

**Probabilistic Models for Information
Extraction: From Cascaded Approach to
Joint Approach**

YU, Xiaofeng

A Thesis
Submitted in Partial Fulfilment of the Requirements for the Degree of
Doctor of Philosophy
in
Systems Engineering and Engineering Management

The Chinese University of Hong Kong
June 2010

UMI Number: 3446023

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3446023

Copyright 2011 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Thesis/Assessment Committee

Professor Lam Kai Pui(Chair)
Professor Lam Wai (Thesis Supervisor)
Professor Cheng Hong (Committee Member)
Professor Wong Man Leung (External Examiner)

Abstract

Information Extraction (IE) aims at identifying specific pieces of information (data) in a unstructured or semi-structured textual document and transforming unstructured information in a corpus of documents or Web pages into a structured database. It can be applied to different types of text, such as newspaper articles, Web pages, medical notes, etc. There are several representative tasks in IE: named entity recognition (NER), which aims at identifying phrases that denote types of named entities, entity relation extraction, which aims at discovering the events or relations related to the entities, and the task of coreference resolution, aims at determining whether two extracted mentions of entities refer to the same object. IE is useful for a wide variety of applications.

Recently, probabilistic graphical models for sequence data have become the predominant formalism for IE, achieving state-of-the-art performance. We have investigated and developed a *cascaded* framework in an attempt to consider entity extraction and qualitative domain knowledge based on undirected, discriminatively-trained probabilistic graphical models. This framework consists of two stages and it is the combination of statistical learning and first-order logic. As a pipeline model, the first stage is a base model and the second stage is used to validate and correct the errors made in the base model. We incorporated domain knowledge that can be well formulated into first-order logic to extract entity candidates from the base model. We have applied this framework and achieved encouraging results in Chinese NER on the People's Daily corpus. We have participated in the Chinese NER shared task of the fourth SIGHAN Chinese language processing bakeoff (SIGHAN-6), which provides large-scale benchmark data for evaluation. Among all the groups participating the official evaluation, we obtained the best performance on the CityU corpus and the fourth place on the MSRA corpus. Moreover, we were the only group that obtained consistently over 90 F-measure on all the benchmark corpora in the NER open track.

The cascaded framework is ubiquitous in IE and it is in pipeline or decoupled architecture – attempting to perform compound IE tasks in several

separate, and independent stages. While comparatively easy to assemble and computationally efficient, this pipeline approach is highly ineffective and suffers 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. We present a general, strongly-coupled, and *bidirectional* architecture based on discriminatively trained factor graphs for information extraction, which consists of two components — segmentation and relation. First we introduce joint factors connecting variables of relevant subtasks to capture dependencies and interactions between them. We then propose a strong bidirectional Markov chain Monte Carlo (MCMC) sampling inference algorithm which allows information to flow in both directions to find the approximate maximum a posteriori (MAP) solution for all subtasks. 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 without considerable modifications.

The cascaded framework for Chinese NER is a simple integration of sequence labeling and logic models. Typically, probabilistic graphical models can deal well with uncertainty, but they are less expressive and flexible than logical or symbolic systems. Usually, they involve propositional, rather than first-order representations. First-order logic, on the other hand, is a powerful paradigm to represent a wide variety of knowledge. It is a more expressive formalism and allows the representation of variables and n -ary predicates, i.e., domain and relational knowledge. While highly expressive, this type of model lacks a sophisticated treatment of degrees of uncertainty and fuzziness, which permeates real-world domains, especially the ones usually associated with intelligence. Clearly, probabilistic graphical models and first-order logic offer complementary strengths and weaknesses for sequence data, and the integration of both is highly desirable.

Inspired by this motivation, we combine the advantages of both probabilistic graphical models for sequence data and first-order logic in a principled way, resulting in an *integrated* discriminative probabilistic framework which models both segmentations in sequence data and relations of different segments simultaneously for IE tasks. This integrated model offers a great flexibility to capture uncertainty for sequence modeling, as well as a variety of first-order domain knowledge. We illustrate the benefits of this model for mining implicit relations and new relation discovery, and capturing sub-structures in named entities. We propose the Metropolis-Hastings

(MH), a theoretically well-founded approximate MCMC algorithm to enable efficient and tractable inference for this model. This algorithm performs efficient sampling from segmentations via Markov chains, and it is guaranteed to converge. Joint parameter estimation in this model can be too expensive or even intractable. We perform parameter estimation somewhat separately for this integrated model.

The end-to-end performance of high-level IE systems for compound tasks is often hampered by the use of *cascaded* frameworks. The *integrated* model we proposed can alleviate some of these problems, but it is only loosely coupled. Parameter estimation is performed independently and it only allows information to flow in one direction. In this top-down integration model, the decision of the bottom sub-model could guide the decision of the upper sub-model, but not vice-versa. Thus, deep interactions and dependencies between different tasks can hardly be well captured.

Based on these observations and analysis, we propose a *joint* discriminative probabilistic framework to optimize all relevant subtasks simultaneously. This framework defines a joint probability distribution for both segmentations in sequence data and relations of segments in the form of an exponential family. This model allows tight interactions between segmentations and relations of segments and it offers a natural way for IE tasks. Since exact parameter estimation and inference are prohibitively intractable, a structured variational inference algorithm is developed to perform parameter estimation approximately. For inference, we propose a strong bi-directional MH approach to find the MAP assignments for joint segmentations and relations to explore mutual benefits on both directions, such that segmentations can aid relations, and vice-versa.

We perform extensive experiments on three important IE tasks using real-world datasets, namely Chinese NER, entity identification and relationship extraction from Wikipedia’s encyclopedic articles, and citation matching, to test our proposed models, including the *bidirectional* model, the *integrated* model, and the *joint* model. Experimental results show that our models significantly outperform current state-of-the-art probabilistic models, such as decoupled and joint models, illustrating the feasibility and promise of our proposed approaches. In addition, the effectiveness of the bi-directional MH algorithm over the greedy, N -best list, and uni-directional MH sampling algorithms is also discussed and compared. More importantly, these promising results will significantly further the applicability of our proposed approaches to other large-scale real world IE tasks.

摘要

信息抽取 (IE) 的目的是在非结构化或半结构化文本文件中识别具体信息 (数据) 以及转化语料库文档或网页中的非结构化信息到结构化数据库。它可应用于不同类型的文本, 如报纸文章, 网页, 医话等。IE 有几个代表性任务: 命名实体识别 (NER), 目的是确定表示命名实体的词组或短语; 实体关系抽取 (entity relation extraction), 其目的是发现与实体相关的事件或关系; 以及共指消解 (coreference resolution) 的任务, 目的是确定是否两个抽取出的实体暗指同一个对象。IE 对多种应用领域都有实用价值。

最近, 序列数据的概率图模型 (probabilistic graphical models) 成为解决 IE 问题的主要形式, 并达到了先进的性能。在无向, 区别性训练 (discriminatively-trained) 的概率图模型基础上, 我们研究开发了一个级联 (cascaded) 框架, 考虑实体抽取和定性领域知识 (domain knowledge)。这个框架包括两个阶段, 它是统计学习和一阶逻辑 (first-order logic) 的结合。作为一个管道 (pipeline) 模型, 第一阶段是基本模型, 第二阶段用来验证和纠正基本模型中的错误。我们纳入领域知识从基本模型中抽取候选实体, 这些领域知识可以很好地由一阶逻辑公式表示。我们应用这个框架, 在人民日报语料库的中文命名实体识别取得了令人鼓舞的结果。我们参加了第四届 SIGHAN 中文处理竞赛 (SIGHAN-6) 中的中文命名实体识别任务, 此竞赛提供大规模的基准数据作为评价。在所有参与官方评价的小组中, 我们在城大 (CityU) 语料库上取得了最佳性能, 在微软亚洲研究院 (MSRA) 语料库上取得第四名。此外, 我们是唯一一个在所有命名实体识别开放测试基准语料库中系统 F 值都超过 90 的小组。

级联框架在 IE 中普遍存在, 它是管道或非耦合架构—意图将复合的 IE 任务分为几个单独的, 独立的阶段。虽然比较容易组合及便于计算, 这种管道方法效率很低, 同时存在几个严重的问题, 如错误繁殖。通常情况下, 管道模型不能产生高度精确的最终输出。另一方面, 有日益增长的集成或联合模型的研究兴趣, 将多个子任务同时进行, 子任务间互惠互利, 互相帮助, 以避免由管道模型造成的问题。然而, 要建造这样的系统通常增加计算复杂度, 而且需要大量的工程。我们提出一个基于区别性训练因子图 (factor graph) 的普遍的, 强耦合双向 (bidirectional) 架构用于信息抽取。此架构有两个组成部分—分割 (segmentation) 和关系 (relation)。

首先我们引入连接相关子任务变量的联合因子 (joint factor) 捕获了任务之间的依赖关系和它们之间的相互作用。然后, 我们提出了一个高度双向的马尔可夫链蒙特卡罗 (MCMC) 采样推理算法以使信息流在两个方向, 为所有子任务找到近似最大后验 (MAP) 解。值得注意的是, 我们的框架相当简单, 便于实现, 效果优于以前的模型。此框架适用性广, 不需要大的修改就可以很容易地应用到各种概率模型。

用于中文命名实体识别的级联框架是序列标注和逻辑模型的简单集成。一般来说, 概率图模型可以处理好不确定性, 但他们不如逻辑或符号系统的表达性与灵活性。通常情况下, 它们是建议或命题式的, 而不是一阶逻辑表述。而另一方面, 一阶逻辑是一个强大的范式表达各种各样的知识。它是一个更有表现力的形式, 可表示变量和 n -元谓词, 即领域和关系知识。虽然有很好的表达性, 这种模型缺乏先进的不确定性和模糊性度量。不确定性和模糊性广泛存在于现实世界, 特别是与智慧有关的领域。很明显, 概率图模型和一阶逻辑对于序列数据优势互补, 故集成两者是非常可取的。

在此动机的激励下, 我们有原则地结合概率图模型和一阶逻辑对于序列数据的优势, 形成了一个集成的 (integrated) 概率框架模型。此模型同时考虑信息抽取中序列数据的分割与分割片段间的关系。此集成模型对序列数据建模能很灵活地处理不确定性, 以及多种一阶逻辑领域知识。我们阐明此模型在挖掘隐含和发现新的关系, 以及捕获命名实体子结构方面的优势。我们提出Metropolis-Hastings (MH), 一个有理论基础的近似MCMC算法, 使此模型可进行高效推理。此算法通过马尔可夫链对序列数据的分割进行有效的采样, 其收敛性得到保证。此模型中的联合参数估计过于复杂, 甚至难以解决。我们对此模型中的子结构进行独立的参数估计。

高级IE系统对于复杂任务端到端的性能往往受到级联框架的阻碍。我们上面提出的集成模型能减轻其中一些问题, 但它只是松耦合的。此模型中的参数估计独立执行, 而且只允许信息流在一个方向流动。在这种“自上而下”的集成模型中, 底部子模型的判断可以指导顶部子模型的判断, 但反之不然。因此, 不同任务间深层次的相互作用和依赖关系难以很好地捕捉。

基于这些观察与分析, 我们提出了一个联合 (joint) 概率框架同时优化所有相关的子任务。此框架定义了一个序列数据分割和分割片段间关系的指数形式的联合概率分布。这个模型考虑分割之间紧密的相互作用与分割片段间的关系, 为IE任务提供了一个理想的方法。由于精确参数估计与推理也极为棘手, 我们提出一种结构化变分推理 (structured variational inference) 算法进行近似参数估计。对于推理, 我们提出了一个高度双向的MH算法查找分割和关系的联合MAP解以求双方的互利互惠, 这样分割可以帮助关系, 反之亦然。

我们对三个代表性的IE任务: 中文命名实体识别, Wikipedia百科全书文章中的实体识别和关系抽取, 以及引文匹配在现实世界数据集上进行了

大量广泛的实验，以测试我们提出的模型，包括双向模型，集成模型及联合模型。实验结果表明，我们的模型明显优于目前最先进的概率模型，如非耦合和其它联合模型。这说明我们提出的模型是可行的，有希望的。此外，双向MH算法优于贪婪法， N -最好列表法（ N -best list），单向MH算法的性能也经过讨论和比较。更为重要的是，这些可喜的成果将使我们提出的方法进一步应用到其它大规模现实世界IE任务中。

Acknowledgements

I would like to thank all people who have helped and inspired me during my doctoral study. First of all, I am profoundly grateful to my advisor, professor Wai Lam, for his guidance during my research and study at the Chinese University of Hong Kong. His tireless pursuit of excellence in research, teaching, advising, and every other aspect of his academic work is truly inspirational. I am indebted to Wai Lam for priceless and copious advice about selecting interesting problems, making progress on difficult ones, pushing ideas to their full development, writing and presenting results in an engaging manner.

