}(\mathbf{w} \cdot \phi(i,j) + \delta(\mathcal{C}_d, i, j))}$ is the normalization factor with a loss-augmented margin term $\delta(\mathcal{C}_d, i, j) = 1 - \mathcal{C}_d(i,j)$, which is 0 if i and j share the same cluster in $\mathcal{C}_d$, otherwise 1. One can think of adding the loss term to the normalization factor as similar in spirit to loss-augmented margin-based classifiers (Tsochantaridis et al., 2004). In fact, as γ approaches zero, our objective function in Eq. (8) approaches latent structural SVMs (LSSVM) (Yu & Joachims, 2009). For $\gamma = 1$, our approach resembles hidden variable conditional random fields (HCRF) (Quattoni et al., 2007). Thus by tuning γ, we consider a learning technique more general than LSSVM and HCRF (see Schwing et al. (2012) for more details.)

Stochastic (sub)gradient based optimization: The objective function in (8) is non-convex and hence is intractable to minimize exactly. With finitely sized training data streams, one can use the Concave-Convex Procedure (CCCP) (Yuille & Rangarajan, 2003) which reaches a local minimum. However, we choose to follow a fast stochastic gradient descent (SGD) strategy, based on the fact that $LL(\mathbf{w})$ decomposes not only over training data streams, but also over individual items in each data stream. In particular, using (3) and (8), we can re-write $LL(\mathbf{w})$ as

$$\begin{aligned} LL(\mathbf{w}) = \tfrac{1}{|D|} \sum_{d \in D} \sum_{i=1}^{m_d} \left(\tfrac{\lambda}{2 m_d} \|\mathbf{w}\|^2 - \log \left(\sum_{0 \leq j < i} e^{\frac{1}{\gamma}(\mathbf{w} \cdot \phi(i,j))} \mathcal{C}_d(i,j) \right) \right. \\ \left. + \log \left(\sum_{0 \leq j < i} e^{\frac{1}{\gamma}(\mathbf{w} \cdot \phi(i,j) + \delta(\mathcal{C}_d, i, j))} \right) \right). \end{aligned} \tag{9}$$

Due to this decomposition, we can compute a SGD on a per-item basis rather than per-data stream basis. So we do not have to wait to perform marginal inference over an entire data stream (which could be potentially very large) to update our model — we can perform rapid SGD updates for each item. The stochastic (sub)gradient w.r.t. item i in data stream d is just a weighted sum of features of all left-links from i given by

$$\nabla LL(\mathbf{w})^i_d \propto \sum_{0 \leq j < i} p_j \phi(i,j) - \sum_{0 \leq j < i} p'_j \phi(i,j) + \frac{\lambda \gamma}{m_d} \mathbf{w}, \tag{10}$$

where p_j and p'_j are probability-like measures given by

$$p_j = Pr[i \to j; d, \mathbf{w}]$$

and

$$p'_j = \frac{\mathcal{C}_d(i,j) Z_i(\mathbf{w}, \gamma)}{Z_i(\mathcal{C}_d, \mathbf{w}, \gamma)} Pr[i \to j; d, \mathbf{w}] \ .$$

Intuitively, the above gradient update promotes a weighted sum of correct left-links from i and demotes a weighted sum of other left-links from i.

It is difficult to characterize the behavior (e.g. convergence) of SGD strategies for non-convex problems. However, SGD is known to be quite successful in practice when applied to many different non-convex learning problems (Guillory et al., 2009; LeCun et al., 1998). We observe that our SGD-based learning converges very quickly and will show in Sec. 4 that it gives great empirical performance. Theoretical characterization of our SGD approach in terms of convergence and improvement of the objective function remains an open problem.

Finally, note that for $\gamma = 0$, our stochastic gradient update algorithm is similar to the latent structured perceptron like algorithm used in Chang et al. (2012). Following Samdani et al. (2012a), we improve over this algorithm by tuning the value of γ using a development set.

4. Case Study: Coreference Resolution

In this section, we study the application of L^3M to coreference clustering. In particular, we study some of the competing approaches for coreference clustering and present experimental results on benchmark English corference datasets — ACE 2004 (NIST, 2004) and Ontonotes-5.0 (Pradhan et al., 2012).

We compare different systems on gold mentions (i.e. we use mentions provided by the dataset) in order to compare systems purely on coreference, unmitigated by errors in mention identification. For all the approaches, we uniformly use the same set of features given by Chang et al. (2012). We compare the systems using three different popular metrics for coreference — MUC (Vilain et al., 1995), BCUB (Bagga & Baldwin, 1998), and Entity-based CEAF (CEAF_e) (Luo, 2005). Following, the CoNLL shared tasks (Pradhan et al., 2012), we pick the averages of these three metric as the main metric of comparison. We tune the regularization penalty for all the models and the value of γ for L^3M to optimize this average over the development set.

