Unsupervised Information Extraction with Distributional Prior Knowledge

Cane Wing-ki Leung[1], **Jing Jiang**[1], **Kian Ming A. Chai**[2], **Hai Leong Chieu**[2], **Loo-Nin Teow**[2]
[1]School of Information Systems, Singapore Management University, Singapore
[2]DSO National Laboratories, Singapore
{caneleung,jingjiang}@smu.edu.sg, {ckianmin,chaileon,tloonin}@dso.org.sg

Abstract

We address the task of automatic discovery of information extraction template from a given text collection. Our approach clusters candidate slot fillers to identify meaningful template slots. We propose a generative model that incorporates distributional prior knowledge to help distribute candidates in a document into appropriate slots. Empirical results suggest that the proposed prior can bring substantial improvements to our task as compared to a K-means baseline and a Gaussian mixture model baseline. Specifically, the proposed prior has shown to be effective when coupled with discriminative features of the candidates.

1 Introduction

Information extraction (IE) is the task of extracting information from natural language texts to fill a database record following a structure called a *template*. Such templates are usually defined based on the domain of interest. For example, the domain in the Sixth Message Understanding Conference (MUC-6, 1995) is management succession, and the pre-defined template consists of the slots *position*, the *person leaving*, the *person joining*, and the *organization*.

Previous research on IE often requires the predefinition of templates. Template construction is usually done manually by domain experts, and annotated documents are often created to facilitate supervised learning approaches to IE. However, both manual template construction and data annotation are labor-intensive. More importantly, templates and annotated data usually cannot be re-used in new domains due to domain dependency. It is therefore natural to consider the problem of unsupervised template induction and information extraction. This is the topic of this paper.

There have been a few previous attempts to address the unsupervised IE problem (Shinyama and Sekine, 2006; Sekine, 2006; Rosenfeld and Feldman, 2006; Filatova et al., 2006). These approaches have a commonality: they try to cluster candidate slot fillers, which are often nouns and noun phrases, into slots of the template to be constructed. However, most of them have neglected the following important observation: a single document or text segment tends to cover different slots rather than redundantly fill the same slot. In other words, during clustering, candidates within the same text segment should be more likely to be distributed into different clusters.

In this paper, we propose a generative model that incorporates this distributional prior knowledge. We define a prior distribution over the possible label assignments in a document or a text segment such that a more diversified label assignment is preferred. This prior is based on the Poisson distribution. We also compare a number of generative models for generating slot fillers and find that the Gaussian mixture model is the best. We then combine the Poisson-based label assignment prior with the Gaussian mixture model to perform slot clustering. We find that compared with a K-means baseline and a Gaussian mixture model baseline, our combined model with the proposed label assignment prior substantially performs better on two of the three data sets we use for evaluation. We further analyze the results on the third data set and find that the proposed prior will have little effect if there are no good discriminative features to begin with. In summary, we find that

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 814–824,
Edinburgh, Scotland, UK, July 27–31, 2011.

our Poisson-based label assignment prior is effective when coupled with good discriminative features.

2 Related Work

One common approach to unsupervised IE is based on automatic IE pattern acquisition on a cluster of similar documents. For instance, Sudo et al. (2003) and Sekine (2006) proposed different methods for automatic IE pattern acquisition for a given domain based on frequent subtree discovery in dependency parse trees. These methods leveraged heavily on the entity types of candidates when assigning them to template slots. As a consequence, potentially different semantic roles of candidates having the same entity type could become indistinguishable (Sudo et al., 2003; Sekine, 2006). This problem is alleviated in our work by exploiting distributional prior knowledge about template slots, which is shown effective when coupled with discriminative features of candidates. Filatova et al. (2006) also considered frequent subtrees in dependency parse trees, but their goal was to build templates around verbs that are statistically important in a given domain. Our work, in contrast, is not constrained to verb-centric templates. We aim to identify salient slots in the given domain by clustering.

Marx et al. (2002) proposed the cross-component clustering algorithm for unsupervised IE. Their algorithm assigned a candidate from a document to a cluster based on the candidate's feature similarity with candidates from *other documents only*. In other words, the algorithm did not consider a candidate's relationships with other candidates in the same document. Our work is based on a different perspective: we model label assignments for all candidates in the same document with a distributional prior that prefers a document to cover more distinct slots. We show empirically that this prior improves slot clustering results greatly in some cases.

Also related to our work is open domain IE, which aims to perform unsupervised *relation extraction*. TEXTRUNNER (Banko et al., 2007), for example, automatically extracts *all possible relations* between pairs of noun phrases from a given corpus. The main difference between open domain IE and our work is that open domain IE does not aim to induce domain templates, whereas we focus on a single domain with the goal of inducing a template that describes salient information structure of that domain. Furthermore, TEXTRUNNER and related studies on unsupervised relation extraction often rely on highly redundant information on the Web or in large corpus (Hasegawa et al., 2004; Rosenfeld and Feldman, 2006; Yan et al., 2009), which is not assumed in our study.

