Bidirectional Integration of Pipeline Models *

Xiaofeng Yu **Wai Lam**

Information Systems Laboratory
Department of Systems Engineering & Engineering Management
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{xfyu,wlam}@se.cuhk.edu.hk

Abstract

Traditional information extraction systems adopt pipeline strategies, which are highly ineffective and suffer from several problems such as error propagation. Typically, pipeline models fail to produce highly-accurate final output. On the other hand, there has been growing interest in integrated or joint models which explore mutual benefits and perform multiple subtasks simultaneously to avoid problems caused by pipeline models. However, building such systems usually increases computational complexity and requires considerable engineering. This paper presents a general, strongly-coupled, and bidirectional architecture based on discriminatively trained factor graphs for information extraction. First we introduce joint factors connecting variables of relevant subtasks to capture dependencies and interactions between them. We then propose a strong bidirectional MCMC sampling inference algorithm which allows information to flow in both directions to find the approximate MAP solution for all subtasks. Extensive experiments on entity identification and relation extraction using real-world data illustrate the promise of our approach.

Introduction

The goal of information extraction (IE) is to automatically extract structured information from free text or semi-structured sources. Most IE consists of compound, aggregate subtasks. Typically, two key subtasks are *segmentation* which identifies candidate records (e.g., word segmentation, chunking and entity recognition), and *relation* learning which discovers certain relations between different records (e.g., relation extraction and entity resolution). For such IE tasks, the availability of robust, flexible, and accurate systems is highly desirable.

Traditionally, the most common approach to IE is a pipeline in which stages are run independently in sequential order, and later stages have access to the output of already-completed earlier stages. While comparatively easy to assemble and computationally efficient, this pipeline approach is highly ineffective and suffers from inherent inferiority such as brittle accumulation of errors. It is therefore disappointing, but not surprising, that the overall performance is limited and upper-bounded (Yu 2007; Poon and Domingos 2007; Zhu et al. 2008).

In contrast, there has been increasing interest in using integrated or joint models across multiple subtasks as a paradigm for avoiding the cascading accumulation of errors in traditional pipelines. Setting up such models is usually very complex, and the computational cost of running them can be prohibitively intractable. While a number of previous researchers have taken steps toward this direction, they have various shortcomings: high computational complexity (Sutton, McCallum, and Rohanimanesh 2007); the number of uncertain hypotheses is severely limited (Wellner et al. 2004); subtasks are only loosely coupled (Zhu et al. 2007; Yu, Lam, and Chen 2009); or the approach is feed-forward or top-down integrated and it only allows information to flow in one direction (Finkel, Manning, and Ng 2006). Joint models can sometimes hurt accuracy, and fully joint approaches are still rare.

A significant amount of recent work has shown the power of conditionally-trained probabilistic graphical models for IE tasks (Sutton and McCallum 2006). Let $\mathcal{G}$ be a factor graph defining a probability distribution over a set of output variables $\mathbf{y}$ conditioned on observation sequences $\mathbf{x}$. $\{\Phi_i\}$ is a set of factors in $\mathcal{G}$, and each factor is defined as the exponential family of an inner product over sufficient statistics $\{f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\}$ and corresponding parameters λ_{ik} as $\Phi_i = \exp\{\sum_k \lambda_{ik} f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\}$. Let $Z(\mathbf{x})$ be the normalization factor, then the probability distribution (Lafferty, McCallum, and Pereira 2001) over $\mathcal{G}$ can be written as $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\Phi_i \in \mathcal{G}} \exp\{\sum_k \lambda_{ik} f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\}$. Practical models rely extensively on parameter tying to use the same parameters for several factors.

In this paper, we propose a highly-coupled, bidirectional integrated architecture based on discriminatively-trained factor graphs for IE tasks, which consists of two components – *segmentation* and *relation*. We introduce *joint factors* connecting variables of relevant subtasks capturing tight interactions between them. We then propose a strong *bidirectional* algorithm based on efficient Markov chain Monte Carlo

*The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No: CUHK4128/07) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050442 and 2050476). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

(MCMC) sampling to enable tractable inference, which allows information to flow bidirectionally and mutual benefits from different subtasks can be well exploited. We perform extensive experiments on entity identification and relation extraction from Wikipedia, and our model substantially outperforms previous state-of-the-art pipeline and joint models. Notably, our framework is considerably simpler to implement, and outperforms previous ones. It is also general and can be easily applied to a variety of probabilistic models and other real-world IE problems without considerable modifications.