I would like to thank my thesis committee members, for their excellent suggestions and thought-provoking questions. I have learned a great deal from their work and their influence on this thesis is immense. I further express my gratitude to professors in the Department of Systems Engineering and Engineering Management for enriching my knowledge.

I would like to express my appreciation and gratitude to all my collaborators for their excellent ideas, hard work and dedication. I would like to thank my friends for their great spiritual support, encourage and advices. My parents have given me love, warmth, unbending support and constant encouragement in the progress of writing this thesis. Many thanks to all of them!

Contents

1	Introduction	1
1.1	Information Extraction	1
1.2	Problem Statement for Joint IE	2
1.3	Graphical Models for IE	3
1.3.1	Directed Graphical Models	4
1.3.2	Undirected Graphical Models	5
1.4	Conditional Random Fields	6
1.4.1	Linear-chain CRFs	7
1.4.2	Semi-CRFs	9
1.5	Markov Logic Networks	11
1.5.1	Parameter Estimation	13
1.5.2	Inference	14
1.6	Contributions	15
1.7	Thesis Outline	17
1.8	Publications Generated	19
2	Related Work	21
2.1	Pipeline Models for IE	21
2.2	Incorporating Probability with Logic	22
2.3	Integrated and Joint Models for IE	24

3	A Preliminary Study	26
3.1	A Cascaded Approach	26
3.2	Framework Overview	27
3.2.1	Applying to Chinese NER	28
3.2.2	CRFs as Base Model	29
3.2.3	Error Analysis	30
3.2.4	MLNs as Error Correction Model	31
3.2.5	Domain Knowledge	32
3.2.6	First-order Logic Construction	33
3.2.7	Implementation and Model Development	35
3.3	Experiments on People's Daily Corpus	35
3.3.1	Data	35
3.3.2	The Baseline NER System	35
3.3.3	Experimental Results	36
3.3.4	Significance Test	37
3.4	Official Results in SIGHAN-6	37
3.4.1	Data and Preprocessing	37
3.4.2	Features and Model Development	38
3.4.3	Official Results	39
3.5	Conclusion and Discussion	40
4	Bidirectional Integration of Pipeline Models	43
4.1	A Brief Introduction	43
4.2	Model	44
4.2.1	Segmentation	45
4.2.2	Relation	46
4.2.3	Collaborative Parameter Estimation	47
4.2.4	Markov chain Monte Carlo	48

4.2.5	Bidirectional MCMC Sampling Inference	50
5	An Integrated Discriminative Probabilistic Approach	53
5.1	A Brief Introduction	53
5.2	Motivating Examples	55
5.2.1	Implicit Relation Extraction and New Relation Discovery	55
5.2.2	Modeling Sub-structures in Named Entities	56
5.3	Model	57
5.4	Inference and Training	61
5.4.1	Inference	62
5.4.2	Parameter Estimation	63
6	Joint Models Incorporating Logic	65
6.1	A Brief Introduction	65
6.2	A Joint Model	67
6.2.1	Preliminaries and Notations	67
6.2.2	Model Formulation	68
6.2.3	Exact Parameter Estimation	72
6.2.4	Approximate Parameter Estimation via Structured Variational Inference	74
6.2.5	Bidirectional MCMC Sampling for Inference	78
7	Experiments	84
7.1	Chinese NER	84
7.1.1	Data	84
7.1.2	Methodology	84
7.1.3	Experimental Results and Analysis	86
7.1.4	Bidirectionality	87
7.2	Entity Identification and Relation Extraction From Wikipedia	89
7.2.1	Wikipedia	89

7.2.2	Data	90
7.2.3	Feature Set and Domain Knowledge	91
7.2.4	Implicit Relation Extraction	93
7.2.5	Methodology	94
7.2.6	Performance of Entity Recognition	95
7.2.7	Performance of Relation Extraction	97
7.2.8	Analysis and Discussion	97
7.2.9	Comparison with Other Methods	98
7.2.10	Bidirectionality	99
7.3	Citation Matching	100
7.3.1	Task Description	100
7.3.2	Data and Methodology	101
7.3.3	Experimental Results and Analysis	102
7.3.4	Bidirectionality	103
8	Conclusions and Future Work	107
8.1	Conclusions	107
8.2	Future Work	108
	Bibliography	109

List of Figures

- 1.1 An example of entity identification (left figure) and relation extraction (right figure) from
- 1.2 (a) A directed acyclic graph (DAG) that defines a partial order on its vertices. (b) An exa
- 1.3 Illustration of undirected graphical models and factor graphs. (a) An undirected graph on
- 1.4 Illustration and comparison of (a) general CRFs, (b) linear-chain CRFs, and (c) semi-CRF
- 1.5 A ground Markov network defined by the formulas in Table 1.1 and the constants Peter(A

- 3.1 Framework overview 28
- 3.2 An example of non-local dependency. The career title “教授” indicates a PER “苏珊” 30

- 5.1 An example of implicit relation extraction. The real lines show explicit (general) relations
- 5.2 Graphical representation of the integrated discriminative probabilistic model consisting of
- 5.3 The transition probability of the Markov chain from state $S^{(t)}$ to state $S^{(t+1)}$ is the conditi

- 6.1 An instance of graphical representation of the joint discriminative probabilistic model for s

- 7.1 Performance comparison of different inference algorithms on Chinese NER. 88
- 7.2 A snapshot of the encyclopedic article about Abraham Lincoln in Wikipedia. 90
- 7.3 An example of entities and relations in the data. 91
- 7.4 Performance comparison of different inference algorithms on entity identification (left) and
- 7.5 An example of citation matching. The notations $[\cdot]_{\text{AUTHOR}}$, $[\cdot]_{\text{TITLE}}$, and $[\cdot]_{\text{VENUE}}$ denote tl
- 7.6 Performance comparison of different inference algorithms on segmentation (left) and entity

List of Tables

1.1	Example of a KB and generated features	12
3.1	Domain knowledge for Chinese NER	33
3.2	Examples of NE candidates and first-order formulas	34
3.5	McNemar's tests on labeling disagreements	37
3.6	Statistics of SIGHAN official NER training and testing corpora.	38
3.7	OOV Rate of NER testing corpora.	38
3.8	Top 5 systems in SIGHAN NER open track on CityU corpus .	40
3.9	Top 5 systems in SIGHAN NER open track on MSRA corpus	40
3.3	Chinese NER by CRF model	42
3.4	Chinese NER by graphical models with logic	42
7.1	Domain knowledge and corresponding first-order formulas for NER.	86
7.2	Comparative performance of our models, CRFs, Semi-CRFs, and MLN models for NER.	8
7.3	Statistics of relation types and corresponding frequencies. . . .	92
7.4	Some representative first-order formulas.	93
7.5	Examples of first-order logic for implicit relation extraction. .	94
7.6	Comparative performance of our models, the CRF+CRF, CRF+MLN, and Single MLN m	
7.7	Performance of the CRF+CRF, CRF+MLN, and Single MLN models for relation extracti	
7.8	Performance of the bidirectional model, the integrated, and joint models for relation extra	
7.9	Performance comparison with other systems on relation extraction.	105

7.10 Comparative performance of different models for segmentation in citation matching. 106

7.11 Comparative performance of different models for entity resolution in citation matching. 106

List of Symbols

Chapter 1

- $\mathcal{G} = (V, E)$ A graph formed by a collection of vertices $V = \{1, 2, \dots, m\}$, and a collection of edges $E \subset V \times V$
- v $v \in V$ takes outcomes from a set \mathcal{V}
- e An edge in the graph \mathcal{G}
- \mathbf{x} A set of input variables that are observed
- \mathbf{y} A set of output variables that we wish to predict
- $pa(v)$ The parent of each vertex v in \mathcal{G}
- $P(v|pa(v))$ The probability distribution (nonnegative function) over the variables $(v, pa(v))$ in a DAG
- C A clique C is a fully connected subset of the vertex set V , meaning that $(s, t) \in E$ for all $s, t \in C$
- $\psi_c(\mathbf{y}_c, \mathbf{x}_c)$ The potential function, and we assume that each local function has the form $\psi_c(\mathbf{y}_c, \mathbf{x}_c) = \exp\{\sum_k \theta_{ck} f_{ck}(\mathbf{y}_c, \mathbf{x}_c)\}$ for some real-valued parameter vector θ_c , and for some set of feature functions or sufficient statistics $\{f_{ck}\}$
- F A set of potential functions (factors), and $F = \{\psi_c\}$
- Z A normalization factor defined as $Z = \sum_{\mathbf{x}, \mathbf{y}} \prod_{c \in C} \psi_c(\mathbf{y}_c, \mathbf{x}_c)$
- $P(\mathbf{y}, \mathbf{x})$ The joint probability distribution for input \mathbf{x} and output \mathbf{y}
- $P(\mathbf{y}|\mathbf{x})$ The conditional probability distribution for input \mathbf{x} and output \mathbf{y}
- C_p We can partition the factors of \mathcal{G} into $C = \{C_1, C_2, \dots, C_P\}$, where each C_p is a clique template whose parameters are tied

Λ	The parameter vector, and $\Lambda = \{\lambda_k\} \in \mathfrak{R}^K$
$\mathcal{D} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^N$	Independent and identically distributed (IID) training data, where each $\mathbf{x}^i = \{x_1^i, x_2^i, \dots, x_T^i\}$ is a sequence of inputs, and each $\mathbf{y}^i = \{y_1^i, y_2^i, \dots, y_T^i\}$ is a sequence of the desired predictions
$\ell(\Lambda)$	The likelihood or conditional log likelihood of the data
$1/2\sigma^2$	The regularization parameter that determines the strength of the penalty
\mathbf{y}^*	The most likely labeling $\mathbf{y}^* = \arg \max_{\mathbf{y}} P(\mathbf{y} \mathbf{x})$
g^k	The feature function for semi-CRFs, and it depends on the current segment, the whole observation, and the label of previous segment, that is, $g^k(i, \mathbf{x}, s) = g^k(y_{i-1}, y_i, t_i, \mu_i, \mathbf{x})$
$\varphi_i(i, \mathbf{x}, s)$	The potential function for semi-CRFs, and $\varphi_i(i, \mathbf{x}, s) = \exp(\alpha_i \cdot g(i, \mathbf{x}, s))$. $g = \langle g^1, g^2, \dots, g^K \rangle$ is a vector of feature functions and $\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_K \rangle$ is the corresponding weight vector
(F_i, w_i)	The pair for Markov logic networks, where F_i is a formula in first-order logic and w_i is a real number
$M_{L,C}$	The ground Markov network defined by a Markov logic network L and a finite set of constants $C = \{c_1, c_2, \dots, c_{ C }\}$
$n_i(x)$	The number of true groundings of F_i in possible worlds x
$x_{\{i\}}$	The true value of the atoms appearing in F_i
$\log P_w^*(X = x)$	The pseudo-log-likelihood of x given weights w
$MB_x(X_l)$	The state of X_l 's Markov blanket
$n_i(x_{ X_l=0})$	The number of true groundings of the i -th formula when we force $X_l = 0$ and leave the remaining data unchanged, and similarly for $n_i(x_{ X_l=1})$

Chapter 4

$\{\Phi_i\}$	A set of factors in \mathcal{G}
$\{f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\}$	A set of sufficient statistics or feature functions
λ_{ik}	Corresponding parameters with respect to $\{f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\}$
$Z(\mathbf{x})$	The normalization factor or normalization constant