4.1. Existing Competing Techniques

Below, we survey some of the existing discriminative supervised clustering approaches applied to coreference. We bifurcate the discussion between non-streaming techniques that have been used for coreference but require looking at all the mentions (i.e. items) together and streaming techniques that can be applied on mentions, one at a time.

4.1.1. Non-Streaming Clustering

Below, we discuss two existing structured prediction techniques for clustering that cluster all the mentions together.

All Link Clustering: Mccallum & Wellner (2003) and Finley & Joachims (2005) model coreference as a correlational clustering (Bansal et al., 2002) problem on a complete graph over the mentions with edge weights w_{ij} given by the pairwise classifier. Following Chang et al. (2011), we call this the *All-Link* approach as this approach scores a clustering of mentions by including all possible pairwise links on this graph.

For a given document d, we specify the target clustering $\mathcal{C}$ by a collection of binary variables $\{y_{ij} \in \{0,1\} | 1 \le i, j \le m_d, i \ne j\}$ where $y_{ij} \equiv \mathcal{C}(i,j)$, that is $y_{ij} = 1$ *if and only if* i and j are in the same cluster in $\mathcal{C}$ (y_{ij} and y_{ji} thus refer to the same variable.) For a document d, given a $\mathbf{w}$, *All-Link* inference finds a clustering by solving the following integer linear programming (ILP) optimization problem:

$$\begin{aligned} \arg\max_{y} \quad & \sum\nolimits_{i,j} w_{ij} y_{ij}, \qquad y_{ij} \in \{0,1\} \\ \text{s.t} \quad & y_{kj} \ge y_{ij} + y_{ki} - 1 \; \forall \text{ mentions } i,j,k \; . \end{aligned} \tag{11}$$

The inequality constraints in Eq. (11) enforce the transitive closure of the clustering. The solution of Eq. (11) is a set of clusters, and the mentions in the same cluster corefer. Correlational clustering is an NP Hard problem (Bansal et al., 2002) and we use an ILP solver in our implementation. While ILP-based All-Link works for the ACE data, it is too slow for a much larger OntoNotes data. Consequently, following Pascal & Baldridge (2009) and Chang et al. (2011) we consider a reduced and faster alternative to the All-Link ILP approach, *All-Link-Red.*, which drops one of the the three transitivity constraints for each triplet of mention variables. Finley & Joachims (2005) learn $\mathbf{w}$ in this setting using a structural SVM formulation, which we also use in our implementation.

Spanning forest Clustering: This approach was proposed by Yu & Joachims (2009). The key motivation for this approach is that most of the $\binom{m_d}{2}$ links considered by All-Link clustering may not contain any useful signal and the coreference decision may likely be figured out transitively after determining a few strong coreference links. Yu and Joachims propose to model these "strong" coreference links using a latent spanning forest. In particular, they posit that a given coreference clustering $\mathcal{C}$ is a result of taking a transitive closure of a spanning forest h — every cluster in $\mathcal{C}$ is a connected component (i.e. a tree) in h, and distinct clusters in $\mathcal{C}$ are not connected by any edge in h.

The task of inference in this case is to find the maximum weight spanning forest over a complete weighted graph connecting all the mentions, where edge (i,j) has weight w_{ij}. This inference can be performed using Kruskal's algorithm. Yu and Joachims learn the pairwise weights $\mathbf{w}$ using a latent structural SVM formulation which they optimize using the CCCP strategy (Yuille & Rangarajan, 2003).

4.1.2. Streaming Techniques for Coreference

We now discuss two existing clustering techniques that can cluster mentions or items appearing in a streaming order.

Best-Left-Link Clustering: The Best-Left-Link inference strategy, also described in Sec. 3, has been vastly successful and popular for coreference clustering (Soon et al., 2001; Ng & Cardie, 2002; Bengtson & Roth, 2008; Stoyanov et al., 2009). However, most works perform learning

in an ad hoc fashion, not relating it to inference in a principled way. For instance, Bengtson & Roth (2008) train $\mathbf{w}$ on binary training data generated by taking for each mention, the closest antecedent coreferent mention as a positive example, and all the other mentions in between as negative examples. No explanation is available as to why this is the right way to train. Other papers also use similar ad hoc techniques. In our experiments, we compare with the *IllinoisCoref* system (Chang et al., 2011) which is state-of-the-art in Best-Left-Link systems.