We propose a generative model with a distributional prior for the unsupervised IE task, where slot fillers correspond to *observations* in the model, and their labels correspond to *hidden variables* we want to learn. In the machine learning literature, researchers have explored the use of similar prior knowledge in the form of *constraints* through model expectation. For example, Graça et al. (2007) proposed to place constraints on the posterior probabilities of hidden variables in a generative model, while Druck et al. (2008) studied a similar problem in a discriminative, semi-supervised setting. These studies model constraints as features, and enforce the constraints through expected feature values. In contrast, we place constraints on label assignments through a probabilistic prior on the distribution of slots. The proposed prior is simple and easy to interpret in a generative model. Nevertheless, it will be interesting to explore how the proposed prior can be implemented within the posterior constraint framework.

3 Problem Overview

In this section, we first formally define our unsupervised IE problem. We then provide an overview of our solution, which is based on a generative model.

3.1 Problem Definition

We assume a collection of documents or short text segments from a certain domain. These documents or text segments describe different events or entities, but they are about the same topic or aspect of the domain. Examples of such collections include a collection of sentences describing the educational background of famous scientists and a collection of aviation incident reports. Our task is to automatically discover an IE template from this collection. The discovered template should contain a set of slots that play different semantic roles in the domain.

Input text:

Topic: Graduate Student Seminar Lunch
Dates: 13-Apr-95
Time: 12:00 PM - 1:30 PM
PostedBy: Edmund J. Delaney on 5-Apr-95 at 16:24 from andrew.cmu.edu
Abstract:
The last Graduate Student Seminar Series lunch will be held on Thursday, April 13 from noon-1:30 p.m. in room 207, Student Activities Center. Professor Sara Kiesler of SDS will speak on Carving A Successful Research Niche.

Output:

Slot	Slot Filler(s)
Slot 1 (*start time*)	12:00PM
Slot 2 (*end time*)	1:30PM, 1:30 p.m.
Slot 3 (*location*)	room 207, Student Activities Center
Slot 4 (*speaker*)	Professor Sara Kiesler
Slot 4 (*irrelevant information*)	Edmund J. Delaney

Figure 1: An input text from a seminar announcement collection and the discovered IE template. Note that the slots are automatically discovered and the slot names are manually assigned.

To construct such a template, we start with identifying candidate slot fillers, hereafter referred to as *candidates*, from the input text. Then we cluster these candidates with the aim that each cluster will represent a semantically meaningful slot. Figure 1 gives an example of an input text from a collection of seminar announcements and the resulting template discovered from the collection. As we can see, the template contains some semantically meaningful slots such as the *start time*, *end time*, *location* and *speaker* of a seminar. Moreover, it also contains a slot that covers an irrelevant candidate. We call such slots covering irrelevant candidates *garbage slots*.

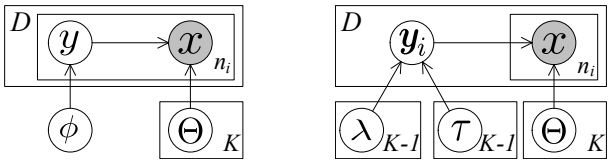
We can make two observations on the mapping from candidates to template slots from real data, such as the text in Figure 1. Firstly, a template slot may be filled by more than one candidate from a single document, although this number has been observed to be small. For example, the template slot *end time* in Figure 1 has two slot fillers: "1:30 PM" from the semi-structured header and "1:30 p.m." from within the abstract. Secondly, a document tends to contain candidates that cover different template slots. We believe that this observation is a consequence of the fact that a document will tend to convey as much information as possible. We further exploit these observations in Section 4.

3.2 A General Solution

Recall that our general solution to the unsupervised IE problem is to cluster candidate slot fillers in order to identify meaningful slots. We leave the details of how to extract the candidates to Section 7.1. In this section, we assume that we have a set of candidates $\boldsymbol{x} = \{x_{i,j}\}$, where $x_{i,j}$ is the j-th candidate from the i-th document in the collection. We cluster these candidates into K groups for a given K.

Let $y_{i,j} \in \{1, \ldots, K\}$ denote the cluster label for $x_{i,j}$ and $\boldsymbol{y}$ denote the set of all the $y_{i,j}$'s. Let $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ denote the sets of all the $x_{i,j}$'s and the $y_{i,j}$'s in the i-th document respectively. We assume a generative model for $\boldsymbol{x}$ and $\boldsymbol{y}$ as follows. For the i-th document in our collection, we assume that the number of candidates is known and we draw a label assignment $\boldsymbol{y}_i$ according to some distribution parameterized by Λ. Then for the j-th candidate, we generate $x_{i,j}$ from $y_{i,j}$ according to a generative model parameterized by Θ. Since the labels $\boldsymbol{y}$ are hidden, the observed log-likelihood of the parameters given the observations $\boldsymbol{x}$ is

$$\begin{aligned} L(\Lambda, \Theta) &= \log p(\boldsymbol{x}; \Lambda, \Theta) \\ &= \sum_i \log \sum_{\boldsymbol{y}_i} p(\boldsymbol{x}_i, \boldsymbol{y}_i; \Lambda, \Theta) \\ &= \sum_i \log \sum_{\boldsymbol{y}_i} p(\boldsymbol{y}_i; \Lambda) \prod_j p(x_{i,j} | y_{i,j}; \Theta). \quad (1) \end{aligned}$$



(a) A multinomial prior. (b) The proposed Poisson-based prior.