\mathbf{X}	A document containing N observation sequences: $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$
\mathbf{X}_i	An observation sequence consisting of p tokens: $\mathbf{X}_i = \{x_{i1}, \dots, x_{ip}\}$
\mathbf{S}_i	A segmentation assignment of observation sequence \mathbf{X}_i , and $\mathbf{S}_i = \{s_{i1}, \dots, s_{iq}\}$
s_{ij}	A segment, and it 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 $\alpha_{ij+1} = \beta_{ij} + 1$.
e_m, e_n	Two arbitrary entities in the document \mathbf{X}
r_{mn}	A semantic relation between entity candidates or the boolean coreference variable indicating whether or not two sequences are referring to each other
\mathbf{R}	The set of relation assignments of all entity pairs within document \mathbf{X}
$\Phi(\mathbf{S}^j, \mathbf{X})$	The conventional segmentation factor, and $\Phi(\mathbf{S}^j, \mathbf{X}) = \exp\{\sum_l^L \sum_i^I \sum_k^K \lambda_k g_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{X}_l)\}$
$\Phi^\nabla(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})$	The joint segmentation factor, and $\Phi^\nabla(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \exp\{\sum_l^L \sum_i^I \sum_k^K \mu_k r_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{R}^j, \mathbf{X}_l) + \sum_l^L \sum_i^I \sum_k^K \nu_k q_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{S}^{j-1}, \mathbf{X})\}$
L, I and K	The number of observation sequences in document \mathbf{X} , the number of segments, and the number of feature functions
$g_k(\cdot), r_k(\cdot)$ and $q_k(\cdot)$	Feature functions for segmentation component
λ_k, μ_k and ν_k	Corresponding weights for feature functions $g_k(\cdot), r_k(\cdot)$ and $q_k(\cdot)$
$\Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})$	The overall segmentation factor, and $\Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \Phi(\mathbf{S}^j, \mathbf{X}) \cdot \Phi^\nabla(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})$
$\Phi(\mathbf{R}^j, \mathbf{X})$	The traditional relation factor, and $\Phi(\mathbf{R}^j, \mathbf{X}) = \exp\{\sum_{m,n}^M \sum_k^K \theta_k f_k(e_m, e_n, r_{mn}^j, \mathbf{X}) + \sum_{m,l,n}^M \sum_k^K \xi_k w_k(r_{ml}^j, r_{nt}^j, r_{mn}^j, \mathbf{X})\}$
M, K	M is the number of arbitrary entities in the document \mathbf{X} and K is the number of feature functions
$\Phi^\nabla(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1})$	The joint relation factor, and $\Phi^\nabla(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) = \exp\{\sum_{m,n}^M \sum_k^K \gamma_k h_k(e_m, e_n, r_{mn}^j, \mathbf{S}^{j-1}, \mathbf{X})\}$
$f_k(\cdot), w_k(\cdot)$ and $h_k(\cdot)$	Feature functions for relation component
θ_k, ξ_k and γ_k	Corresponding weights for feature functions $f_k(\cdot), w_k(\cdot)$ and $h_k(\cdot)$

$\Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1})$	The overall relation factor, and $\Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) = \Phi(\mathbf{R}^j, \mathbf{X}) \cdot \Phi^\nabla(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1})$
$c_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{R}^j, \mathbf{S}^{j-1}, \mathbf{X})$	The general form of functions $g_k(\cdot)$, $r_k(\cdot)$ and $q_k(\cdot)$
δ_k	The general form of weights λ_k, μ_k and ν_k
$b_k(e_m, e_n, r_{mt}^j, r_{nt}^j, r_{mn}^j, \mathbf{S}^{j-1}, \mathbf{X})$	The general form of $f_k(\cdot)$, $w_k(\cdot)$ and $h_k(\cdot)$
η_k	The general form of parameters θ_k , ξ_k and γ_k
$\mathcal{L}, \mathcal{L}'$	The regularized log-likelihood functions for segmentation component and relation component
\mathbf{S}^*	The most likely segmentation assignment, and $\mathbf{S}^* = \arg \max_{\mathbf{S}} P(\mathbf{S} \mathbf{R}, \mathbf{X})$
\mathbf{R}^*	The most likely relation assignment, and $\mathbf{R}^* = \arg \max_{\mathbf{R}} P(\mathbf{R} \mathbf{S}, \mathbf{X})$
$(\mathbf{S}, \mathbf{R})^*$	The pair of most likely segmentation and corresponding relation assignment, and $(\mathbf{S}, \mathbf{R})^* = \arg \max_{(\mathbf{S}, \mathbf{R})} P(\mathbf{S}, \mathbf{R} \mathbf{X})$
$P(S^{t+1} S^t, R^t, \mathbf{X})$	The distribution over possible transitions from state S^t to state S^{t+1}
$P(S^{t+1} R^t, \mathbf{X})$	The target distribution when sampling from the segmentation component
$Q(\hat{S} S^t, R^t, \mathbf{X})$	The proposal distribution for $P(S^{t+1} R^t, \mathbf{X})$
$A(S^t, \hat{S})$	The acceptance probability in the MH algorithm
c_t	A decreasing cooling schedule with $\lim_{t \rightarrow \infty} c_t = 0$ for simulated annealing
$P(R^{t+1} S^{t+1}, \mathbf{X})$	The target distribution when sampling from the relation component

Chapter 5

\mathbf{X}	An observation sequence of tokens
$ \mathbf{X} $	The length of the sequence (i.e., number of tokens)

S	A segmentation of the input sequence, and $S = \langle S_1, S_2, \dots, S_L \rangle$. Each entry is a segment which is a triple $S_i = \langle t_i, \mu_i, y_i \rangle$, with t_i as a start position, μ_i as an end position, and y_i as the label of this segment. $y_i \in \mathcal{Y}$ where \mathcal{Y} is the label set
R	A first-order logic possible world of segment relations expressed as a set of ground predicates R_i with truth value assigned, and $R = \langle R_1, R_2, \dots, R_M \rangle$
$\langle R, S \rangle$	The pair of segmentation and relation assignments in the integrated model. A valid assignment $\langle R, S \rangle$ must satisfy the condition that both of the two assignments are optimized, that is, the assignments of the segments and the assignments of the relations of segments are maximized simultaneously
\hat{R}, \hat{S}	The most likely relation assignment and segmentation assignment
$p(S \mathbf{X})$	Probability distribution of segmentations S conditioned on observation sequence \mathbf{X}
$p(R S, \mathbf{X})$	Probability distribution of relations R of segments given a segmentation S and observation sequence \mathbf{X}
C	A set of general constraints expressed as first-order logic formulas, and $C = \{C_1, C_2, \dots, C_N\}$. Each C_i contains some predicates representing constraints on elements in the domain
B	The ground predicates generated from the input sequence \mathbf{X} . B contains atomic formulas whose arguments are not variables
W_R	A set of segments, a set of functions, and a set of relations of segments; together with an interpretation, they determine the truth value of each ground atom
$n_i(W_R)$	The number of true groundings of a formula in the i -th first-order logic formula
$\phi_i(W_{R\{i\}})$	The potential function for the i -th logical formula, and $\phi_i(W_{R\{i\}}) = e^{\theta_i}$
$W_{R\{i\}}$	The truth value of the grounded predicate appearing in the formulas
λ, θ	Parameter vectors of the two sub-structures in the integrated model, $\lambda = \langle \lambda_1, \lambda_2, \dots, \lambda_K \rangle$ and $\theta = \langle \theta_1, \theta_2, \dots, \theta_L \rangle$

KB	A set of first-order logic formulas known as a knowledge base
Y	The pair of segmentations S and possible worlds in first-order logic of segment relations R for an observation sequence \mathbf{X} , and $Y = \{R, S\}$
Y^*	The pair of most likely relation assignment R^* and segmentation assignment S^* , and $Y^* = \{R^*, S^*\}$
$P(Y \mathbf{X})$	The joint conditional probability distribution for segmentations S and relations R given observation sequences \mathbf{X}
\mathcal{C}_s	A set of factors modeling segmentations for observation sequences, and $\mathcal{C}_s = \{\Phi_c(\mathbf{S}_c, \mathbf{X}_c)\}$. \mathbf{X}_c is a set of input variables and \mathbf{S}_c is a set of output variables, and they are arguments to the non-negative potential functions Φ_c
\mathcal{C}_r	A set of factors modeling relations between different segments in observation-sequences, $\mathcal{C}_r = \{\Psi_d(\mathbf{R}_d, \mathbf{S}_d, \mathbf{X}_d)\}$. \mathbf{X}_d and \mathbf{S}_d are sets of input variables, \mathbf{R}_d is a set of output variables. These are arguments to Ψ_d
$Z(\mathbf{X})$	The normalization factor for the joint model, and $Z(\mathbf{X}) = \sum_Y \prod_{c \in \mathcal{C}_s} \Phi_c(\mathbf{S}_c, \mathbf{X}_c) \prod_{d \in \mathcal{C}_r} \Psi_d(\mathbf{R}_d, \mathbf{S}_d, \mathbf{X}_d)$
$f_j(W_{\mathbf{R}_d})$	The number of true groundings of a formula in the j -th first-order logic formula
$W_{\mathbf{R}_d}$	Possible worlds of segment relations \mathbf{R}_d . And it is a set of segments, a set of functions, and a set of relations of segments; together with an interpretation. They determine the truth value assignment of each possible ground predicate of segment relations.
Θ	A set of real-valued weights for the joint model, and $\Theta = \{\lambda_1, \lambda_2, \dots, \lambda_K, \theta_1, \theta_2, \dots, \theta_L\} \in \mathfrak{R}^{K,L}$
\mathcal{D}	A set of training data $\mathcal{D} = \{(\mathbf{X}^l, Y_o^l)\}_{l=0}^N$, where \mathbf{X}^l is the l -th sample, and Y_o^l is the corresponding label and $Y_o^l = \{R_o^l, S_o^l\}$
$MB_{W_r}(\mathbf{X}_c)$	The state of the Markov blanket of \mathbf{X}_c in the training data
$P(Y Y_o, \mathbf{X})$	The desired joint conditional probability distribution
$Q(Y Y_o, \mathbf{X})$	The variational distribution and it serves as an approximation of $P(Y Y_o, \mathbf{X})$
$KL(Q P)$	The Kullback-Leibler (KL) divergence between the two distributions $Q(Y Y_o, \mathbf{X})$ and $P(Y Y_o, \mathbf{X})$

$\mathcal{L}(\Theta)$	The lower bound of the log-likelihood $\mathcal{L}(\Theta)$, and $\mathcal{L}(\Theta) \leq \mathcal{L}(\Theta)$
$\mathbb{H}(Q)$	$\mathbb{H}(Q) = -\sum_{S,R} Q(S,R S_o,R_o,\mathbf{X}) \log Q(S,R S_o,R_o,\mathbf{X})$ is the entropy of the variational distribution
$\mathbb{E}_Q\{\log P(S,R,S_o,R_o \mathbf{X})\}$	$\mathbb{E}_Q\{\log P(S,R,S_o,R_o \mathbf{X})\} = \sum_{S,R} Q(S,R S_o,R_o,\mathbf{X}) \log P(S,R,S_o,R_o \mathbf{X})$ is the expectation with respect to $Q(S,R S_o,R_o,\mathbf{X})$
\mathcal{F}	$\mathcal{F} = -\mathcal{L}(\Theta)$ is call the variational free energy and the lower bound $\mathcal{L}(\Theta)$ can be expressed as the difference of two free energies as $\mathcal{L}(\Theta) = \mathcal{F}_\infty - \mathcal{F}_0$, where $\mathcal{F}_\infty = -\log Z(\mathbf{X})$ is the free energy when we use model distribution with all variables free, and \mathcal{F}_0 is the free energy when we use data distribution with observable labels clamped to their values
s, r	s is an instantiation of the variable set S , and r is an instantiation of the variable set R , respectively
d_i	The projection of the instantiations s and r to the variables in $KL_i \subseteq \{R, S\}$, and the subsets $\{KL_{i=1}^I\}$ can be overlapped
V_1, \dots, V_M	Subsets (clusters) of variables $\{R, S\}$, and v_m is the projection of the instantiation $\{r, s\}$ to the variables in V_m
$Q(S), Q(R)$	We assume that $Q(S, R)$ can be factorized as $Q(S, R) = Q(S)Q(R)$, and we further assume $Q(S)$ to be of the form $Q(S) = \frac{1}{Z_{Q_S}} \prod_j \phi_j(u_j)$ and $Q(R)$ to be $Q(R) = \frac{1}{Z_{Q_R}} \prod_k \psi_k(w_k)$. Z_{Q_S} and Z_{Q_R} are two local normalization factors
U_1, \dots, U_J	Possibly overlapped subsets of the variable S , u_j is the projection of the instantiation s to the variables in U_j
W_1, \dots, W_K	Possibly overlapped subsets of the variable R , w_k is the projection of the instantiation r to the variables in W_k
$\tilde{Q}(S), \tilde{Q}(R)$	The un-normalized distributions. $\tilde{Q}(S) = \prod_j \phi_j(u_j)$ and $\tilde{Q}(R) = \prod_k \psi_k(w_k)$, $\tilde{Q}(S) \propto Q(S)$ and $\tilde{Q}(R) \propto Q(R)$
h_{mj}, e_{ij}, h'_{nk} and e'_{ik}	Indicators in the structured variational inference algorithm

Chapter 1

Introduction

1.1 Information Extraction

Information Extraction (IE) [19] is the process of filling the fields and records of a database from unstructured or loosely formatted text by automatically extracting sub-sequences of human readable text. It can be applied to different types of text, e.g., Web pages, corporate memos, news articles, research reports, e-mail, blogs, and historical documents. IE involves some major tasks: (1) finding the starting and ending boundaries of the text snippets that will fill a database field. For example, in the U.S. Department of Labor's continuing education extraction problem, the course title must be extracted, and segmentation must find the first and last words of the title, being careful not to include extra words ("Intro to Linguistics is taught") or to chop off too many words ("Intro to"). (2) determining which database field is the correct destination for each text segment. For example, "Introduction to Bookkeeping" belongs in the course title field, "Dr. Dallon Quass" in the course instructor field, and "This course covers..." in the course description field. (3) determining which fields belong together in the same record. For example, some courses may be described by multiple paragraphs of text, and other courses by just one; extraction must determine which field values from which paragraphs are referring to the same course. (4) putting information in a standard format in which it can be reliably compared. (5) collapsing redundant information so you don't get duplicate records in your database. For example, a course may be cross-listed in more than one department, and thus appear on more than one Web page; it will then be extracted multiple times, but we want only one record for it in our database.