Model

Let $\mathbf{X}$ be a document containing N observation sequences: $\mathbf{X} = \{\mathbf{X}_1, \cdots, \mathbf{X}_N\}$, each $\mathbf{X}_i$ consists of p tokens: $\mathbf{X}_i = \{x_{i1}, \cdots, x_{ip}\}$. Let $\mathbf{S}_i = \{s_{i1}, \cdots, s_{iq}\}$ be a segmentation assignment of observation sequence $\mathbf{X}_i$. Each segment s_{ij} is a triple $s_{ij} = \{\alpha_{ij}, \beta_{ij}, y_{ij}\}$, where α_{ij} is a start position, β_{ij} is an end position, and y_{ij} is the label assigned to all tokens of this segment. The segment s_{ij} satisfies $0 \leq \alpha_{ij} < \beta_{ij} \leq |\mathbf{X}_i|$ and $\alpha_{ij+1} = \beta_{ij} + 1$. Let e_m and e_n be two arbitrary entities in the document $\mathbf{X}$, and r_{mn} be the relation assignment between them. And $\mathbf{R}$ is the set of relation assignments of all entity pairs within document $\mathbf{X}$. For example, e_m and e_n can be entity candidates from segments or entire observation sequences. And r_{mn} can be a semantic relation (e.g., *member_of*) between entity candidates or the boolean coreference variable indicating whether or not two sequences (e.g., paper citations) are referring to each other.

To enable a bidirectional integration of two components – *segmentation* and *relation* in our framework, we introduce *joint factors* capturing interactions between variables in these components. The hypotheses from one component can be used for another to guide its decision making iteratively. The information flows between the two components form a closed loop. The two components are optimized in a collaborative manner such that both of their performance can be enhanced.

Segmentation

Due to its iterative manner, we use the superscript j to indicate the decision in the j-th iteration. Besides the conventional segmentation factor $\Phi(\mathbf{S}^j, \mathbf{X})$, the joint factor $\Phi^{\nabla}(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})$ involves both relation hypotheses in the j-th iteration and segmentation assignments from the last iteration. We assume that all potential functions factorize according to a set of feature functions and a corresponding set of real-valued weights. Suppose L, I and K are the number of observation sequences in document $\mathbf{X}$, the number of segments, and the number of feature functions. λ_k, μ_k and ν_k are corresponding weights for feature functions $g_k(\cdot)$, $r_k(\cdot)$ and $q_k(\cdot)$, respectively. The factor $\Phi(\mathbf{S}^j, \mathbf{X}) = \exp\{\sum_l^L \sum_i^I \sum_k^K \lambda_k g_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{X}_l)\}$. Similar to semi-CRFs (Sarawagi and Cohen 2004), the value of segment feature function $g_k(\cdot)$ depends on the current segment $s^j_{l,i}$, the previous segment $s^j_{l,i-1}$, and the whole observation $\mathbf{X}_l$. And transitions within a segment can be non-Markovian. The joint factor $\Phi^{\nabla}(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \exp\{\sum_l^L \sum_i^I \sum_k^K \mu_k r_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{R}^j, \mathbf{X}_l) + \sum_l^L \sum_i^I \sum_k^K \nu_k q_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{S}^{j-1}, \mathbf{X})\}$. The newly introduced feature function $r_k(\cdot)$ uses the decision of relation component in the j-th iteration $\mathbf{R}^j$ as its additional input. The function $q_k(\cdot)$ includes observation sequences in the entire document $\mathbf{X}$ and segmentation results $\mathbf{S}^{j-1}$ in the last iteration. Using $q_k(\cdot)$, the segmentation and labeling evidences from other occurrences all over the document can be exploited by referring the decision $\mathbf{S}^{j-1}$. Thus evidences for the same entity segments are shared among all their occurrences within the document. This can significantly alleviate the label consistency problem caused in previous probabilistic models. According to the celebrated Hammersley-Clifford theorem, the factor of the segmentation component in the j-th iteration is defined as a product of all potential functions over cliques in the graph:

$$\begin{aligned}
&\Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \Phi(\mathbf{S}^j, \mathbf{X}) \cdot \Phi^{\nabla}(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) \\
&= \exp\Bigg\{ \sum_l^L \sum_i^I \sum_k^K \lambda_k g_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{X}_l) + \sum_l^L \sum_i^I \sum_k^K \mu_k r_k \\
&(s^j_{l,i-1}, s^j_{l,i}, \mathbf{R}^j, \mathbf{X}_l) + \sum_l^L \sum_i^I \sum_k^K \nu_k q_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{S}^{j-1}, \mathbf{X}) \Bigg\} \quad (1)
\end{aligned}$$

Then the probability distribution of the segmentation component in the j-th iteration can be defined as

$$P(\mathbf{S}^j | \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \frac{1}{Z(\mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})} \prod \Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) \quad (2)$$

where $Z(\mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \sum_{\mathbf{S}^j} \prod \Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})$ is the normalization factor.