Sum-Link Clustering: This supervised streaming data clustering technique was proposed by Haider et al. (2007) for detecting batches of spam emails. To the best of our knowledge, we are the first to apply it to coreference. This technique is derived from the All-Link technique and is very related to L^{3}M. In particular, it considers the items or mentions in a streaming order, and when determining the score of connecting an item i to a cluster c, it adds the score of pairwise links from i to all items in c: $\sum_{j \in c} \mathbf{w} \cdot \phi(i, j)$. It connects i to the cluster with highest score if the score is greater than 0. Like L^{3}M, once an item is assimilated in a cluster, the cluster membership is never changed later. Haider et al. (2007) proposed an efficient qudratic programming based learning technique for this model.

At the first glance, there does not seem to be a substantial difference between this technique and L^{3}M as both combine weights obtained from multiple pairwise links between a given item i and a cluster c. However, there is a fundamental difference in terms of how the weights are combined. In particular, L^{3}M is a non-linear model and puts significantly more importance on high scoring links (through exponentiation) than average or low scoring links, whereas Sum-Link combines all the links linearly. For instance, consider a scenario where we want to determine whether to link an item i to a cluster c containing two items. For Sum-Link, the case when c contains a left-link with score 10 and a left-link with score -6 is the same as when c contains two links with score 2. However, L^{3}M will associate a significantly higher score on the former case than the latter. In fact, with $\gamma = 0$, L^{3}M only considers the best scoring links. We argue that L^{3}M is more suitable for streaming data clustering for coreference than Sum-Link as it is believed that only a few, and not all, mentions in a cluster are likely to be informative (Ng & Cardie, 2002) when clustering a new mention. We will experimentally show that L^{3}M significantly outperforms Sum-Link.

4.2. Experimental Results

In this section, we present experimental results on the ACE and OntoNotes datasets.

Technique	MUC	BCUB	CEAF$_e$	AVG
ACE 2004				
IllinoisCoref	76.02	81.04	77.6	78.22
All-Link	77.39	80.3	77.83	78.51
All-Link-Red.	77.45	81.1	77.57	78.71
Spanning	73.31	79.25	74.66	75.74
Sum-Link	72.7	78.75	76.42	75.96
L^{3}M ($\gamma = 0$)	77.57	81.77	78.15	79.16
L^{3}M (tuned γ)	**78.18**	**82.09**	**79.21**	**79.83**
OntoNotes-5.0				
IllinoisCoref	80.84	74.29	65.96	73.70
All-Link-Red.	83.72	75.59	64.00	74.44
Spanning	83.64	74.83	61.07	73.18
Sum-Link	83.09	77.17	65.8	75.35
L^{3}M ($\gamma = 0$)	83.44	78.12	64.56	75.37
L^{3}M (tuned γ)	**83.97**	**78.25**	**65.69**	**75.97**

Table 1. Performance on ACE 2004 and OntoNotes-5.0. IllinoisCoref is a Best-Left-Link system; All-Link and All-Link-Red. are based on correlational clustering; Spanning is based on latent spanning forest based clustering; Sum-Link is a streaming data clustering technique by Haider et al. (2007). Our proposed approach is L^{3}M— L^{3}M with tuned γ is when we tune the value of γ using a development set; L^{3}M ($\gamma = \mathbf{0}$) is with γ fixed to 0.

ACE 2004 Corpus ACE 2004 (NIST, 2004) data contains 443 documents. Bengtson & Roth (2008) split these documents into 268 training, 68 development, and 106 testing documents; this was subsequently used by other works and we use the same split. The results are presented in Tab. 1. Clearly, our L^{3}M approach outperforms all the competing baselines. In particular, L^{3}M with tuned γ is better than L^{3}M with $\gamma = 0$ by 0.7 points in terms of the average showing that considering multiple links is actually helpful. Also, as opposed to what is reported by Yu & Joachims (2009), the spanning forest approach performs worse than the All-Link approach. We think that this is because we compare the systems on different metrics than them and also because we use exact ILP inference for correlational clustering whereas Yu and Joachims used approximate greedy inference.

OntoNotes-5.0 Corpus OntoNotes-5.0 is the coref dataset used for CoNLL 2012 Shared Task (Pradhan et al., 2012). This data set is by far the largest annotated corpus on coreference — about 10 times larger than ACE. It consists of different kinds of documents — newswire, bible, broadcast transcripts, magazine articles, and web blogs. Since the actual test data for the shared task competition was never released, we use the provided development set

for testing, and split the provided training data into training and development sets. Furthermore, we train and validate separate models for different parts of the corpus (like newswire or bible).