IE is useful for a wide range of applications. It has made much progress in the past decade, and further research and industrial creativity continue

Abraham Lincoln_{PER} ((February 12)_{DATE}, [1809]_{YEAR} — [April 15]_{DATE}, [1865]_{YEAR}) was the 16th [President of the United States]_{MISC}, and the first president from the [Republican Party]_{ORG}. He was born to [Thomas Lincoln]_{PER} and [Nancy Hanks]_{PER}, two farmers in southeast [Hardin County]_{LOC}. When Lincoln was nine, his father remarried to [Sarah Bush Johnston]_{PER}. In [1841]_{YEAR}, Lincoln entered law practice with [William Herndon]_{PER}, a fellow member of the [Whig Party]_{ORG}. On [November 4]_{DATE}, [1842]_{YEAR}, Lincoln married [Mary Todd]_{PER}. Lincoln survived an assassination attempt in [Baltimore]_{LOC}. He successfully led the [American Civil War]_{MISC} to end slavery.

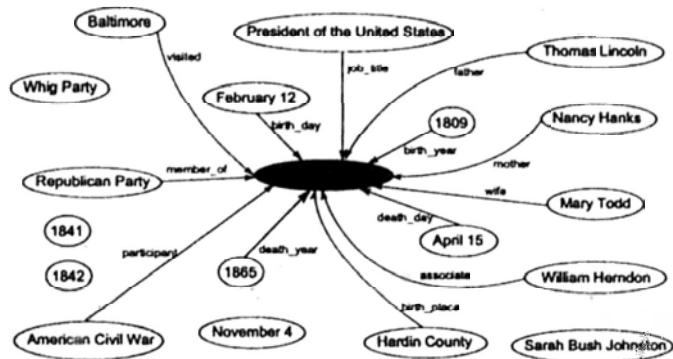


Figure 1.1: An example of entity identification (left figure) and relation extraction (right figure) from Wikipedia. In the left figure, the principal entity Abraham Lincoln is boxed and all secondary entities are bracketed as [·]. The notations [·]_{DATE}, [·]_{YEAR}, [·]_{PER}, [·]_{LOC}, [·]_{ORG}, and [·]_{MISC} denote that the entities are date, year, person, location, organization, and miscellaneous, respectively. In the right figure, the principal entity is in green (darker) and identified secondary entities are in yellow.

to push this progress. Extraction is being applied to increasingly complex problems and is being designed for more sophisticated yet easy use by non-technical end users.

1.2 Problem Statement for Joint IE

Most high-level IE consists of compound, aggregate subtasks. For example, relation extraction between entities consists of recognizing structured information about entities (e.g., person, location, and organization names) [94][102] and extracting the relationships between entities (e.g., visited, associate, and executive) [21][97][106]. Citation matching requires extracting bibliographic records from citation lists in technical papers (segmentation), and then identifying duplicate records (entity resolution) [88][67]. For such IE tasks, the availability of robust, flexible, and accurate systems is highly attractive.

The problem of joint information extraction is to solve all relevant subtasks in information extraction simultaneously, that is, all relevant IE subtasks are optimized at the same time and decisions of them are made together in a single coherent manner. This problem is usually very challenging, and often increases the model complexity.

Take the task of identifying entities and discovering semantic relation-

ships between entity pairs from English encyclopedic articles in Wikipedia¹ for example. The basic document is an *article*, which mainly defines and describes an entity (known as *principal entity*). This document mentions some other entities as *secondary entities* related to the principal entity. Clearly, our task consists of two subtasks — first, for entity identification, we need to recognize the secondary entities (both the boundaries and types of them) in the document². Second, after all the secondary entities are identified, our goal for relation extraction is to predict what relation, if any, each secondary entity has to the principal entity. We assume that there is no relationship between any two secondary entities in one document.

As an illustrative example, Figure 1.1 shows the task of entity identification and relationship extraction from encyclopedic documents in Wikipedia. Here, we use a part of the document about Abraham Lincoln. Our task consists of assigning a set of pre-defined entity types (e.g., YEAR, DATE, and PER) to segmentations in encyclopedic documents and assigning a set of pre-defined relations (e.g., *birth_day*, *birth_year*, and *job_title*) for each identified secondary entity to the principal entity. For example, *February 12* is identified as a DATE and *Republican Party* is identified as an ORG. And their relations to the principal entity *Abraham Lincoln* are *birth_day* and *member_of*, respectively. As shown in Figure 1.1, some secondary entities may not have any relation to the principal entity.

1.3 Graphical Models for IE

Graphical models bring together graph theory and probability theory in a powerful formalism for capturing complex dependencies among random variables, and building large-scale multivariate statistical models. In various applied fields including information extraction, statistical models have long been formulated in terms of graphs, and algorithms for computing basic statistical quantities such as likelihoods and score functions have often been expressed in terms of recursions operating on these graphs. Graphical models provide a natural tool for formulating variations on these classical architectures, as well as for exploring entirely new families of statistical models. Accordingly, in fields that involve the study of large numbers of interacting variables, graphical models are increasingly in evidence.

A graphical model consists of a collection of probability distributions that *factorize* according to the structure of an underlying graph. The main

¹<http://www.wikipedia.org/>

²Since the topic of an article usually defines a principal entity (e.g., a famous person) and it is easy to identify. In this thesis we only focus on secondary entity identification.

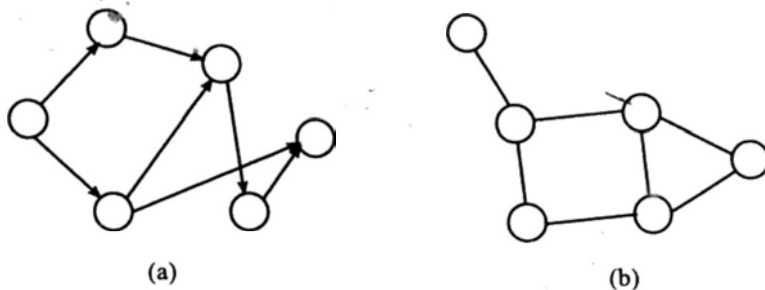


Figure 1.2: (a) A directed acyclic graph (DAG) that defines a partial order on its vertices. (b) An example of undirected graphical model with 6 vertices and 7 edges.

idea is to represent a distribution over a large number of random variables by a product of local functions that each depend on only a small number of variables. We consider probability distributions over sets of random variables $V = \mathbf{x} \cup \mathbf{y}$, where \mathbf{y} be a set of output variables that we wish to predict, and \mathbf{x} be a set of input variables that are observed. Every variable $v \in V$ takes outcomes from a set \mathcal{V} , which can be either continuous or discrete. A graph $\mathcal{G} = (V, E)$ is formed by a collection of vertices $V = \{1, 2, \dots, m\}$, and a collection of edges $E \subset V \times V$. Each edge consists of a pair of vertices $s, t \in E$, and may either be *undirected*, in which case there is no distinction between edge (s, t) and edge (t, s) , or *directed*, in which case we write $(s \rightarrow t)$ to indicate the direction.

1.3.1 Directed Graphical Models

A *directed* graphical model, also known as a Bayesian network, is based on a directed graph. See Figure 1.2(a) for an illustration of. In directed graphs, the edges signify asymmetric relations between the variables, loosely speaking the edges follow causal effects. Now suppose that \mathcal{G} is a directed acyclic graph (DAG), meaning that every edge is directed, and that the graph contains no directed cycles. Given a DAG, for each vertex v and its parents $pa(v)$, let $P(v|pa(v))$ denote a nonnegative function over the variables $(v, pa(v))$, normalized such that $\int P(v|pa(v))dv = 1$. In terms of these local functions, a directed graphical model consists of a collection of joint probability distributions (densities or mass functions) that factorize in the following way:

$$P(\mathbf{y}, \mathbf{x}) = \prod_{v \in V} P(v|pa(v)) \quad (1.1)$$

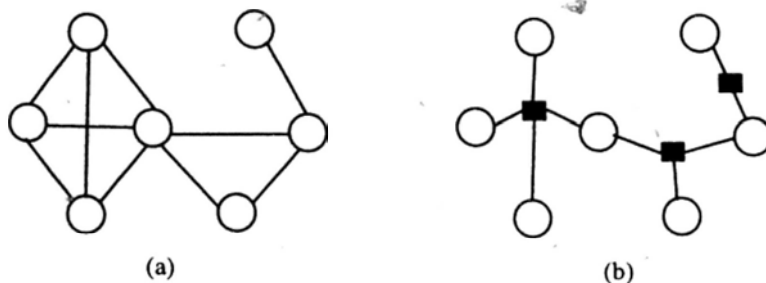


Figure 1.3: Illustration of undirected graphical models and factor graphs. (a) An undirected graph on 7 vertices. (b) Equivalent representation of the undirected graph in (a) as a factor graph, assuming that we define compatibility functions only on the maximal cliques in (a). The factor graph is a bipartite graph with vertex set and factor set, one for each of the compatibility functions of the original undirected graph.

We use the term *generative* model to refer to a directed graphical model in which the outputs topologically precede the inputs, that is, no x can be a parent of an output y . Essentially, a generative model is one that directly describes how the outputs probabilistically “generate” the inputs.

1.3.2 Undirected Graphical Models

In the undirected case, as shown in Figure 1.2 (b), the probability distribution factorizes according to functions defined on the cliques of the graph. A clique C is a fully connected subset of the vertex set V , meaning that $(s, t) \in E$ for all $s, t \in C$. With this notation, an *undirected* graphical model — also known as a Markov random field (MRF), or a Gibbs distribution — is a collection of distributions that factorize as

$$P(\mathbf{y}, \mathbf{x}) = \frac{1}{Z} \prod_{c \in C} \psi_c(\mathbf{y}_c, \mathbf{x}_c) \quad (1.2)$$

for any choice of factors $F = \{\psi_c\}$, where $\psi_c : \mathcal{V}^n \rightarrow \mathfrak{R}^+$. (These functions are also called *local functions* or *compatibility functions*.)

The constant Z is a normalization factor defined as $Z = \sum_{\mathbf{x}, \mathbf{y}} \prod_{c \in C} \psi_c(\mathbf{y}_c, \mathbf{x}_c)$, which ensures that the distribution sums to 1. The quantity Z , considered as a function of the set F of factors, is called the *partition function* in the statistical physics and graphical models communities. Computing Z is intractable in general, but much work exists on how to approximate it.

Graphically, we represent the factorization 1.2 by a *factor graph*. A factor graph is a bipartite graph $G = (V, F, E)$ in which a variable node $v_s \in V$ is connected to a factor node $\psi_c \in F$ if v_s is an argument to ψ_c . An example of a factor graph is shown graphically in Figure 1.3. In that figure, the circles are variable nodes, and the shaded boxes are factor nodes. We will assume that each local function has the form $\psi_c(\mathbf{y}_c, \mathbf{x}_c) = \exp\{\sum_k \theta_{ck} f_{ck}(\mathbf{y}_c, \mathbf{x}_c)\}$ for some real-valued parameter vector θ_c , and for some set of *feature functions* or *sufficient statistics* $\{f_{ck}\}$. This form ensures that the family of distributions over V parameterized by θ is an exponential family. Much of the discussion in this thesis actually applies to exponential families in general.

1.4 Conditional Random Fields

Directed graphical models are generative, and they assign a joint probability $P(\mathbf{y}, \mathbf{x})$ to paired observations; the parameters are typically trained to maximize the joint likelihood of training instances. To define a joint probability over observations, a generative model needs to enumerate all possible observation sequences, typically requiring a representation in which observations are task-appropriate atomic entities. In particular, it is not practical to represent multiple interacting features or long-range dependencies of the observations, since the inference problem for such models is intractable.

This difficulty is one of the main motivations for looking at conditional models as an alternative. A conditional model specifies the probabilities $P(\mathbf{y}|\mathbf{x})$ of possible label sequences \mathbf{y} given an observation sequence \mathbf{x} , and it is also called *discriminative*. A discriminative model does not expand modeling effort on the observations. The principal advantage of discriminative modeling is that it is better suited to rich, overlapping and agglomerative features. The probability of a transition between labels may depend not only on the current observation, but also on past and future observations, if available. In contrast, generative models must make very strict independence assumptions on the observations, for instance conditional independence given the labels, to achieve tractability.