Relation

In the j-th iteration, the traditional relation factor $\Phi(\mathbf{R}^j, \mathbf{X})$ in this component is written as $\exp\{\sum_{m,n}^M \sum_k^K \theta_k f_k(e_m, e_n, r^j_{mn}, \mathbf{X}) + \sum_{m,t,n}^M \sum_k^K \xi_k w_k(r^j_{mt}, r^j_{nt}, r^j_{mn}, \mathbf{X})\}$ to model relations r^j_{mn} between all possible entity pairs $\{e_m, e_n\}$ in the document $\mathbf{X}$ and to enforce transitivity for relations, where M is the number of arbitrary entities in the document $\mathbf{X}$ and K is the number of feature functions. $1 \leq m, t, n \leq M$,$m \neq t, t \neq n$, and $m \neq n$. The joint factor $\Phi^{\nabla}(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1})$ is defined as $\exp\{\sum_{m,n}^M \sum_k^K \gamma_k h_k(e_m, e_n, r^j_{mn}, \mathbf{S}^{j-1}, \mathbf{X})\}$, taking the segmentation hypotheses in the $(j-1)$-th iteration $\mathbf{S}^{j-1}$ as its input. This joint factor captures tight dependencies between segmentations and relations. For example, if two segments are labeled as a *location* and a *person*, the semantic relation between them can be *birth_place* or *visited*, but cannot be *employment*. Such dependencies are crucial and modeling them often leads to improved performance. $f_k(\cdot)$, $w_k(\cdot)$ and $h_k(\cdot)$ are feature functions and θ_k, ξ_k and γ_k are their corresponding weights. Then the factor of the relation component in the j-th iteration can be written as follows:

$$\begin{aligned}
&\Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) = \Phi(\mathbf{R}^j, \mathbf{X}) \cdot \Phi^{\nabla}(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) \\
&= \exp\Bigg\{ \sum_{m,n}^M \sum_k^K \theta_k f_k(e_m, e_n, r^j_{mn}, \mathbf{X}) + \sum_{m,t,n}^M \sum_k^K \xi_k w_k \\
&(r^j_{mt}, r^j_{nt}, r^j_{mn}, \mathbf{X}) + \sum_{m,n}^M \sum_k^K \gamma_k h_k(e_m, e_n, r^j_{mn}, \mathbf{S}^{j-1}, \mathbf{X}) \Bigg\} \quad (3)
\end{aligned}$$

Similarly, we can get the conditional distribution of this component in the j-th iteration as follows:

$$P(\mathbf{R}^j | \mathbf{X}, \mathbf{S}^{j-1}) = \frac{1}{Z(\mathbf{X}, \mathbf{S}^{j-1})} \prod \Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) \quad (4)$$

where $Z(\mathbf{X}, \mathbf{S}^{j-1}) = \sum_{\mathbf{R}^j} \prod \Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1})$ is the normalization factor to make $P(\mathbf{R}^j | \mathbf{X}, \mathbf{S}^{j-1})$ a probability distribution.

Collaborative Parameter Estimation

Although both segmentation and relation components contain new variables, we show that they can be trained efficiently in a collaborative manner. Once we have trained a component, the decision of this component can guide the learning and decision making for another component. The two components run iteratively until converge. Such iterative optimization can boost both the performance of the two components.

Assume that the training instances are independent and identically distributed (IID). Under this assumption, we ignore the summation operator $\sum_{\mathbf{X}}$ in the log-likelihood during the following derivations. To reduce over-fitting, we use regularization and a common choice is a spherical Gaussian prior with mean 0 and covariance $\sigma^2 I$. Then the regularized log-likelihood function $\mathcal{L}$ for the segmentation component on the training document $\mathbf{X}$ is defined as

$$\mathcal{L} = \log\left[\Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})\right] - \log\left[Z(\mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})\right] - \sum_k \frac{\delta_k^2}{2\sigma^2} \quad (5)$$

To simplify the expression, let $c_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{R}^j, \mathbf{S}^{j-1}, \mathbf{X})$ be the general form of functions $g_k(\cdot)$, $r_k(\cdot)$ and $q_k(\cdot)$, and let δ_k be the general form of weights λ_k,μ_k and ν_k. Taking derivatives of this function over the parameter δ_k yields:

$$\frac{\partial \mathcal{L}}{\partial \delta_k} = \sum_l^L \sum_i^I c_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{R}^j, \mathbf{S}^{j-1}, \mathbf{X}) - \sum_l^L \sum_i^I c_k(s^j_{l,i-1}, s^j_{l,i}, \mathbf{R}^j, \mathbf{S}^{j-1}, \mathbf{X}) \times P(\mathbf{S}^j|\mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) - \sum_k^K \frac{\delta_k}{\sigma^2} \quad (6)$$