Figure 2: Generative models with different label assignment priors. D denotes the number of documents in the given collection, n_i denotes the number of candidates in the i-th document, and K is the number of slots (clusters).

For a given functional form of $p(\boldsymbol{y}_i; \Lambda)$ and $p(x_{i,j}|y_{i,j}; \Theta)$, the best model parameters can be estimated by maximizing Eq. (1). In the next section, we detail two designs of the prior $p(\boldsymbol{y}_i; \Lambda)$, followed by different generative models for the distribution $p(x_{i,j}|y_{i,j}; \Theta)$ in Section 5. Then we describe the estimation of model parameters in Section 6.

4 Label Assignment Prior

The label assignment prior, $p(\boldsymbol{y}_i; \Lambda)$, models the generation of labels for candidates in a document. In this section, we first describe a commonly used multinomial prior, and then introduce the proposed Poisson-based prior for the unsupervised IE task.

4.1 A Multinomial Prior

Usually, one would assume that the labels for the different candidates in the same document are generated independently, that is, $p(\boldsymbol{y}_i; \Lambda) = \prod_j p(y_{i,j}; \Lambda)$. Under this model, we assume that each $y_{i,j}$ is generated from a multinomial distribution parameterized by $\boldsymbol{\phi}$, where ϕ_y denotes the probability of generating label y. Our objective function in Eq. (1) then becomes:

$$\begin{aligned} L(\Lambda, \Theta) &= \log p(\boldsymbol{x}; \Lambda, \Theta) \\ &= \sum_{i,j} \log \sum_y \phi_y p(x_{i,j}|y; \Theta). \end{aligned} \quad (2)$$

Figure 2(a) depicts a generative model with this multinomial prior in plate notation. Note that the independence assumption on label assignment in this model does not capture our observation that candidates in a document are likely to cover different semantic roles.

4.2 The Proposed Poisson-based Prior

We propose a prior distribution that favors more diverse label assignments. Our proposal takes into consideration the following three observations. Firstly, candidates in the same document are likely to cover different semantic roles. The proposed prior distribution should therefore assign higher probability to a label assignment that covers more distinct slots. Secondly, the same piece of information is not likely to be repeated many times in a document. Our design thus allows a slot to generate multiple fillers in a document, up to a limited number of times. Thirdly, there may exist candidates that do not belong to slots in the extracted template. Therefore, we introduce a *dummy slot* or *garbage slot* to the label set to collect such candidates. Yet, we shall not assume any prior/domain knowledge about candidates generated by the garbage slot as they are essentially irrelevant in the given domain.

We now detail the prior that exploits the above observations. First, we fix the K-th slot (or cluster) in the label set to be the garbage slot. For each of the non-garbage slot $k = 1, \ldots, K-1$, we also fix the maximum number of fillers that can be generated, which we denote by λ_k. There is no λ_K for the garbage slot because the number of fillers is not constrained for this slot. This allows all candidates in a document to be generated by the garbage slot. Let n_i be the number of candidates in the i-th document. Given K, $\{\lambda_k\}_{k=1}^{K-1}$ and n_i, the set of possible label assignments for the i-th document can be generated. We illustrate this with an example. Let $K = 2$ and $\lambda_1 = 1$. The label set is $\{1, 2\}$, where 2 represents the garbage slot. Let the number of candidates be $n_i = 2$. The possible label assignments within this setting are $(1, 2)$, $(2, 1)$ and $(2, 2)$.

The set of possible label assignments for the i-th document is the sample space on which we place the prior distribution $p(\boldsymbol{y}_i; \Lambda)$. We need a prior that gives a higher probability to a more diverse label assignment. For a given $\boldsymbol{y}_i$ for the i-th document, let $n_{i,k}$ be the number of candidates in the document that have been assigned to slot k. That is, $n_{i,k} \stackrel{\text{def}}{=} \sum_{j=1}^{n_i} \mathbf{1}(y_{i,j} = k)$, where $\mathbf{1}(\cdot)$ is the indicator variable. We propose the following distribution based on the Poisson distribution:

$$p(\boldsymbol{y}_i;\Lambda) \stackrel{\text{def}}{=} Z_i^{-1} \prod_{k=1}^{K-1} \text{Poisson}(n_{i,k};\tau_k), \quad (3)$$

where Z_i is the normalizing constant, and τ_k is the mean parameter of the k-th Poisson distribution, $k = 1, \ldots K-1$. The absence of a factor that depends on $n_{i,K}$ reflects the lack of prior knowledge on the number of garbage slot fillers. Figure 2(b) depicts the proposed generative model with the Poisson-based prior in plate notation.

5 Generating Slot Fillers

Different existing generative models can be used to model the generation of a slot filler given a label, that is, $p(x|y;\Theta)$. We explore four of them for our task, namely, the naive Bayes model, the Bernoulli mixture model, the Gaussian mixture model, and a locally normalized logistic regression model proposed by Berg-Kirkpatrick et al. (2010).