Conditional random fields (CRFs) [47][82] are undirected graphical models trained to maximize the conditional probability of the desired outputs given the corresponding inputs. CRFs have the great flexibility to encode a wide variety of arbitrary, overlapping, and non-independent features and to straightforwardly combine rich domain knowledge. Furthermore, they are discriminatively trained, and are often more accurate than generative models, even with the same features. Let \mathcal{G} be a factor graph over \mathbf{y} and \mathbf{x} with factors $C = \{\Phi_c(\mathbf{y}_c, \mathbf{x}_c)\}$, where \mathbf{x}_c is the set of input variables that are ar-

guments to the local function Φ_c , and similarly for \mathbf{y}_c . CRFs are defined as follows:

Definition 1 (conditional random fields): Let $C = \{\Phi_c(\mathbf{y}_c, \mathbf{x}_c)\}$ be a set of factors over graph \mathcal{G} . Then the distribution P is a conditional random field if and only if

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Phi_c(\mathbf{y}_c, \mathbf{x}_c), \quad (1.3)$$

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in C} \Phi_c(\mathbf{y}_c, \mathbf{x}_c)$ is a normalization factor over all state sequences for the sequence \mathbf{x} .

We assume the potentials factorize according to a set of features $\{f_k(\mathbf{y}_c, \mathbf{x}_c)\}$ as

$$\Phi_c(\mathbf{y}_c, \mathbf{x}_c) = \exp \left(\sum_k \lambda_k f_k(\mathbf{y}_c, \mathbf{x}_c) \right), \quad (1.4)$$

so that the family of distributions $\{P\}$ is an exponential family. And we assume that the features are given and fixed. The model parameters are a set of real-valued weights $\Lambda = \{\lambda_k\}$, one weight for each feature.

Practical models rely extensively on *parameter tying* and we can partition the factors of \mathcal{G} into $C = \{C_1, C_2, \dots, C_P\}$, where each C_p is a clique template whose parameters are tied. Each clique template C_p is a set of factors which has a corresponding set of sufficient statistics $\{f_{pk}(\mathbf{x}_p, \mathbf{y}_p)\}$ and parameters $\theta_p \in \mathfrak{R}^K$. Then the CRF model can be written as

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{C_p \in C} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{y}_c, \mathbf{x}_c; \theta_p), \quad (1.5)$$

where each factor is parameterized as $\Psi_c(\mathbf{y}_c, \mathbf{x}_c; \theta_p) = \exp\{\sum_{k=1}^{K(p)} \lambda_{pk} f_{pk}(\mathbf{y}_c, \mathbf{x}_c)\}$ and the normalization function is $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{C_p \in C} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{y}_c, \mathbf{x}_c; \theta_p)$.

CRFs have been successfully applied to a number of real-world tasks, including NP chunking [75], Chinese word segmentation [62], information extraction [64, 63], named entity identification [54, 74], and many others.

1.4.1 Linear-chain CRFs

One of the most important CRFs is the linear-chain CRFs in which a first-order Markov assumption is made among labels. In this case, the cliques of the conditional model are the nodes and edges, so that there are feature functions $f_k(y_t, y_{t-1}, \mathbf{x})$ for each label transition. Feature functions can be arbitrary (Here we write the feature functions as potentially depending on the entire input sequence). Linear-chain CRFs have efficient exact training

and inference algorithms, as we will show below³. We first give the formal definition of linear-chain CRFs:

Definition 2 (linear-chain CRFs): Let $\Lambda = \{\lambda_k\} \in \mathfrak{R}^K$ be a parameter vector, and $f_k(y, y', \mathbf{x}_t)_{k=1}^K$ be a set of real-valued feature functions. Then a linear-chain conditional random field is a distribution $P(\mathbf{y}|\mathbf{x})$ that takes the form

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left(\sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right), \quad (1.6)$$

where $Z(\mathbf{x})$ is an instance-specific normalization function and $Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp \left(\sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right)$.

Parameter Estimation

We discuss how to estimate the parameters $\Lambda = \{\lambda_k\}$ of a linear-chain CRF. Given independent and identically distributed (IID) training data $\mathcal{D} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^N$, where each $\mathbf{x}^i = \{x_1^i, x_2^i, \dots, x_T^i\}$ is a sequence of inputs, and each $\mathbf{y}^i = \{y_1^i, y_2^i, \dots, y_T^i\}$ is a sequence of the desired predictions. Parameter estimation is typically performed by penalized maximum likelihood or conditional log likelihood of the data as:

$$\ell(\Lambda) = \sum_{i=1}^N P(\mathbf{y}^i|\mathbf{x}^i) \quad (1.7)$$

After substituting in the CRF model 1.6 into the likelihood 1.7, we get the following expression:

$$\ell(\Lambda) = \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(y_t^i, y_{t-1}^i, \mathbf{x}_t^i) - \sum_{i=1}^N \log Z(\mathbf{x}^i). \quad (1.8)$$

To avoid over-fitting, we use *regularization* and a common choice of penalty is based on the Euclidean norm of Λ and on a regularization parameter $1/2\sigma^2$ that determines the strength of the penalty. Then the regularized log likelihood is

$$\ell(\Lambda) = \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(y_t^i, y_{t-1}^i, \mathbf{x}_t^i) - \sum_{i=1}^N \log Z(\mathbf{x}^i) - \sum_{k=1}^K \frac{\lambda_k^2}{2\sigma^2} \quad (1.9)$$

³Parameter estimation for general CRFs is essentially the same as for linear-chains, except that computing the model expectations requires more general inference algorithms. Any inference algorithm for graphical models, such as the exact junction tree algorithm or various approximate inference can be exploited.

The parameter σ^2 is a free parameter which determines how much to penalize large weights. Determining the best regularization parameter can require a computationally-intensive parameter sweep. Fortunately, often the accuracy of the final model does not appear to be sensitive to changes in σ^2 . The partial derivatives of 1.9 are

$$\frac{\partial \ell}{\partial \lambda_k} = \sum_{i=1}^N \sum_{t=1}^T f_k(y_t^i, y_{t-1}^i, \mathbf{x}_t^i) - \sum_{i=1}^N \sum_{t=1}^T \sum_{y, y'} f_k(y, y', \mathbf{x}_t^i) P(y, y' | \mathbf{x}^i) - \sum_{k=1}^K \frac{\lambda_k}{\sigma^2}. \quad (1.10)$$

The function $\ell(\Lambda)$ is concave, and adding regularization ensures that ℓ is strictly concave, thus it can be efficiently maximized by second-order techniques such as conjugate gradient and L-BFGS algorithms. Both the partition function $Z(\mathbf{x})$ in the likelihood and the marginal distributions $P(y_t, y_{t-1} | \mathbf{x})$ in the gradient can be computed by forward-backward, which uses computational complexity $\mathcal{O}(TM^2)$ for each training instance and a total training cost of $\mathcal{O}(TM^2NG)$, where N is the number of training examples, and G the number of gradient computations required by the optimization procedure.

Inference

There are two common inference problems for CRFs. First, during training, computing the gradient requires marginal distributions for each edge $P(y_t, y_{t-1} | \mathbf{x})$, and computing the likelihood requires $Z(\mathbf{x})$. Second, to label an unseen instance, we compute the most likely labeling $\mathbf{y}^* = \arg \max_{\mathbf{y}} P(\mathbf{y} | \mathbf{x})$. In linear-chain CRFs, both inference tasks can be performed efficiently and exactly by variants of the standard dynamic-programming algorithms for HMMs. These standard inference algorithms are described in more detail by [68]. A final inference task that is useful in some applications is to compute a marginal probability $P(y_t, y_{t+1}, \dots, y_{t+k} | \mathbf{x})$ over a range of nodes. For example, this is useful for measuring the model's confidence in its predicted labeling over a segment of input. This marginal probability can be computed efficiently using constrained forward-backward, as described by [20]. We omit these inference algorithms in this thesis.

1.4.2 Semi-CRFs

The semi-Markov conditional random fields (semi-CRFs) [18, 72] are an extension of the linear-chain CRFs [47] for sequence data segmentation and labeling. In this model, \mathbf{x} is a token sequence and $|\mathbf{x}|$ is the length of the

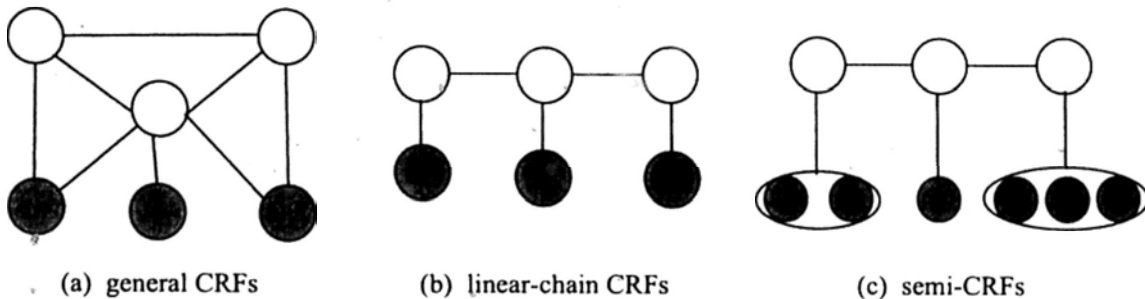


Figure 1.4: Illustration and comparison of (a) general CRFs, (b) linear-chain CRFs, and (c) semi-CRFs. The gray (darker) nodes represent input variables (e.g., sequence tokens) and the blank nodes represent output variables (e.g., labels). In (c), each ellipse represents a segment consisting of several consecutive sequence tokens.

sequence (i.e., number of tokens). The vector $s = \langle s_1, s_2, \dots, s_n \rangle$ is a segmentation of \mathbf{x} , and each entry is a segment which is a triple $s_i = \langle t_i, \mu_i, y_i \rangle$, with t_i as a start position, μ_i as an end position, and y_i as the label of this segment. Thus, a segment s_i means that the label y_i is assigned to all the observations between the start position t_i and the end position μ_i in the observation sequence \mathbf{x} . It is reasonable to assume that segments have positive lengths and adjacent segments touch, that is, $0 \leq t_i \leq \mu_i \leq |\mathbf{x}|$ and $t_{i+1} = \mu_i + 1$. Let g^k be a feature function, and it depends on the current segment, the whole observation, and the label of previous segment, that is, $g^k(i, \mathbf{x}, s) = g^k(y_{i-1}, y_i, t_i, \mu_i, \mathbf{x})$. Let $g = \langle g^1, g^2, \dots, g^K \rangle$ be a vector of feature functions and $\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_K \rangle$ be the corresponding weight vector. As a CRF model, the probability distribution $P(s|\mathbf{x})$ is also of the form but with the traditional label assignment y replaced by a segmentation s and the cliques are replaced by segments:

$$P(s|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{i=1:|s|} \varphi_i(i, \mathbf{x}, s), \quad (1.11)$$

where $\varphi_i(i, \mathbf{x}, s) = \exp(\alpha_i \cdot g(i, \mathbf{x}, s))$, and $Z(\mathbf{x}) = \sum_s \prod_{i=1}^{|s|} \varphi_i(i, \mathbf{x}, s)$.

The semi-CRF model is capable of measuring properties of segments, and transitions within a segment can be non-Markovian. Parameter estimation and finding the maximum a posterior segmentation can be efficiently carried out via a dynamic programming algorithm. The computational complexity is a constant factor more than that of the traditional linear-chain model when

the maximum length of the segments is assumed to be fixed.

Besides linear-chain and semi-CRFs, several special cases of conditional random fields are of particular interest. For example, relational Markov networks (RMNs) [85] are a type of general CRF in which the graphical structure and parameter tying are determined by an SQL-like syntax. Dynamic conditional random fields (DCRFs) [84, 83] are sequence models which allow multiple labels at each time step, rather than single labels as in linear-chain CRFs. 2D CRFs [107] are two-dimensional conditional random fields incorporating the two-dimensional neighborhood dependencies in Web pages. And the graphical representation of the 2D CRF model is a 2D grid. Hierarchical CRFs [108][48] are a class of CRFs with hierarchical structure and they are suitable for hierarchical classification problems such as Web data mining and Web page understanding. When observation data have distinct sub-structure, models that exploit hidden state are advantageous. A related model is presented in [31, 51], who build a hidden-state CRF (HCRF) which can estimate a class label given a segmented sequence in a phone classification task. A similar model for natural language parsing is shown in [44]. More recently, a hidden dynamic conditional random field (HDCRF) [96] model which can capture both internal and external class dynamics to label sequence data is presented. [96] introduces a small number of hidden state variables to model the sub-structure of a observation sequence and learn dynamics between different class labels.