Let $b_k(e_m, e_n, r^j_{mt}, r^j_{nt}, r^j_{mn}, \mathbf{S}^{j-1}, \mathbf{X})$ be the general form of $f_k(\cdot)$, $w_k(\cdot)$ and $h_k(\cdot)$, and let η_k be the general form of parameters θ_k, ξ_k and γ_k. Similarly, for the relation component, the log-likelihood function $\mathcal{L}'$ is defined as

$$\mathcal{L}' = \log\left[\Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1})\right] - \log\left[Z(\mathbf{X}, \mathbf{S}^{j-1})\right] - \sum_k \frac{\eta_k^2}{2\sigma^2} \quad (7)$$

and the derivative of this function with respect to the parameter η_k is as follows

$$\frac{\partial \mathcal{L}'}{\partial \eta_k} = \sum_{m,t,n}^M b_k(e_m, e_n, r^j_{mt}, r^j_{nt}, r^j_{mn}, \mathbf{S}^{j-1}, \mathbf{X}) - \sum_{m,t,n}^M b_k(e_m, e_n, r^j_{mt}, r^j_{nt}, r^j_{mn}, \mathbf{S}^{j-1}, \mathbf{X}) \times P(\mathbf{R}^j|\mathbf{X}, \mathbf{S}^{j-1}) - \sum_k^K \frac{\eta_k}{\sigma^2} \quad (8)$$

Both of functions $\mathcal{L}$ and $\mathcal{L}'$ are concave, and can therefore be efficiently maximized by standard techniques such as stochastic gradient and L-BFGS algorithms. The segmentation component is optimized by using both the relation hypotheses from the relation component and the segmentation and labeling results from its last iteration as additional feature functions. The relation component benefits from the segmentation component by using the segmentation and labeling results explicitly in its feature functions. For initialization, we run segmentation component first without relation assignments. Since it is powerful enough to make a reasonable segmentation decision. The two components are optimized iteratively until convergence criteria is reached.

Bidirectional MCMC Sampling Inference

Ideally, the objective of inference is to find the most likely segmentation assignments $\mathbf{S}^*$ and corresponding most likely relation assignment $\mathbf{R}^*$, that is, to find $(\mathbf{S}, \mathbf{R})^* = \arg\max_{(\mathbf{S},\mathbf{R})} P(\mathbf{S}, \mathbf{R}|\mathbf{X})$ such that both of them are optimized simultaneously. Unfortunately, exact inference to this problem is generally intractable, since the search space is the Cartesian product of all possible segmentation and relation assignments. Consequently, approximate inference becomes an alternative. Instead of solving the joint optimization problem described above, we can solve two simpler inference problems $\mathbf{S}^* = \arg\max_{\mathbf{S}} P(\mathbf{S}|\mathbf{R}, \mathbf{X})$ and $\mathbf{R}^* = \arg\max_{\mathbf{R}} P(\mathbf{R}|\mathbf{S}, \mathbf{X})$ to optimize $\mathbf{S}$ and $\mathbf{R}$ iteratively.

We propose a *bidirectional* MCMC sampling algorithm to find the maximum a posteriori (MAP) assignments for both segmentations and relations. This algorithm is strongly coupled to inference based on efficient Metropolis-Hastings (MH) sampling (Hastings 1970) from both segmentations and relations to find an approximate solution of $(\mathbf{S}, \mathbf{R})^*$. This algorithm is a theoretically well-founded MCMC algorithm, and is guaranteed to converge. And it allows inference information to flow bidirectionally, such that mutual benefits from segmentations and relations can be well captured.

The MCMC methods are an efficient class of methods for approximate inference based on sampling. We can construct a Markov chain whose states are the variables we wish to sample. We can walk the Markov chain, occasionally outputting samples, and that these samples are guaranteed to be drawn from the target distribution. Let S^t be the current state of one segmentation sequence S and S^{t+1} be the next state of S. We assume that the current relation samples R^t have already been drawn. To draw segmentation samples from $P(S|R^t, \mathbf{X})$ in the model, we define the Markov chain as follows: from each state sequence we transfer to a state sequence obtained by changing the state at a particular segment S_i. If $|S_i| = 1$, we only change the label of this segment. If $1 < |S_i| \leq \mathbb{L}$ where $\mathbb{L}$ is the upper bound on segment length, we divide S_i into k sub-segments $S_{i1}S_{i2}\cdots S_{ik}$ with different labels. Thus the distribution over these possible transitions from state S^t to state S^{t+1} is defined as:

$$P(S^{t+1}|S^t, R^t, \mathbf{X}) = P(S^{t+1}_i|S^t_{-i}, R^t, \mathbf{X}) \quad (9)$$

where $S_i = (S_{i1}\cdots S_{ik})$, S_{-i} is all segments except S_i, and $S^t_{-i} = S^{t+1}_{-i}$. If $k = 1$, we assume $S_{i1} = S_i$.