5.1 Multinomial Naive Bayes

In the multinomial naive Bayes model, features of an observation x are assumed to be independent and each generated from a multinomial distribution. We first introduce some notations. Let f denote a feature (e.g. entity type) and $\mathcal{V}_f$ denote the set of possible values for f. Let $x^f \in \mathcal{V}_f$ be the value of feature f in x (e.g. *person*). For a given label y, feature f follows a multinomial distribution parameterized by $\boldsymbol{\psi}_{y,f}$, where $\psi_{y,f,v}$ denotes the probability of feature f taking the value $v \in \mathcal{V}_f$ given label y. The functional form of the conditional probability of x given a label y is then

$$p(x|y;\Theta) = \prod_f p(x^f|y;\Theta) = \prod_f \psi_{y,f,x^f}. \quad (4)$$

5.2 Bernoulli Mixture Model

In the naive Bayes model our features are defined to be categorical. For the Bernoulli mixture model, as well as the Gaussian mixture model and the locally normalized logistic regression model in the next subsections, we first convert each observation x into a binary feature vector $\mathbf{x} \in \{0,1\}^F$ where F is the number of binary features. An example of a binary features is "the entity type is *person*".

We assume that, for a given label y, observations are generated from a multivariate Bernoulli distribution parameterized by $\boldsymbol{\varphi}_{y,f}$, where $\varphi_{y,f,v}$ denotes the probability of feature f taking the value $v \in \{0,1\}$ given label y. The conditional probability of $\mathbf{x}$ given y can then be written as

$$\begin{aligned} p(\mathbf{x}|y;\Theta) &= \prod_f p(x_f = 1|y;\Theta)^{x_f} \cdot p(x_f = 0|y;\Theta)^{1-x_f} \\ &= \prod_f \varphi_{y,f,x^f}. \end{aligned} \quad (5)$$

5.3 Gaussian Mixture Model

In the Gaussian mixture model, we assume that a given label y generates observations with a multivariate Gaussian distribution $\mathcal{N}(\mu_y, \Sigma_y)$, where $\mu_y \in \mathbb{R}^F$ is the mean and $\Sigma_y \in \mathbb{R}^{F\times F}$ is the covariance matrix of the Gaussian. If we assume that the different feature dimensions are independent and have the same variance, that is, $\Sigma_y = \sigma_y^2 I$, where I is the identity matrix, then the conditional density of $\mathbf{x}$ given y is

$$p(\mathbf{x}|y;\Theta) = \frac{1}{(2\pi\sigma_y^2)^{F/2}} \exp\left(-\frac{\|\mathbf{x}-\mu_y\|^2}{2\sigma_y^2}\right). \quad (6)$$

5.4 Locally Normalized Logistic Regression

Berg-Kirkpatrick et al. (2010) proposed a method for incorporating features into generative models for unsupervised learning. Their method models the generation of $\mathbf{x}$ given y as a logistic function parameterized by a weight vector $\mathbf{w}_y$, defined as follows:

$$p(\mathbf{x}|y;\Theta) = \frac{\exp\langle \mathbf{x}, \mathbf{w}_y\rangle}{\sum_{\mathbf{x}'} \exp\langle \mathbf{x}', \mathbf{w}_y\rangle}. \quad (7)$$

$\langle \mathbf{x}, \mathbf{w}\rangle$ denotes the inner product between $\mathbf{x}$ and $\mathbf{w}$. The denominator considers all data points $\mathbf{x}'$ in the data set, thus Eq. (7) gives a probability distribution over data points for a given y.

6 Parameter Estimation

We can apply the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to maximize the log-likelihood functions under both multinomial prior in Eq. (2) and the proposed Poisson-based prior in Eq. (1). For the multinomial prior, there are standard closed form solutions for the naive Bayes, the Bernoulli mixture and the Gaussian mixture models. For locally normalized logistic regression, model parameters can also be learned via EM, but with a gradient-based M-step (Berg-Kirkpatrick et al., 2010). We leave out the details here and focus on parameter estimation in the proposed generative model with the Poisson-based prior.

We assume that in the Poisson-based prior, the parameters $\{\lambda_k\}_{k=1}^{K-1}$ and $\{\tau_k\}_{k=1}^{K-1}$ are fixed rather than learned in this work. For the distribution $p(x|y;\Theta)$, let $\Theta^{(t-1)}$ and $\Theta^{(t)}$ denote parameter estimates from two consecutive EM iterations. At the t-th iteration, the E-step updates the responsibilities of each label assignment $\boldsymbol{y}_i$ for each document:

$$\begin{aligned}\alpha_{i,\boldsymbol{y}_i} &= p(\boldsymbol{y}_i|\boldsymbol{x_i};\Lambda,\Theta^{(t-1)}) \\ &= \frac{p(\boldsymbol{y}_i;\Lambda)p(\boldsymbol{x}_i|\boldsymbol{y}_i;\Theta^{(t-1)})}{\sum_{\boldsymbol{y}'_i} p(\boldsymbol{y}'_i;\Lambda)p(\boldsymbol{x}_i|\boldsymbol{y}'_i;\Theta^{(t-1)})}, \quad (8)\end{aligned}$$

where α_i is a distribution over all possible label assignments $\boldsymbol{y}_i$'s for the i-th document. The M-step updates the estimates of $\Theta^{(t)}$ based on the current values of α_i's and $\Theta^{(t-1)}$. This is done by maximizing the following objective function:

$$\sum_i \sum_{\boldsymbol{y}_i} \alpha_{i,\boldsymbol{y}_i} \log\Big(p(\boldsymbol{y}_i;\Lambda)\prod_j p(x_{i,j}|y_{i,j};\Theta^{(t-1)})\Big). \quad (9)$$