1.5 Markov Logic Networks

A Markov network (also known as Markov random field) is a model for the joint distribution of a set of variables [61]. It is composed of an undirected graph $G = (V, E)$ and a set of real-valued potential functions ϕ_k . A first-order knowledge base (KB) [28] is a set of sentences or formulas in first-order logic.

A Markov logic network (MLN) [69] is a KB with a weight attached to each formula (or clause). Together with a set of constants representing objects in the domain, it specifies a ground Markov network containing one feature for each possible grounding of a first-order formula F_i in the KB, with the corresponding weight w_i . The basic idea in MLNs is that: when a world violates one formula in the KB it is less probable, but not impossible. The fewer formulas a world violates, the more probable it is. The weights associated with the formulas in an MLN jointly determine the probabilities of those formulas (and vice versa) via a *log-linear model*.

Definition 3 (Markov logic networks): A Markov logic network L is a

Table 1.1: Example of a KB and generated features

First-order Logic (KB)	Generated Features
$\forall x, y \text{ Employ}(x, y) \Rightarrow$ $\text{Person}(x), \text{Company}(y)$	$\text{Employ}(\text{Peter}, \text{IBM}) \Rightarrow \text{Person}(\text{Peter}), \text{Company}(\text{IBM})$ $\text{Employ}(\text{Smith}, \text{IBM}) \Rightarrow \text{Person}(\text{Smith}), \text{Company}(\text{IBM})$
$\forall x, y, z \text{ Colleague}(x, y) \Rightarrow$ $\text{Employ}(x, z) \wedge \text{Employ}(y, z)$	$\text{Colleague}(\text{Peter}, \text{Smith}) \Rightarrow \text{Employ}(\text{Peter}, \text{IBM})$ $\wedge \text{Employ}(\text{Smith}, \text{IBM})$

set of pairs (F_i, w_i) , where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = \{c_1, c_2, \dots, c_{|C|}\}$, it defines a Markov network $M_{L,C}$ (Equation 1.12) as follows:

1. $M_{L,C}$ contains one binary node for each possible grounding of each predicate appearing in L . The value of the node is 1 if the ground atom is true, and 0 otherwise.
2. $M_{L,C}$ contains one feature for each possible grounding of each formula F_i in L . The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_i associated with F_i in L .

An MLN is a statistical relational model that defines a probability distribution over Herbrand interpretations (possible worlds), and can be thought of as a *template* for constructing Markov networks. Given different sets of constants, it will produce different networks. These networks will have certain regularities in structure and parameter given by the MLN and they are called ground Markov networks. Suppose $\text{Peter}(A)$, $\text{Smith}(B)$ and $\text{IBM}(X)$ are 3 constants, a KB and generated features are listed in Table 1.1. The formula $\text{Employ}(x, y) \Rightarrow \text{Person}(x), \text{Company}(y)$ means x is employed by y and $\text{Colleague}(x, y) \Rightarrow \text{Employ}(x, z) \wedge \text{Employ}(y, z)$ means x and y are colleagues if they are employed by the same company. Figure 1.5 shows the graph of the ground Markov network defined by the formulas in Table 1.1 and the 3 constants $\text{Peter}(A)$, $\text{Smith}(B)$ and $\text{IBM}(X)$. The probability distribution over possible worlds x specified by the ground Markov network $M_{L,C}$ is given by

$$P(X = x) = \frac{1}{Z} \exp\left(\sum w_i n_i(x)\right) = \frac{1}{Z} \prod \phi_i(x_{\{i\}})^{n_i(x)} \quad (1.12)$$

where $n_i(x)$ is the number of true groundings of F_i in x , $x_{\{i\}}$ is the true value of the atoms appearing in F_i , and $\phi_i(x_{\{i\}}) = e^{w_i}$.

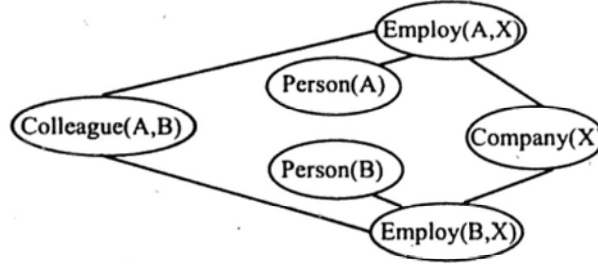


Figure 1.5: A ground Markov network defined by the formulas in Table 1.1 and the constants $\text{Peter}(A)$, $\text{Smith}(B)$ and $\text{IBM}(X)$.

1.5.1 Parameter Estimation

Given a relational database, MLN weights can in principle be learned generatively by maximizing the likelihood of this database on the closed world assumption. The gradient of the log-likelihood with respect to the weights is

$$\frac{\partial}{\partial w_i} \log P_w(X = x) = n_i(x) - \sum P_w(X = x') n_i(x') \quad (1.13)$$

where the sum is over all possible databases x' , and $P_w(X = x')$ is $P(X = x')$ computed using the current weight vector $w = (w_1, \dots, w_i, \dots)$. Unfortunately, computing these expectations can be very expensive. Instead, we can maximize the *pseudo-log-likelihood* of the data more efficiently. If x is a possible database and x_l is the l th ground atom's truth value, the *pseudo-log-likelihood* of x given weights w is

$$\log P_w^*(X = x) = \sum_{l=1}^n \log P_w \left(X_l = x_l \mid MB_x(X_l) \right) \quad (1.14)$$

where $MB_x(X_l)$ is the state of X_l 's *Markov blanket*⁴ in the data. Correspondingly, the gradient of the pseudo-log-likelihood is

$$\frac{\partial}{\partial w_i} \log P_w^*(X = x) = \sum_{l=1}^n \left[n_i(x) - P_w(X_{l=0} \mid MB_x(X_l)) n_i(x_{[X_l=0]}) - P_w(X_{l=1} \mid MB_x(X_l)) n_i(x_{[X_l=1]}) \right] \quad (1.15)$$

⁴The Markov blanket of a node is the minimal set of nodes that renders it independent of the remaining network; in a MLN, this is simply the node's neighbors in the graph.

where $n_i(x_{|X_i=0})$ is the number of true groundings of the i -th formula when we force $X_i = 0$ and leave the remaining data unchanged, and similarly for $n_i(x_{|X_i=1})$. Computing Equation 1.14 and its gradient does not require inference over the model, and is therefore much faster. We can optimize the pseudo-log-likelihood using the limited-memory BFGS algorithm [49]. The MLN parameters can also be learned discriminatively via efficient algorithms, as in [78] and [35].

1.5.2 Inference

If F_1 and F_2 are two formulas in first-order logic, C is a finite set of constants including any constants that appear in F_1 or F_2 , and L is an MLN, then

$$\begin{aligned}
 P(F_1 \mid F_2, L, C) &= P(F_1 \mid F_2, M_{L,C}) \\
 &= \frac{P(F_1 \wedge F_2 \mid M_{L,C})}{P(F_2 \mid M_{L,C})} \\
 &= \frac{\sum_{x \in \chi_{F_1} \cap \chi_{F_2}} P(X = x \mid M_{L,C})}{\sum_{x \in \chi_{F_2}} P(X = x \mid M_{L,C})}
 \end{aligned} \tag{1.16}$$

where χ_{F_i} is the set of worlds where F_i holds, and $P(x \mid M_{L,C})$ is given by Equation 1.12. The question of whether a knowledge base entails a formula F in first-order logic is the question of whether $P(F \mid L_{KB}, C_{KB,F}) = 1$, where L_{KB} is the MLN obtained by assigning infinite weight to all the formulas in KB, and $C_{KB,F}$ is the set of all constants appearing in KB or F .

A large number of efficient inference techniques are applicable to MLNs. The most widely used approximate solution to probabilistic inference in MLNs is Markov chain Monte Carlo (MCMC) [29]. In this framework, the Gibbs sampling algorithm is to generate an instance from the distribution of each variable in turn, conditional on the current values of the other variables. The key to the Gibbs sampler is that one only considers univariate conditional distributions — the distribution when all of the random variables but one are assigned fixed values. The probability of a ground atom X_i when its Markov blanket B_i is in state b_i is

$$\begin{aligned}
 P(X_i = x_i \mid B_i = b_i) &= \\
 &= \frac{\exp(\sum_{f_i \in F_i} w_i f_i(X_i = x_i, B_i = b_i))}{\exp(\sum_{f_i \in F_i} w_i f_i(X_i = 0, B_i = b_i)) + \exp(\sum_{f_i \in F_i} w_i f_i(X_i = 1, B_i = b_i))}
 \end{aligned} \tag{1.17}$$

where F_i is the set of ground formulas that X_i appears in, and $f_i(X_i = x_i, B_i = b_i)$ is the value (0 or 1) of the feature corresponding to the i -th

ground formula when $X_l = x_l$ and $B_l = b_l$. The estimated probability of a conjunction of ground literals is simply the fraction of samples in which the ground literals are true, after the Markov chain has converged.

One way to speed up Gibbs sampling is by Simulated Tempering [52], which performs simulation in a *generalized ensemble*, and can rapidly achieve an equilibrium state. [66] proposed MC-SAT, an inference algorithm that combines ideas from MCMC and satisfiability. MC-SAT works well and is guaranteed to be sound, even when deterministic or near-deterministic dependencies are present in real-world reasoning. Besides MCMC framework, maximum a posteriori (MAP) inference can be carried out using a weighted satisfiability solver like MaxWalkSAT. It is closely related to maximum likelihood (ML), but employs an augmented optimization objective which incorporates a prior distribution over the quantity one wants to estimate. MAP estimation can therefore be seen as a regularization of ML estimation.

1.6 Contributions

This thesis addresses the problem of joint information extraction, which is generally challenging and offers new opportunities for investigating. We propose several probabilistic graphical models, from cascaded to joint approaches, to deal with this problem. We focus on exact algorithms as well as approximate techniques where exact inference is intractable. We also propose tractable and efficient algorithms, including parameter estimation and inference, for these models. We perform extensive experiments on three representative IE tasks, and compare with current state-of-the-art models, to exhibit the feasibility and effectiveness of our proposed models. We now summarize the major contributions of this thesis as follows:

- We conduct preliminary investigation on a cascaded framework attempting to consider entity extraction and qualitative domain knowledge. This framework consists of two stages incorporating domain knowledge to capture the essential features of the Chinese named entity recognition (NER) task via MLNs. To the best of our knowledge, this is the first attempt at using MLNs for the NER problem in the NLP community. Compared to current state-of-the-art models for Chinese NER, this framework achieves promising results on both People's Daily corpus and official datasets (open track) in the Chinese NER shared task of the fourth SIGHAN Chinese language processing bake-off (SIGHAN-6).
- We propose 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 Metropolis-Hastings (MH) sampling algorithm is proposed to enable approximate inference, and this algorithm allows information to flow in both directions to capture mutual benefits. Notably, our model is considerably simpler to implement and requires much less engineering. It is also general and can be easily applied to a wide range of probabilistic models.

- We propose an integrated discriminative probabilistic approach to modeling both segmentations in sequence data and relations of segments simultaneously. This model combines the advantage of both probabilistic sequence models and first-order logic, and it offers a great flexibility to deal with uncertainty for modeling sequence data, and a variety of domain knowledge which can be concisely and easily formulated by first-order logic. This paradigm offers a natural way for information extraction which requires uncertainty modeling as well as dependency and deeper knowledge representation. We propose the Metropolis-Hastings, a theoretically well-founded MCMC algorithm, which consists of efficient sampling from segmentations to perform the maximum a posteriori (MAP) inference of this model. We also illustrate the benefits of this model for implicit relation extraction and new relation discovery, and sub-structures modeling in named entities.
- We propose a discriminative framework defining a joint probability distribution for both segmentations in sequence data and possible worlds of segment relations in the form of an exponential family. This joint model has several advantages over previous probabilistic graphical models. Since exact parameter estimation in this model can be too expensive or even intractable, we propose a structured variational inference algorithm to conduct approximate learning for the model's parameters. The variational inference method provides a fast, deterministic approximation to otherwise unattainable posteriors. Also its convergence time is independent of dimensionality. Moreover, we propose a highly-coupled, bi-directional Metropolis-Hastings (MH) sampling algorithm to enable efficient and tractable inference for this model, which allows information to flow in both directions and explores mutual benefits. Compared to the integrated model mentioned above, this joint model has several strong points: the integrated model is only loosely-

coupled in a “top-down” architecture, since the parameter estimation is performed independently for the two components. For inference, the information can only flow in one direction. In this joint paradigm, parameters for all subtasks are optimized simultaneously via structured variational approximation to capture deep interactions between different subtasks. Moreover, the inference is strongly bi-directional, thus information can flow in both directions to exploit mutual benefits.