Tab. 1 reports results on OntoNotes. Once again, our L^3M approaches outperforms all the other baselines and L^3M with tuned γ outperforms L^3M with γ fixed to 0.

5. Conclusions

We presented a feature-based discriminative latent variable model for clustering of streaming data. We used a temperature parameter to tune the entropy of the probability associated with different links. We proposed an efficient inference algorithm for our model, as well as proposed a learning algorithm that generalizes and interpolates between hidden variable CRF and latent structural SVM. Our learning algorithm uses stochastic gradients computed on a per-data item basis. We applied our model to the task of coreference resolution and showed that it outperforms the key existing structured prediction approaches as well as state-of-the-art streaming data clustering approaches.

Future work includes applying our model to more clustering applications and speeding up our inference routine so that it scales linearly with the number of items.

References

Bagga, A. and Baldwin, B. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, 1998.

Bansal, N., Blum, A., and Chawla, S. Correlation clustering. In *FOCS*, 2002.

Bengtson, E. and Roth, D. Understanding the value of features for coreference resolution. In *EMNLP*, 2008.

Chang, K.-W., Samdani, R., Rozovskaya, A., Rizzolo, N., Sammons, M., and Roth, D. Inference protocols for coreference resolution. In *CoNLL Shared Task*, 2011.

Chang, K.-W., Samdani, R., Rozovskaya, A., Sammons, M., and Roth, D. Illinois-coref: The ui system in the conll-2012 shared task. In *CoNLL Shared Task*, 2012.

Fernandes, E. R., dos Santos, C. N., and Milidiú, R. L. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, 2012.

Finley, T. and Joachims, T. Supervised clustering with support vector machines. In *ICML*, 2005.

Gimpel, K. and Smith, N. A. Softmax-margin CRFs: Training log-linear models with cost functions. In *NAACL*, 2010.

Guha, S., Meyerson, A., Mishra, N., Motwani, R., and O'Callaghan, L. Clustering data streams: Theory and practice. *IEEE Trans. on Knowl. and Data Eng.*, 2003.

Guillory, A., Chastain, E., and Bilmes, J. Active learning as nonconvex optimization. *JMLR*, 2009.

Haider, P., Brefeld, U., and Scheffer, T. Supervised clustering of streaming data for email batch detection. In Ghahramani, Zoubin (ed.), *ICML*, 2007.

LeCun, Y., Bottou, L., Orr, G., and Muller, K. Efficient backprop. In Orr, G. and K., Muller (eds.), *Neural Networks: Tricks of the trade*. Springer, 1998.

Luo, X. On coreference resolution performance metrics. In *EMNLP*, 2005.

Mccallum, A. and Wellner, B. Toward conditional models of identity uncertainty with application to proper noun coreference. In *NIPS*, 2003.

Ng, V. Supervised noun phrase coreference research: the first fifteen years. In *ACL*, 2010.

Ng, Vincent and Cardie, Claire. Improving machine learning approaches to coreference resolution. In *ACL*, 2002.

NIST. The ace evaluation plan., 2004. URL `http://www.itl.nist.gov/iad/mig//tests/ace/ace04/index.html`.

Pascal, D. and Baldridge, J. Global joint models for coreference resolution and named entity classification. In *Procesamiento del Lenguaje Natural*, 2009.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *CoNLL 2012*, 2012.

Quattoni, Ariadna, Wang, Sybor, Morency, Louis-Philippe, Collins, Michael, and Darrell, Trevor. Hidden conditional random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007. ISSN 0162-8828.

Samdani, R., Chang, M., and Roth, D. Unified expectation maximization. In *NAACL*, 2012a.

Samdani, R., Chang, M., and Roth, D. A framework for tuning posterior entropy in unsupervised learning. In *ICML workshop on Inferning: Interactions between Inference and Learning*, 2012b.

Schwing, A. G., Hazan, T., Pollefeys, M., and Urtasun, R. Efficient structured prediction with latent variables for general graphical models. In *ICML*, 2012.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 2001.

Stoyanov, V., Gilbert, N., Cardie, C., and Riloff, E. Conundrums in noun phrase coreference resolution: making sense of the state-of-the-art. In *ACL*, 2009.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, 1995.

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2002.

Yu, C. and Joachims, T. Learning structural svms with latent variables. In *ICML*, 2009.

Yuille, A. L. and Rangarajan, A. The concave-convex procedure. *Neural Computation*, 2003.