The exact formulas used in the M-step for updating Θ depend on the functional form of $p(x_{i,j}|y_{i,j};\Theta)$. As an example, we give the formulas for the Gaussian mixture model, in which Θ contains the set of means $\{\mu_k^{(t)}\}_{k=1}^K$ and variances $\{\sigma_k^{(t)}\}_{k=1}^K$. Taking the derivatives of Eq. (9) with respect to μ_k and to σ_k, and then setting the derivations to zero, we can solve for μ_k and for σ_k to get:

$$\mu_k^{(t)} = \frac{\sum_i \sum_{\boldsymbol{y}_i} \alpha_{i,\boldsymbol{y}_i} \sum_j \mathbf{1}(y_{i,j}=k)\mathbf{x}_{i,j}}{\sum_i \sum_{\boldsymbol{y}_i} \alpha_{i,\boldsymbol{y}_i} \sum_j \mathbf{1}(y_{i,j}=k)}, \quad (10)$$

$$\sigma_k^{(t)} = \frac{\sum_i \sum_{\boldsymbol{y}_i} \alpha_{i,\boldsymbol{y}_i} \sum_j \mathbf{1}(y_{i,j}=k)||\mathbf{x}_{i,j}-\mu_k^{(t)}||^2}{F\sum_i \sum_{\boldsymbol{y}_i} \alpha_{i,\boldsymbol{y}_i} \sum_j \mathbf{1}(y_{i,j}=k)}, \quad (11)$$

where $\mathbf{1}(\cdot)$ is the indicator variable. We skip the derivations here due to space limit.

Closed form solutions also exist for the naive Bayes and the Bernoulli mixture models. For locally normalized logistic regression, parameters can be learned with a gradient-based M-step as in the multinomial prior setting. Existing optimization algorithms, such as L-BFGS, can be used for optimizing model parameters in the M-step as discussed in (Berg-Kirkpatrick et al., 2010).

7 Experiments

In this section, we first describe the data sets we used in our experiments, detailing the target slots and candidates in each data set, as well as features we extract for the candidates. We then describe our evaluation metrics, followed by experimental results.

7.1 Data Sets

We use three data sets for evaluating our unsupervised IE task. Note that to speed up computation, we only include documents or text segments containing no more than 10 candidates in our experiments. The first data set contains a set of seminar announcements (Freitag and McCallum, 1999), annotated with four slot labels, namely *stime* (start time), *etime* (end time), *speaker* and *location*. We used as candidates all strings labeled in the annotated data as well as all named entities found by the Stanford NER tagger for CoNLL (Finkel et al., 2005). There are 309 seminar announcements with 2262 candidates in this data set.

The second data set is a collection of paragraphs describing aviation incidents, taken from the Wikipedia article on "List of accidents and incidents involving commercial aircraft" (Wikipedia, 2009). Each paragraph in the article contains one to a few sentences describing an incident. In this domain, we take each paragraph as a separate document, and all hyperlinked phrases in the original Wikipedia article as candidates. For evaluation, we manually annotated the paragraphs of incidents from 2006 to 2009 with five slot labels: the *flight number* (FN), the *airline* (AL), the *aircraft model* (AC), the *exact*

location (LO) of the incident (e.g. airport name), and the *country* (CO) where the incident occurred. The entire data set consists of 564 paragraphs with 2783 candidates. The annotated portion consists of 74 paragraphs with 395 candidates.

The third data set comes from the management succession domain used in the Sixth Message Understanding Conference (MUC-6, 1995). We extract from the original data set all sentences that were tagged with a management succession event, and use as candidates all tagged strings in those sentences. This domain has four target slots, namely *PersonIn* (the person moving into a new position), *PersonOut* (the person leaving a position), *Org* (the corporation's name) and *Post* (the position title). Sentences containing candidates with multiple labels (candidates annotated as both *PersonIn* and *PersonOut*) are discarded. The extracted data set consists of 757 sentences with 2288 candidates.

7.2 Features

To extract features for candidates, we first normalize each word to its lower-case, with digits replaced by the token *digit*. We extract the following features for every candidate: the candidate phrase itself, its head word, the unigram and bigram before and after the candidate in the sentence where it appeared, its entity type (*person*, *location*, *organization*, and *date/time*), as well as features derived from dependency parse trees. Specifically, we first apply the Stanford lexical parser to our data (de Marneffe et al., 2006). Then for each candidate, we follow its dependencies in the corresponding dependency parse tree until we find a relation $r \in \{nsubj, csubj, dobj, iobj, pobj\}$ in which the candidate is the dependent. We then construct a feature (r, v) where v is governor of the relation.