1.7 Thesis Outline

Chapter 1 introduces the joint IE task, and graphical models for this task. We give some basics and necessary backgrounds, including directed and undirected graphical models, several variants of CRFs and MLNs. The rest of the chapters in the thesis is organized as follows:

Chapter 2. Related Work: In this chapter, we review some closely related models for IE tasks, including pipeline models, models incorporating probability with logic, integrated and joint models. We compare our approaches with these models, pointing out some shortcomings of these models and the superiority of our proposed approaches over these models.

Chapter 3. A Preliminary Study: We develop a cascaded architecture incorporating probabilistic graphical models and first-order logic for Chinese NER, as a preliminary study. This architecture captures a variety of linguistic characteristics in Chinese NEs as domain knowledge, and formulates them into first-order logic easily and concisely. Using linear-chain CRFs as a base model, we do error analysis, describe well-engineered features and domain knowledge, and show how the domain knowledge can be represented into first-order logic to conduct relational learning. We apply and test our framework on both People’s Daily corpus and official datasets in the Chinese NER shared task of the fourth SIGHAN Chinese language processing bakeoff (SIGHAN-6), and our proposed model outperforms previous state-of-the-art models and achieves consistently high performance. For example, our system won the first place on the CityU open track and fourth place on the MSRA open track in the SIGHAN-6, respectively. Our proposed framework can also be extendable to NER for other languages, due to the simplicity of the domain knowledge we could access.

Chapter 4. Bidirectional Integration of Pipeline Models: We present

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. And parameter estimation can be performed efficiently using evidences from multiple subtasks, such that they aid each other to boost the performance. We then propose a strong bidirectional algorithm based on efficient Markov chain Monte Carlo (MCMC) sampling to enable tractable inference, which allows information to flow bidirectionally and mutual benefits from different subtasks can be well exploited. Our framework 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.

Chapter 5. An Integrated Discriminative Probabilistic Approach:

Motivated by mining implicit relations and new relation discovery, and capturing sub-structures in named entities, in this chapter, we combine the advantages of both probabilistic graphical models for sequence data and first-order logic in a principled way, resulting in an integrated discriminative probabilistic framework which models both segmentations in sequence data and relations of different segments simultaneously for IE tasks. We propose the Metropolis-Hastings, a MCMC algorithm for approximate Bayesian inference to find the maximum a posteriori (MAP) assignment of all the variables of this model. We perform parameter estimation somewhat separately for this integrated model, to reduce the model complexity.

Chapter 6. Joint Models Incorporating Logic: In this chapter, we formally define the problem of joint optimization of information extraction, and propose a joint discriminative probabilistic framework to optimize all relevant subtasks simultaneously. This framework offers a great flexibility to incorporate the advantage of both uncertainty for sequence modeling and first-order logic for domain knowledge. The first-order logic model provides a more expressive formalism tackling the issue of limited expressiveness of traditional attribute-value representation. Our framework defines a joint probability distribution for both segmentations in sequence data and possible worlds of relations between segments in the form of an exponential family. Since exact parameter estimation and inference are prohibitively intractable in this model, a structured variational inference algorithm is developed to perform parameter estimation approximately. For inference, we propose a

highly-coupled, bi-directional Metropolis-Hastings (MH) algorithm to find the maximum a posteriori (MAP) assignments for both segmentations and relations.

Chapter 7. Experiments: We apply and test our proposed models, including the bidirectional integrated models in Chapter 4, the integrated discriminative probabilistic models in Chapter 5, and the joint models incorporating first-order logic in Chapter 6, on three important real-world IE tasks, namely Chinese NER, entity identification and relation extraction from Wikipedia, and citation matching. Extensive experimental study shows that our proposed models achieve substantial improvement over current state-of-the-art models, demonstrating the effectiveness and feasibility of our approaches. In addition, the superiority of the bi-directional MH algorithm over the greedy, N -best list, and uni-directional MH sampling algorithms is also analyzed and compared.

Chapter 8. Conclusions and Future Work: We review the main contributions of the thesis and summarize their significance and applicability. We discuss extensions and future research directions not addressed in the thesis. For example, these proposed models allow extensive further investigation, both for parameter learning and inference algorithms. Feature engineering is also an important issue to seek further gains of these models. We plan to apply and test our models to other real-world IE applications.

1.8 Publications Generated

Some of the work described in this thesis has been published. The content of Chapter 3 was published in [94, 100] and in [101] which mainly presents the official results in SIGHAN-6. Work on bidirectional integrated models (Chapter 4), integrated models combining probabilistic graphical models for sequence data and first-order logic (Chapter 5) was published in [98] and [102], respectively. Some content in Chapter 7 using MLNs for encyclopedia relation extraction with multiple features was also published in [97]. A paper on probabilistic joint models incorporating first-order logic and learning via structured variational approximation (Chapter 6) [99] is currently under review.

In addition to the above publications directly generated from this research work, some other indirectly related publications have not been included. A

cascaded approach based on discriminative probabilistic models for biomedical named entity recognition was published in [13] and [14] (joint work with Shing-Kit Chan). A hidden dynamic conditional random field (HDCRF) model which can capture both internal and external class dynamics to label sequence data was presented in [96]. And work on using expressive logic models for coreference resolution was published in [12] (joint work with Ki Chan).

Chapter 2

Related Work

2.1 Pipeline Models for IE

Most existing approaches to compound, aggregate IE problems are in pipeline or decoupled architecture — attempting to perform compound tasks in several separate, and independent stages. This decoupled strategy is ubiquitous in IE [94][25][34], in which stages are run in some order, and later stages have access to the output of completed earlier stages. The simplest way is the 1-best feed forward architecture which greedily takes the best output at each stage in the pipeline and pass it on to the next stage. While comparatively easy to assemble and computationally efficient, this pipeline approach is highly ineffective and we summarize the shortcomings as follows [25][67]: (1) Error Propagation: since many stages are performed separately and independently in decoupled architecture, errors accumulate as information progresses through the pipeline, and an error once made in the previous stages can hardly be corrected in the current stage. It is therefore disappointing, but not surprising, that the overall performance is limited and upper-bounded. For example, if we naively use the single most likely output of a part-of-speech tagger as the input to a syntactic parser, and those parse trees as the input to a coreference system, and so on, errors in each step will propagate to later ones: each components 90% accuracy multiplied through six components becomes only 53%. (2) Lack of Mutual Interactions: the pipeline architecture does not capture the dependencies and interactions between different stages. In the relation extraction task, for example, knowing the entities (both entity types and boundaries) is very helpful for extracting relations between them. Also, knowing the relation between two entities is useful for entity identification (e.g., the *employment* relation can only exist between an *organization* and a *person*, and cannot exist between an *or-*

ganization and a *location*, or a *location* and a *person*). Unfortunately, the pipeline architecture does not achieve these interactions, resulting in reduced performance. (3) Lack of Long-distance Dependency Modeling: in information extraction, long-distance dependencies always exist between different attribute elements. And there are always many irrelevant elements or noise elements appearing between the attributes. However, flat models like linear-chain CRFs [47] cannot incorporate long-distance dependencies because of their first-order Markov assumption.

A common improvement on this architecture is to pass N -best lists between processing stages, and this usually gives useful improvements. However, efficiently enumerating N -best lists often requires very substantial cognitive and engineering effort [86]. At the other extreme, one can maintain the entire space of representations (and their probabilities) at each level, and use this full distribution to calculate the full distribution at the next level. In most cases, maintaining entire probability distributions is usually infeasible, because for most intermediate tasks, there is an exponential number of possible labelings. Doing this normally also involves a very high cognitive and engineering effort, and in practice this solution is infrequently adopted.

Some work focused on improving the pipeline architecture [25][34]. Finkel et al. [25] modeled pipelines as Bayesian networks, with each low level task corresponding to a variable in the network. This framework samples the output for each component, then the pipeline is run repeatedly so that different combinations of output throughout the pipeline are evaluated. This architecture has the drawback that it only allows information to flow in one direction. Hollingshead and Roark [34] proposed an approach that uses output from later stages of a pipeline to constrain earlier stages of the same pipeline iteratively. All these approaches suffer from inherent inferiority such as brittle accumulation of errors caused by their pipeline architecture.

2.2 Incorporating Probability with Logic

Some work dedicated to combining probability and first-order logic. One major challenge of logic is its insufficient handling of uncertainty and fuzziness. One early work is probabilistic logic programming (PLP) which places constraints on distributions [32]. It makes use of a logic program syntax and the concept of least Herbrand model to specify the random variables. A recent approach known as Bayesian logic programming (BLP) [41] treats atoms as random variables whereas PLP treats atoms as states of random variables. Besides, two common models that received some attention are relational Bayesian networks (RBNs) [38] and probabilistic relational models

(PRMs) [26]. RBNs are basically Bayesian networks whose nodes are the extensions of first-order predicates. In other words, each node is the assignment to the set all atoms of a certain predicate. Needless to say, inference in such a network would be extremely inefficient since each node would have an extremely large number of values. PRMs make use of directed graphical models that bring a notion of causality. The need to avoid cycles in PRMs causes significant representational and computational difficulties. Inference in PRMs is done by creating the complete ground network, which limits their scalability. PRMs require specifying a complete conditional model for each attribute of each class, which in large complex domains can be quite burdensome. In relational domains it is often the case that random variables depend on each other without a clear notion of causality. For this reason, relational Markov networks (RMNs) [85] recast PRMs so they generate undirected graphical models (Markov networks) instead of Bayesian networks. RMNs use database queries as clique templates, and have a feature for each state of a clique. RMNs are exponential in clique size, and do not specify a complete joint distribution for the variables in the model. The disadvantage of RMNs is that learning in undirected graphical models is harder than in directed ones. RMNs use MAP estimation with belief propagation for inference, which makes learning quite slow, despite the simplified discriminative setting. However, these models have not been applied to large-scale IE problems.

Another branch of models is first-order probabilistic languages (FOPLs) [60] which explicitly represent objects and relations between them. BLOG [57] is one such approach based on generative models and has been applied to solve text mining problems [11]. A BLOG model is basically defined by a generative process for tackling situations where possible worlds with varying object sets and identity uncertainty. However, BLOG does not allow first-order knowledge to be easily incorporated. BLOG is generative and is constrained to assume independence or explicitly model the causal interactions between features of data. Our proposed model is discriminative, and can capture complex dependencies among inputs. Culotta et al. [22] attempted to incorporate domain knowledge into CRFs. They designed a restricted form of first-order logic, which is basically cluster-wise compatibility of the observed tokens. Such logic is then incorporated into CRFs in the form of features which can only capture some raw facts observed from data and are unable to conduct logical inference. Consequently, both expressiveness and reasoning power are very limited.

Markov logic networks (MLNs) [69] combine first-order logic and probabilistic graphical models in a single framework, and a joint inference method based on MLNs was proposed in [67], where segmentation of all records and

entity resolution are performed together in a single MLN framework for citation matching. This method mainly consists of designing some logical formulas to capture interactions between segmentation and resolution. However, the inference is not strongly bi-directional but only weakly-coupled, and do not enforce transitivity. Since the logic formulas only examine pairs of consecutive labels, not whole fields. For this reason, citation coreference compatibility is measured using features of the un-segmented citation. Our proposed model is strongly-coupled and enforces transitivity, and significantly outperforms the single MLN model in [67]. Zhu et al. [106] used MLNs to extract relationships between entities and built an entity relation extraction system called *StatSnowball*. Similar to [21][97], the entities were extracted and known in advance in [106]. Our tasks involving joint entity identification and relation extraction, and joint segmentation and entity resolution are more difficult, and offer new opportunities for information extraction.

2.3 Integrated and Joint Models for IE

Integrated and joint models exploring mutual benefits on different tasks have shown great promise in many areas including natural language processing, data mining and information extraction. Some early work includes [59, 53, 9, 88]. Pasula et al. [59] developed a “collective” model for citation matching, but performed segmentation in a pre-processing stage, allowing boundaries to occur only at punctuation marks. This model required considerable engineering. [59] combined several models with separately learned parameters, a number of hard-wired components, and data from a variety of sources (including a database of names from the 2000 US Census, manually-segmented citations, and a large AI BibTex bibliography). Bunescu and Mooney [9] employed relational Markov networks (RMNs) [85] to represent influences and dependencies between different extractions. Dependencies must be defined in the model structure and [9] used crude heuristic part-of-speech patterns. Another disadvantage of this approach is that it uses loopy belief propagation and a voted perceptron for approximate learning and inference – ill-founded and inherently unstable algorithms which are noted by the authors to have caused convergence problems. Using first-order logic formulism, our model allows a much broader class of relations and dependencies. The experiment in [9] applied joint segmentation to protein name extraction, but did not perform entity resolution. We perform two IE tasks using real-world data from Wikipedia and Cora. Mccallum and Jensen [53] advocated the use of joint probabilistic models that perform extraction and data mining in an integrated inference procedure — the evidence for an

outcome being the result of simultaneously making inferences both “bottom up” from extraction, and “top down” from data mining. And Wellner et al. [88] proposed one such model for citation matching. Wellner et al. [88] extended the pipeline model by passing uncertainty from the segmentation phase to the entity resolution phase, and by including a one-time step from resolution to segmentation. However, this model is not a real joint model since it did not “close the loop” by repeatedly propagating information between different substructures. Separate learning and inference is employed to reduce the model complexity. The used N -best list for inference is a restricted approximation for the full distribution of large-output components. In contrast, by training all parameters simultaneously, our model captures deep interactions between substructures from two relevant tasks. Using MH sampling for inference bi-directionally, the full probability distribution can be better approximated and mutual benefits can be gained.