We can walk the Markov chain to loop through segment S_i from $i = 1$ to $i = I$, and the attribute (boundary and label) of every segment can be changed dynamically. And for each one, we re-sample the state at segment S_i from distribution given in Equation 9. Let y_{ij} be the label of the sub-segment $S_{ij}(1 \leq j \leq k)$ and $\mathcal{Y}$ be the label set, after re-sampling all I segments, we can sample the whole segmentation sequences from the conditional distribution

$$P(S^{t+1}|S^t, R^t, \mathbf{X}) = \frac{P((S_{i1}\cdots S_{ik})^{t+1}, S^t_{-i}, R^t, \mathbf{X})}{\sum_{y_{ij}\in\mathcal{Y}} P((S_{i1}\cdots S_{ik})^{t+1}, S^t_{-i}, R^t, \mathbf{X})} \quad (10)$$

An MH step of the target distribution $P(S^{t+1}|R^t, \mathbf{X})$ and the proposal distribution $Q(\hat{S}|S^t, R^t, \mathbf{X})$ involves sampling a candidate sequence $\hat{S}$ given the current value S^t according to $Q(\hat{S}|S^t, R^t, \mathbf{X})$, and uses an acceptance/rejection scheme

to define a transition kernel with $P(S^{t+1}|S^t, R^t, \mathbf{X})$. The Markov chain then moves towards $\hat{S}$ (as the next state S^{t+1}) with acceptance probability $\mathcal{A}(S^t, \hat{S})$ and with probability $1 - \mathcal{A}(S^t, \hat{S})$ it is rejected and the next state remains at S^t. Moreover, to perform global optimization, a more principled strategy is to adopt simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983) in the MH algorithm, and the acceptance probability $\mathcal{A}(S^t, \hat{S})$ is written as

$$\mathcal{A}(S^t, \hat{S}) = \min\left\{1, \frac{P^{1/c_t}(\hat{S}|R^t, \mathbf{X})Q(S^t|\hat{S}, R^t, \mathbf{X})}{P^{1/c_t}(S^t|R^t, \mathbf{X})Q(\hat{S}|S^t, R^t, \mathbf{X})}\right\} \quad (11)$$

where c_t is a decreasing cooling schedule with $\lim_{t\to\infty} c_t = 0$. And this annealing technique has been shown to be very effective for optimization.

The proposal distribution $Q(\hat{S}|S^t, R^t, \mathbf{X})$ can be computed via Equation 10, and $Q(S^t|\hat{S}, R^t, \mathbf{X})$ can also be easily computed as

$$Q(S^t|\hat{S}, R^t, \mathbf{X}) = P(S_i^t|\hat{S}_{-i}, R^t, \mathbf{X}) = \frac{P(S_i^t, \hat{S}_{-i}, R^t, \mathbf{X})}{\sum_{y_i \in \mathcal{Y}} P(S_i^t, \hat{S}_{-i}, R^t, \mathbf{X})} \quad (12)$$

After we obtain the segmentation sample S^{t+1}, we can draw relation samples from $P(R|S^{t+1}, \mathbf{X})$. Similar MH procedure can also be exploited, and we omit the description due to space limitation. In summary, this bidirectional MH sampling algorithm will work as follows. Given initialized segmentation and relation assignments S^0 and R^0, and a user-defined sample size $\mathbb{N}$, it draws samples $\hat{S}$ from $P(S^{t+1}|R^t, \mathbf{X})$ $(0 \le t < \mathbb{N})$ while computing $\mathcal{A}(S^t, \hat{S})$ and setting $S^{t+1} = \hat{S}$ with probability $\mathcal{A}(S^t, \hat{S})$; otherwise setting $S^{t+1} = S^t$, and draws samples $\hat{R}$ from $P(R^{t+1}|S^{t+1}, \mathbf{X})$ via similar procedure. We run this algorithm to perform sampling for both segmentations and relations bidirectionally and iteratively for enough time, and it is guaranteed to converge to its stationary distribution. Thus it will find an approximate MAP solution for the most likely pair $(\mathbf{S}, \mathbf{R})^*$.