7.3 Evaluation Baseline and Method

We use the standard K-means algorithm (Macqueen, 1967) as a non-generative baseline, since K-means is commonly used for clustering. To evaluate clustering results, we match each slot in the labeled data to the cluster that gives the best F1-measure when evaluated for the slot. We report the precision (P), recall (R) and F1-measure for individual slot labels, as well as the macro- and micro- average results across all labels for each experiment. We conduct 10 trials of experiment on each model and each data set with different random initializations. We report the trials that give the smallest within-cluster sum-of-squares (WCSS) distance for K-means, and those that give the highest log-likelihood of data for all other models. Experimental trials are run until the change in WCSS/log-likelihood between two EM iterations is smaller than 1×10^{-6}. All trials converged within 30 minutes.

All models we evaluate involve a parameter K, which is the number of values that y can take on. The value of K is manually fixed in this study. As noted, we use a garbage slot to capture irrelevant candidates, thus the value of K is set to the number of target slots plus 1 for each data set. We empirically set the adjustable parameters in the proposed prior, and the weight of the regularization term in the locally normalized logistic regression model (Berg-Kirkpatrick et al., 2010), denoted by β. Exact settings are given in the next subsection. Note that the focus of our experiments is on evaluating the effectiveness of the proposed prior. We leave the task of learning the various parameter values to future work.

7.4 Results

Evaluation on existing generative models

We first evaluate the existing generative models described in Section 5 with the multinomial prior. Table 1 summarizes the performance of Naive Bayes (NB), the Bernoulli mixture model (BMM), the Gaussian mixture model (GMM), the locally normalized logistic regression (LNLR) model, and K-means. We only show the F1 measures in the table due to space limit.

We first observe that NB does not perform well for our task. LNLR, which is an interesting contribution in its own right, does not seem to be suitable for our task as well. While NB and LNLR are inferior to K-means for all three data sets, BMM shows mixed results. Specifically, BMM outperforms K-means for aviation incidents, but performs poorly for seminar announcements. GMM and K-means achieve similar results, which is not surprising because K-means can be viewed as a special case of the spherical GMM we used (Duda et al., 2001).

Overall speaking, results show that GMM is the best among the four generative models for the distri-

(a) Results on **seminar announcements**. No macro- and micro-average result is reported for NB and BMM as they merged the *etime* cluster with the *stime* cluster. Numbers in brackets are the respective measures of the *stime* cluster when evaluated for *etime*.

Model	*stime*	*etime*	*speaker*	*location*	Macro-avg	Micro-avg	Parameter
NB	0.558	(0.342)	0.276	0.172	—	—	Nil
BMM	0.822	(0.440)	0.412	0.402	—	—	Nil
GMM	0.450	0.530	0.417	0.426	0.557	0.455	Nil
LNLR	0.386	0.239	0.200	0.208	0.264	0.266	β = .0005
K-means	0.560	0.574	0.335	0.426	0.538	0.452	Nil

(b) Results on **aviation incidents**. Target slots are *airline (AL)* , *flight number (FN)*, *aircraft model (AC)*, *location (LO)* and *country (CO)*.

Model	*AL*	*FN*	*AC*	*LO*	*CO*	Macro-avg	Micro-avg	Parameter
NB	0.896	0.473	0.676	0.504	0.533	0.618	0.628	Nil
BMM	0.862	0.794	0.656	0.695	0.614	0.741	0.724	Nil
GMM	0.859	0.914	0.635	0.576	0.538	0.730	0.692	Nil
LNLR	0.597	0.352	0.314	0.286	0.291	0.379	0.396	β = .0005
K-means	0.859	0.936	0.661	0.576	0.538	0.729	0.701	Nil

(c) Results on **management succession events**. Target slots are *person joining (PersonIn)*, *person leaving (PersonOut)*, *organization (Org)*, and *position (Post)*.

Model	*PersonIn*	*PersonOut*	*Org*	*Post*	Macro-avg	Micro-avg	Parameter
NB	0.545	0.257	0.473	0.455	0.459	0.437	Nil
BMM	0.550	0.437	0.800	0.767	0.650	0.648	Nil
GMM	0.583	0.432	0.813	0.803	0.679	0.676	Nil
LNLR	0.419	0.245	0.319	0.399	0.351	0.346	β = .0002
K-means	0.372	0.565	0.835	0.814	0.645	0.665	Nil

Table 1: Performance summary of the different generative models and K-means in terms of F1.

Data set	Parameter	Value
Seminar announcements	$\{\lambda_k\}_{k=1}^4$	$\{2\}_{k=1}^4$
	$\{\tau_k\}_{k=1}^4$	$\{1\}_{k=1}^4$
Aviation incidents	$\{\lambda_k\}_{k=1}^5$	$\{1\}_{k=1}^5$
	$\{\tau_k\}_{k=1}^5$	$\{1\}_{k=1}^5$
Management succession	$\{\lambda_k\}_{k=1}^4$	{1,2,2,2}
	$\{\tau_k\}_{k=1}^4$	{1,2,2,2}

Table 2: Parameter settings for $p(\boldsymbol{y}_i; \Lambda)$.

bution $p(x|y; \Theta)$. We proceed with incorporating the proposed prior into GMM for further explorations.