Note that the proposed algorithm is different from Finkel et al. (2005), who incorporated a limited number of constraints into probabilistic sequence models by Gibbs sampling, which is just a special case for the MH sampler; and Finkel et al. (2006), who modeled pipelines as Bayesian networks which are feed-forward and only allow information to flow into one direction. Exploring bidirectional information is appealing, especially during the inference procedure. And we will demonstrate and analyze its efficiency in the experiments.

Experiments

Data

We investigate the task of identifying entities and discovering semantic relationships between entity pairs from English encyclopedic articles in Wikipedia (http://www.wikipedia.org/). Our dataset consists of 1127 paragraphs from 441 pages from the online encyclopedia Wikipedia. We labeled 7740 entities into 8 categories, yielding 1243 *person*, 1085 *location*, 875 *organization*, 641 *date*, 1495 *year*, 38 *time*, 59 *number*, and 2304 *miscellaneous* names. This dataset also contains 4701 relation instances and 53 labeled relation types. The 10 most frequent relation types are *job_title*, *visited*, *birth_place*, *associate*, *birth_year*, *member_of*, *birth_day*, *opus*, *death_year*, and *death_day*. Note that this compound IE task involving entity identification and relation extraction is very challenging, and modeling tight interactions between entities and their relations is highly attractive (Yu and Lam 2008).

Methodology

We set the upper bound of the segment length $\mathbb{L}$ to 4 to enable efficient computation, since over 95% of the entities are within this threshold. Using L-BFGS algorithm, the training procedure is converged quickly within 3 loops between segmentation and relation components. And we set the sample size $\mathbb{N}$ to 10000 for the bidirectional MH inference algorithm. All experiments were performed on the Linux platform, with a 3.2GHz Pentium 4 CPU and 4 GB of memory.

Accurate entities enable features that are naturally expected to be useful to boost relation extraction. And a wide range of rich, overlapping features can be exploited in our model. These features include contextual features, part-of-speech (POS) tags, morphological features, entity-level dictionary features, clue word features. Feature conjunctions are also used. In leveraging relation extraction to improve entity identification, we use a combination of syntactic, entity, keyword, semantic, and Wikipedia characteristic features. More importantly, our model can incorporate multiple mention features $q_k(\cdot)$, which are used to collect evidences from other occurrences of the same entities for consistent segmentation and labeling. $r_k(\cdot)$ uses relation hypotheses and $h_k(\cdot)$ uses segmentation hypotheses as features. These features capture deep dependencies between entities and relations, and they are natural and useful to enhance the performance.

We perform four-fold cross-validation on this dataset, and take the average performance. For performance evaluation, we use the standard measures of Precision (P), Recall (R), and F-measure (the harmonic mean of P and R: $\frac{2PR}{P+R}$) for both entity identification and relation extraction. We also record the token-wise labeling Accuracy. We compare our approach with two pipeline models **CRF+CRF**, **CRF+MLN** and two joint models **Single MLN**, **FCRF**. **CRF+CRF** uses two linear-chain CRFs (Lafferty, McCallum, and Pereira 2001) to perform entity identification and relation extraction separately. **CRF+MLN** uses Markov logic network (MLN) (Richardson and Domingos 2006) for relation extraction, which is a highly expressive language for first-order logic and can conduct relational learning between entity pairs from CRF. **Single MLN** performs joint inference for both subtasks in a single MLN framework (Poon and Domingos 2007). **FCRF** (Sutton, McCallum, and Rohanimanesh 2007) is a factorial CRF used to jointly solve the two subtasks. All these models exploit standard parameter learning and inference algorithms in our experiments.

Performance of Entity Identification

Table 1 shows the performance of entity identification for different models. Our model substantially outperforms all baseline models on F-measure, and it is statistically significantly better (p-value < 0.05 with a 95% confidence inter-

val) according to McNemar's paired tests. The improvement demonstrates the merits of our approach by using joint factors to explore tight interactions between entities and relations such that both of them can be optimized in a collaborative manner to aid each other, resulting in improved performance. The pipeline models **CRF+CRF** and **CRF+MLN** perform entity identification and relation extraction independently without considering the mutual correlation between them, leading to reduced performance. By modeling interactions between two subtasks, boosted performance is achieved, as illustrated by **Single MLN** and **FCRF**.

Table 1: Comparative performance of different models for entity identification.

Method	Accuracy	Precision	Recall	F-measure
CRF+CRF	96.93	89.55	88.70	89.12
CRF+MLN	96.93	89.55	88.70	89.12
Single MLN	97.24	90.45	90.45	90.45
FCRF	97.29	90.98	90.37	90.67
Our model	97.55	94.03	93.89	**93.96**

Performance of Relation Extraction

Table 2 shows the performance of relation extraction, and our model achieves the best performance. All improvements of our model over baseline models are statistically significant. Both of the models **CRF+CRF** and **CRF+MLN** suffer from pipeline inherent inferiority such as brittle accumulation of errors. For example, they cannot correctly extract relations between mis-recognized entities. The **Single MLN** model captures some dependencies between entities and relations via first-order logic, however, limitations of first-order logic make it difficult to specify a relation factor that uses the uncertain output of segmentation (Singh, Schultz, and Mccallum 2009). Joint inference in **Single MLN** is only weakly coupled, and does not enforce transitivity, since the logic formulas only examine pairs of consecutive labels, not whole fields. As can be seen, our model achieves stronger interactions between two subtasks, and the proposed inference algorithm is strongly coupled and bidirectional, taking advantage of evidences from both subtasks. The **FCRF** model uses loopy belief propagation (LBP) for approximate learning and inference, which is inherently unstable and may cause convergence problems. In contrast, the L-BFGS algorithm we used is very efficient for parameter optimization, and the bidirectional MH sampling inference is theoretically well-founded, and it is guaranteed to converge.

Table 2: Comparative performance of different models for relation extraction.

Method	Accuracy	Precision	Recall	F-measure
CRF+CRF	93.72	70.40	57.85	63.51
CRF+MLN	93.81	69.39	59.53	64.08
Single MLN	93.96	68.54	61.75	64.97
FCRF	93.90	69.30	60.22	64.44
Our model	96.92	72.89	64.20	**68.27**

Efficiency

The efficiency of different models is summarized in Table 3. Compared to pipeline models, the learning time of our model is only increased linearly due to its bidirectional architecture. It is particularly notable that our model takes much less time than joint models. In particular, our model is over orders of magnitude (approximately 13 times) faster than **FCRF** for running. When the graph has large treewidth as in our case, the LBP algorithm in **FCRF** is inefficient, and is slow to converge.

Table 3: Efficiency comparison of different models on learning time (sec.) and inference time (sec.).

Method	Learning time	Inference time
CRF+CRF	2822	8
CRF+MLN	3479	118
Single MLN	8766	263
FCRF	105993	127
Our model	7157	859

Comparison with Other Methods

Table 4 compares our results with some recently published results on the same dataset. Notably, our approach outperforms previous ones given that we deal with a fairly more challenging problem involving both entity identification and relation extraction. All other listed systems assume that the golden-standard entities are already known and they only perform relation extraction (due to this reason, we only compare the performance on relation extraction.). However, such assumption is not valid in practice. And our model is more applicable to real-world IE tasks.

Table 4: Performance comparison with other systems.

System	Precision	Recall	F-measure
Culotta et al. (2006)	75.53	61.69	67.91
Nguyen et al. (2007)	29.07	53.86	37.76
Yu et al. (2009)	72.80	59.80	65.66
Our model	72.89	64.20	**68.27**

Bidirectionality

We also examine the nature and effectiveness of our proposed bidirectional MH inference algorithm and Figure 1 exhibits its feasibility over the greedy, N-best list, and unidirectional MH algorithms. It shows that the bi-directional MH algorithm consistently outperforms others on both tasks. For N-best list, we maintain and re-rank N-best list of segmentation assignments and corresponding relation assignments from our model, and take the most probable segmentation along with its relation assignment as the final output. We set $N = 20$ according to the holdout methodology. The greedy algorithm is the special case of the N-best list when $N = 1$. For the unidirectional MH algorithm, we draw segmentation samples from segmentation component and then we draw relation samples given the generated segmentation samples. And we also set the sample size to 10000. The greedy algorithm is very simple, but it only makes use of one-best list of segmentations and corresponding relations, losing much useful information. The N-best list gives useful improvements over the greedy. However, N-best list does not necessarily correspond to the best N list, and the N-best list is a very limited approximation for the full distribution of our model. The unidirectional MH algorithm outperforms N-best list when enough samples are drawn, since sampling

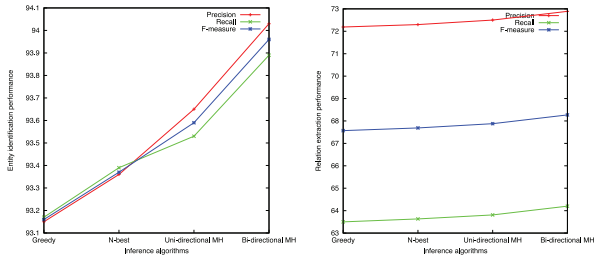

Figure 1: Performance comparison of different inference algorithms on entity identification (left) and relation extraction (right) tasks.

gives more diversity at each state and the full probability distribution can be better approximated. But this algorithm is only weakly coupled since it is feed-forward and information can only flow in one direction from segmentations to relations. This figure demonstrates the bidirectionality of our inference algorithm, which is highly coupled and mutual benefits from both subtasks can be well captured.