Effectiveness of the proposed prior

We evaluate the effectiveness of the proposed prior by combining it with GMM. Specifically, the combined model follows Eq. (1), with $p(\boldsymbol{y}_i; \Lambda)$ computed using the Poisson-based formula in Eq. (3) and $p(x_{i,j}|y_{i,j}; \Theta)$ following Eq. (6) as in GMM.

We empirically determine the parameters used in $p(\boldsymbol{y}_i; \Lambda)$ to maximize data's log-likelihood as noted. Table 2 reports the values of $\{\lambda_k\}_{k=1}^{K-1}$ and $\{\tau_k\}_{k=1}^{K-1}$ for different data sets. Recall that λ_k specifies the *maximum* number of candidates that the k-th slot can generate, and its value is observed to be small in real data. τ_k specifies the *expected* number of candidates that the k-th slot will generate.

Table 3 reports the performance of the combined model ("GMM with prior") on the three data sets, along with results of GMM and K-means for easy comparison. The combined model improves over both GMM and K-means for seminar announcements and aviation incidents, as can be seen from the models' macro- and micro-average performance.

The advantages brought by the proposed prior are mainly reflected in slots that are difficult to cluster under GMM and K-means. Taking seminar announcements as an example, GMM and K-means achieve high precision but low recall for *stime*, and low precision but high recall for *etime*. When examining the clusters produced by these two models, we found one small cluster that contains mostly *stime* fillers (thus high precision but low recall), and another much larger cluster that contains mostly *etime* fillers together with most of the remaining *stime* fillers (thus low precision but high recall for *etime*).

(a) Results on **seminar announcements**.

Model	Metric	*stime*	*etime*	*speaker*	*location*	Macro-avg	Micro-avg
GMM with Prior	P	0.964	0.983	0.232	0.253	0.608	0.416
	R	0.680	0.932	0.952	0.481	0.761	0.738
	F1	0.798	0.957	0.374	0.331	0.676	0.532
GMM	P	1.000	0.362	0.300	0.436	0.524	0.407
	R	0.291	0.984	0.686	0.416	0.594	0.518
	F1	0.450	0.530	0.417	0.426	0.557	0.455
K-means	P	0.890	0.434	0.222	0.436	0.496	0.389
	R	0.408	0.847	0.679	0.416	0.588	0.541
	F1	0.560	0.574	0.335	0.426	0.538	0.452

(b) Results on **aviation incidents**.

Model	Metric	*AL*	*FN*	*AC*	*LO*	*CO*	Macro-avg	Micro-avg
GMM with Prior	P	1.000	1.000	1.000	0.741	0.833	0.915	0.908
	R	0.753	0.877	0.465	0.588	0.727	0.682	0.673
	F1	0.859	0.935	0.635	0.656	0.777	0.782	0.773
GMM	P	1.000	1.000	1.000	0.563	0.433	0.799	0.724
	R	0.753	0.842	0.465	0.588	0.709	0.672	0.664
	F1	0.859	0.914	0.635	0.576	0.538	0.730	0.692
K-means	P	1.000	0.981	0.830	0.563	0.433	0.761	0.711
	R	0.753	0.895	0.549	0.588	0.709	0.699	0.691
	F1	0.859	0.936	0.661	0.576	0.538	0.729	0.701

(c) Results on management succession events.

Model	Metric	*PersonIn*	*PersonOut*	*Org*	*Post*	Macro-avg	Micro-avg
GMM with Prior	P	0.458	0.610	0.720	0.774	0.640	0.642
	R	0.784	0.352	0.969	0.846	0.738	0.731
	F1	0.578	0.447	0.826	0.809	0.686	0.683
GMM	P	0.464	0.605	0.725	0.792	0.647	0.648
	R	0.782	0.336	0.925	0.815	0.715	0.707
	F1	0.583	0.432	0.813	0.803	0.679	0.676
K-means	P	0.382	0.515	0.733	0.839	0.607	0.639
	R	0.363	0.625	0.969	0.791	0.687	0.693
	F1	0.372	0.565	0.835	0.814	0.645	0.665

Table 3: Comparison between the combined model (GMM with the proposed prior), GMM and K-means.

This shows that GMM, when used with the multinomial prior, and K-means have difficulties separating candidates from these two slots. In contrast, the combined model improves the recall of *stime* to 68%, as compared to 29.1% achieved by GMM with the multinomial prior and 40.8% by K-means, without sacrificing precision. It also improves the precision of *etime* from 36.2% to 98.3%.

For aviation incidents, the advantage of the proposed prior is reflected in the *location (LO)* and *country (CO)* slots, which may confuse the various models as they both belong to the entity type *location*. The proposed prior improves the precision of these two slots greatly by trying to distribute them into appropriate slots in the clustering process.

The three models achieve very similar performance on management succession events as Table 3(c) shows. Surprisingly, incorporating the Poisson-based prior into GMM does not seem useful in separating *PersonIn* and *PersonOut* slot fillers. To investigate the possible reasons for this, we examine feature values in the centriods of the two clusters learned by the three models.

Tables 4 and 5 respectively list the top-10 features in the *PersonIn* cluster and the *PersonOut* cluster learned by the combined model[1], and their corresponding values in the centriods of the two clusters. The two clusters share 3 of the top-5 features, some

[1]We made similar observations from centriods learned in GMM and K-Means, which are therefore not reported here.