Related Work

Several closely related approaches have been proposed. Wellner et al. (2004) proposed an integrated model based on CRFs for citation matching, but the N-best list inference is a restrictive approximation for the full distribution. Zhu et al. (2007) and Yu et al. (2009) integrated two submodels together, but they are only loosely coupled in that they performed parameter estimation separately and inference information can only flow in one direction, similar as (Finkel, Manning, and Ng 2006). Hollingshead and Roark (2007) proposed pipeline iteration, using output from later stages of a pipeline to constrain earlier stages of the same pipeline, but it lacks the ability to model internal tight dependencies between stages. And these models often have complex structures and involve considerable engineering. For example, Zhu et al. (2008) used variational optimization approach for more learnable parameters in dynamic hierarchical modeling, and conducted extensive feature engineering. As can be seen, our approach outperforms previous integrated or joint models (Poon and Domingos 2007; Sutton, McCallum, and Rohanimanesh 2007), and is considerably easier to build and requires much less engineering. It is also general and can be easily applied to a wide range of probabilistic models and other real-world IE tasks.

Conclusion

We present a highly-coupled, bidirectional approach to integrating probabilistic pipeline models for information extraction. Joint factors are introduced to explore tight correlations between subtasks to aid each other, and parameter estimation can be performed collaboratively and efficiently to boost the performance. A strong bidirectional MH sampling algorithm is proposed to enable approximate inference, and this algorithm allows information to flow in both directions to capture mutual benefits. Experimental results exhibit that our model significantly outperforms recent state-of-the-art models while also running much faster than the joint models. The nature of the bidirectional inference algorithm is also analyzed and discussed. Directions for future work include further improving the scalability of the approach and applying it to other problems.

References

Culotta, A.; McCallum, A.; and Betz, J. 2006. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of HLT/NAACL-06*, 296–303.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of ACL-05*, 363–370.

Finkel, J. R.; Manning, C. D.; and Ng, A. Y. 2006. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of EMNLP-06*, 618–626.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109.

Hollingshead, K., and Roark, B. 2007. Pipeline iteration. In *Proceedings of ACL-07*, 952–959.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* 220:671–680.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*, 282–289.

Nguyen, D. P. T.; Matsuo, Y.; and Ishizuka, M. 2007. Relation extraction from Wikipedia using subtree mining. In *Proceedings of AAAI-07*, 1414–1420.

Poon, H., and Domingos, P. 2007. Joint inference in information extraction. In *Proceedings of AAAI-07*, 913–918.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Sarawagi, S., and Cohen, W. W. 2004. Semi-Markov conditional random fields for information extraction. In *Proceedings of NIPS-04*.

Singh, S.; Schultz, K.; and Mccallum, A. 2009. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Proceedings of ECML/PKDD-09*, 414–429.

Sutton, C., and McCallum, A. 2006. An introduction to conditional random fields for relational learning. In Getoor, L., and Taskar, B., eds., *Introduction to Statistical Relational Learning*. MIT Press.

Sutton, C.; McCallum, A.; and Rohanimanesh, K. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research* 8:693–723.

Wellner, B.; McCallum, A.; Peng, F.; and Hay, M. 2004. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of UAI-04*, 593–601.

Yu, X., and Lam, W. 2008. An integrated probabilistic and logic approach to encyclopedia relation extraction with multiple features. In *Proceedings of COLING-08*, 1065–1072.

Yu, X.; Lam, W.; and Chen, B. 2009. An integrated discriminative probabilistic approach to information extraction. In *Proceedings of CIKM-09*, 325–334.

Yu, X. 2007. Chinese named entity recognition with cascaded hybrid model. In *Proceedings of HLT/NAACL-07*, 197–200.

Zhu, J.; Zhang, B.; Nie, Z.; Wen, J.-R.; and Hon, H.-W. 2007. Webpage understanding: an integrated approach. In *Proceedings of KDD-07*, 903–912.

Zhu, J.; Nie, Z.; Zhang, B.; and Wen, J.-R. 2008. Dynamic hierarchical Markov random fields for integrated Web data extraction. *Journal of Machine Learning Research* 9:1583–1614.