Top-10 features	Values in the centriod of: *PersonIn*	*PersonOut*
type:⟨person⟩	0.9985	1
unigram after:,	0.7251	0.3404
unigram before:⟨s⟩	0.2705	0
bigram after:, ⟨digits⟩	0.2105	0.1879
bigram after:, who	0.1404	0.0567
unigram before:,	0.1067	0.0035
dobj:succeeds	0.0906	0
unigarm before:succeeds	0.0892	0
nsubj:resigned	0.0746	0.0284
unigram before:said	0.0673	0

Table 4: Top-10 features in the *PersonIn* cluster, as learned by GMM with the proposed prior.

Top-10 features	Values in the centriod of: *PersonOut*	*PersonIn*
type:⟨person⟩	1	0.9985
unigram before:mr.	0.9894	0
bigram before:⟨s⟩ mr.	0.5213	0
unigram after:,	0.3404	0.7251
bigram after:, ⟨digits⟩	0.1879	0.2105
unigram after:was	0.1667	0.0556
nsubj:president	0.1667	0.0117
nsubj:succeeds	0.1028	0.0102
bigram before:, mr.	0.0957	0
unigram after:'s	0.0745	0.0073

Table 5: Top-10 features in the *PersonOut* cluster, as learned by GMM with the proposed prior.

of them being general context features that might not help characterizing candidates from different slots (e.g. the unigram after the candidate is a comma). Both lists also contain features from dependency parse trees. Note that the "dobj:succeeds" feature in the *PersonIn* cluster is in fact contributed by *PersonOut* slot fillers, while the "nsubj:succeeds" feature in the *PersonOut* cluster is contributed by *PersonIn* slot fillers. Although listed among the top-10, these features have relatively low values in the learned centriods (about 0.1). These observations may suggest that the management succession data set lacks strong, discriminative features for all models to effectively distinguish between *PersonIn* and *PersonOut* candidates in an unsupervised manner.

To conclude, the proposed prior is effective in assigning different but confusing candidate slot fillers into appropriate slots, when there exist reasonable features that can be exploited in the label assignment process. This is evident by the improvements the proposed prior brings to GMM in the seminar announcement and aviation incident data sets.

8 Conclusions

We propose a generative model that incorporates distributional prior knowledge about template slots in a document for the unsupervised IE task. Specifically, we propose a Poisson-based prior that prefers label assignments to cover more distinct slots in the same document. The proposed prior also allows a slot to generate multiple fillers in a document, up to a certain number of times depending on the domain of interest.

We experimented with four existing generative models for the task of clustering slot fillers with a multinomial prior, which assumes that labels are generated independently in a document. We then evaluate the effectiveness of the proposed prior by incorporating it into the Gaussian mixture model (GMM), which is shown to be the best among the four existing models in our experiments. By incorporating the proposed prior into GMM, we can obtain significantly better clustering results on two out of three data sets.

Further improvements to this work are possible. Firstly, we assume that some adjustable parameters in the proposed prior can be manually fixed, such as the number of template slots in the output and the maximum numbers of fillers that can be generated by different slots. We are looking into methods for automatically learning such parameters. This will help improve the applicability of our work to different domains as an unsupervised model. Secondly, we currently consider in the prior a probability distribution over all possible label assignments for every document. This can be computationally expensive if input documents are long, or when we aim to discover large templates with large values of K. An alternative is to consider an approximate solution that evaluates, for instance, only the top few label assignments that are likely to maximize the likelihood of our observations. This remains as an interesting future work of this study.

Acknowledgments

This work is supported by DSO National Laboratories. We thank the anonymous reviewers for their helpful comments.

References

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *International Joint Conference on Artificial Intelligence*, pages 2670–2676.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Gregory Druck, Gideon Mann, and Andrew McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602.

Richard O. Duda, Peter E. Hart, and David G. Stork. 2001. *Pattern classification*. Wiley-Interscience, 2nd edition.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 207–214, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Dayne Freitag and Andrew Kachites McCallum. 1999. Information extraction with HMMs and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*.

João Graça, Kuzman Ganchev, and Ben Taskar. 2007. Expectation maximization and posterior constraints. In *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415, Morristown, NJ, USA. Association for Computational Linguistics.

J. B. Macqueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1*, pages 281–297.

Zvika Marx, Ido Dagan, and Eli Shamir. 2002. Cross-component clustering for template induction. In *Proceedings of the 2002 ICML Workshop on Text Learning*.

MUC-6. 1995. *Proceedings of the Sixth Message Understanding Conference*. Morgan Kaufmann, San Francisco, CA.

Benjamin Rosenfeld and Ronen Feldman. 2006. URES : An unsupervised Web relation extraction system. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 667–674.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL Main conference poster sessions*, pages 731–738.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 304–311.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wikipedia. 2009. List of accidents and incidents involving commercial aircraft. http://en.wikipedia.org/wiki/List_of_accidents_and_incidents_involving_commercial_aircraft.

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. 2009. Unsupervised relation extraction by mining Wikipedia texts using information from the web. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*.