Recently, Zhu et al. [111] proposed an integrated probabilistic approach to Web page understanding. In this “top-down” integration model, the decision of the upper hierarchical CRF model [108][48] could guide the decision of the bottom semi-CRF model. However, the drawback of the top-down architecture is that the decision of the semi-CRF model can hardly be used by the hierarchical CRF model to refine its decision-making. Later they proposed dynamic, hierarchical models to incorporate structural uncertainty for Web data extraction [109]. A simple variational mean field approach is exploited in their method. As stated in Section 6.2.4, the structured variational approximation in our model is more general than the simple variational approach by exploiting tractable substructures. The superiority is that, deep interactions between entities and relations can be captured and mutual benefits between two tasks can be exploited better. Moreover, one major difference between our model and [111][109] is that our model has the advantage of incorporating the expressiveness of first-order logic and a variety of domain knowledge.

More recently, Yang et al. [93] improved the model in [111] by introducing additional potential functions capturing dependencies between two sub-models in [111]. The resulting framework is a bi-directional integration of page structure understanding and text understanding. Our work differs by several modeling choices: combining probability with first-order logic instead of pure probability, joint parameter learning via structured variational approximation instead of separate training, bi-directional sampling instead of 1-best iteration. Luo et al. [50] combined Web classification and Web information extraction based on the CRF model. However, since it was defined according to the DOM tree structure, this model cannot be applied to our task.

Chapter 3

A Preliminary Study

3.1 A Cascaded Approach

Named entity recognition (NER) [65] is the task of identifying and classifying phrases that denote certain types of named entities (NEs), such as person names (PERs), locations (LOCs) and organizations (ORGs) in text documents. The NER task is, given a sentence, first to segment which words are part of entities, and then to classify each entity by type (PER, LOC, ORG, and OTHER (meaning not an entity)). The challenge of this problem is that many named entities are too rare to appear even in a large training set, and therefore the system must identify them based only on context. It is a well-established task in the NLP and data mining communities and is regarded as crucial technology for many higher-level applications, such as information extraction, question answering, information retrieval and knowledge management. The NER problem has generated much interest and great progress has been made, as evidenced by its inclusion as an understanding task to be evaluated in the Message Understanding Conference (MUC) [2], the Automatic Content Extraction (ACE) evaluation [1], and the Conference on Computational Natural Language Learning (CoNLL) [4][3].

CRFs [47][82] have been the state-of-the-art model adopted in a variety of IE tasks (e.g., NER), achieving very good performance. The basic idea of CRF-based methods is to formulate the IE problem as a sequence labeling task. However, one disadvantage of sequence labeling models, such as CRF-based methods, is the limited expressiveness of attribute-value representation of features. While attribute-value representation is suitable for statistical learning approaches. They cannot handle text mining problems involving complex knowledge which requires richer representational power facilitating logical inference or reasoning.

Another major limitation of sequence labeling models is their incapability of incorporating entity-level domain knowledge commonly found in practical situations. For example, one useful feature is the degree of matching between the candidate entity and the entries found in a specialized lexicon. Such feature requires the processing of the entire candidate entity. On the other hand, sequence learning methods operate on the token level, while this kind of feature needs to work on the whole entity level. So there is a fundamental mismatch in representation resulting in difficulty in adopting such kind of feature.

Logic is a powerful paradigm that can overcome the knowledge representation problems mentioned above found in sequence learning methods. It provides much more expressive power than attribute-value representation of features. It can capture complex entity-level domain knowledge commonly found in human experts or in practical situations. Generally, incorporating logic into text learning models is quite challenging. One major challenge of logic is its handling of uncertainty and fuzziness which is common in text.

We have investigated and developed a two-stage cascaded framework in an attempt to consider entity extraction and qualitative domain knowledge based on the combination of statistical learning and first-order logic. First, we employ conditional random fields (CRFs), a discriminatively trained undirected graphical model which has theoretical justification and has been shown to be an effective approach to segmenting and labeling sequence data, as our base system. We then exploit a variety of domain knowledge into Markov logic networks (MLNs), a powerful combination of logic and probability, to validate and correct errors made in the base system. We show how a variety of domain knowledge can be formulated as first-order logic and incorporated into MLNs. We use three Markov chain Monte Carlo (MCMC) algorithms, including Gibbs sampling, Simulated Tempering, as well as MC-SAT, and Maximum a posteriori/Most Probable Explanation (MAP/MPE) algorithm for probabilistic inference in MLNs.

3.2 Framework Overview

We propose a framework based on probabilistic graphical models with first-order logic. As shown in Figure 3.1, the framework is composed of three main components. The CRF model is used as a base model. Then we incorporate domain knowledge that can be well formulated into first-order logic to extract entity candidates from CRF results. Finally, the MLN, an undirected graphical model for *statistical relational learning*, is used to validate and correct the errors made in the base model.

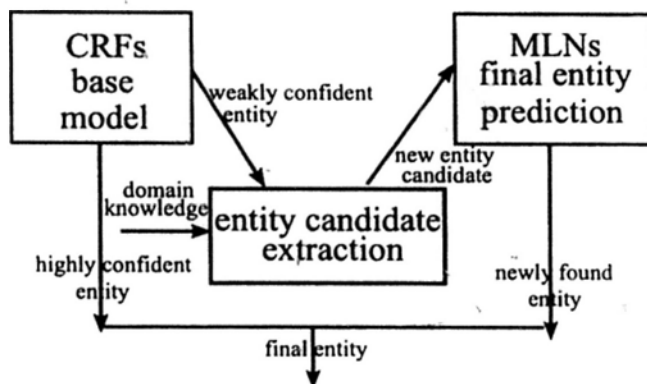


Figure 3.1: Framework overview

3.2.1 Applying to Chinese NER

Compared to European-language NER, Chinese NER¹ seems to be more difficult [95]. Recent approaches to Chinese NER are a shift away from manually constructed rules or finite state patterns towards machine learning or statistical methods. However, rule-based NER systems lack robustness and portability. Statistical methods often suffer from the problem of data sparsity, and machine learning approaches (e.g., Hidden Markov Models (HMMs) [6, 104], Support Vector Machines (SVMs) [36], Maximum Entropy (Max-Ent) [7, 17], Transformation-based Learning (TBL) [8] or variants of them) might be unsatisfactory to learn linguistic information in Chinese NERs. Current state-of-the-art models often view Chinese NER as a *classification* or *sequence labeling* problem *without* considering the linguistic and structural information in Chinese NERs. They assume that entities are independent, however in most cases this assumption does not hold because of the existing relationships among the entities. They seek to locate and identify named entities in text by sequentially classifying tokens (words or characters) as to whether or not they participate in an NE, which is sometimes prone to noise and errors.

In fact, Chinese NERs have distinct linguistic characteristics in their composition and human beings usually use prior knowledge to recognize NERs. For example, about 365 of the highest frequently used surnames cover 99% Chinese surnames [81]. Some LOCs contain location salient words, while some ORGs contain organization salient words. For the LOC “香港特区/Hong Kong Special Region”, “香港/Hong Kong” is the name part and “特

¹In this thesis we only focus on PERs, LOCs and ORGs. Since temporal, numerical and monetary phrases can be well identified with rule-based approaches.

“区/Special Region” is the salient word. For the ORG “香港特区政府/Hong Kong Special Region Government”, “香港/Hong Kong” is the LOC name part, “特区/Special Region” is the LOC salient word and “政府/Government” is the ORG salient word. Some ORGs contain one or more PERs, LOCs and ORGs. A more complex example is the nested ORG “北京市海淀区清华大学计算机学院/School of Computer Science, Tsinghua University, Haidian District, Beijing City” which contains two ORGs “清华大学/Tsinghua University” and “计算机学院/School of Computer Science” and two LOCs “北京市/Beijing City” and “海淀区/Haidian District”. The two ORGs contain ORG salient words “大学/University” and “学院/School”, while the two LOCs contain LOC salient words “市/City” and “区/District” respectively.

In the case of Chinese NER, a named entity can be connected to another named entity for instance, because they share the same location salient word. Thus in an undirected graph, two node types exist, the LOC nodes and the location salient word nodes. The links (edges) indicate the relation (LOCs contain location salient words) between them. This representation can be well expressed by MLNs.

However, one problem concerning relational data is, how to extract useful relations for Chinese NER. There are many kinds of relations between NEs, some relations are critical to the NER problem while others are not. Another problem that we address is whether these relations can be formulated in first-order logic and combined in MLNs. In Section 3.2.5, we exploit domain knowledge. We will show how these knowledge can capture essential characteristics of Chinese NEs and can be well and concisely formulated in first-order logic in Section 3.2.6.

3.2.2 CRFs as Base Model

Recently, CRFs have been shown to perform exceptionally well on Chinese NER shared task on the third SIGHAN Chinese language processing bake-off (SIGHAN-04) ([105], [15], [16]). We follow the state-of-the-art CRF models using features that have been shown to be very effective in Chinese NER, namely the current character and its part-of-speech (POS) tag, several characters surrounding (both before and after) the current character and their POS tags, current word and several words surrounding the current word, and dictionary features. In addition, we exploit clue word features which can capture non-local dependencies. We employ 412 career titles (e.g., 总统/President, 教授/Professor, 警察/Police), 59 family titles (e.g., 爸爸/Father, 妹妹/Sister), 33 personal pronouns (e.g., 你们/Your, 我们/We) and 109 direction words (e.g., 以北/North, 南部/South) to represent non-local information. Career titles, family titles and personal pronouns may

Susam is an American economics professor
苏珊 是 一名 美国 经济学 教授

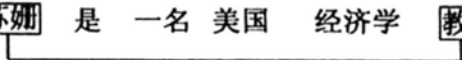


Figure 3.2: An example of non-local dependency. The career title “教授” indicates a PER “苏珊”

imply a nearby PER and direction words may indicate a LOC or an ORG. Figure 3.2 illustrates an example of non-local dependency. This gives us a competitive baseline CRF model using both local and non-local information for Chinese NER.

We also observe some important issues that significantly influence the performance as follows:

Window size: The primitive window size we use is 5 (2 characters preceding the current character and 2 following the current character). We extend the window size to 7 but find that it slightly hurts. The reason is that CRFs can deal with non-independent features. A larger window size may introduce noisy and irrelevant features.

Feature representation: For character features, we use character identities. For word features, BIES representation (each character is beginning of a word, inside of a word, end of a word, or a single word) is employed.

Labeling scheme: The labeling scheme can be BIO, BIOE or BIOES representation. In BIO representation, each character is tagged as either the beginning of a named entity (B), a character inside a named entity (I), or a character outside a named entity (O). In BIOE, the last character in an entity is labeled as E while in BIOES, single-character entities are labeled as S. In general, BIOES representation is more informative and yields better results than both BIO and BIOE.

3.2.3 Error Analysis

Even though the CRF model is able to accommodate a large number of well-engineered features which can be easily obtained across languages, some NEs, especially LOCs and ORGs are difficult to identify due to the lack of linguistic or structural characteristics. Since predictions are made token by token, some typical and serious tagging errors are still made, as shown below:

- **ORG is incorrectly tagged as LOC:** In Chinese, many ORGs contain location information. The CRF model only tags the location information (in the ORGs) as LOCs. For example, “唐山理工学院/Tangshan Techni-

cal Institute” and “海南省省委/Hainan Provincial Committee ” are ORGs and they contain LOCs “唐山/Tangshan” and “海南省/Hainan Province”, respectively. “唐山/Tangshan” and “海南省/Hainan Province” are only incorrectly tagged as LOCs. This affects the tagging performance of both ORGs and LOCs.

- **LOC is incorrectly tagged as ORG:** The LOCs “悉尼歌剧院/Sydney Opera” and “北京体育馆/Beijing Gymnasium” are mistakenly tagged as ORGs by the CRF model without taking into account the location salient words “歌剧院/Opera” and “体育馆/Gymnasium”.
- **The boundary of entity is tagged incorrectly:** This mistake occurs for all the entities. For example, the PER “汤姆·克鲁斯/Tom Cruise” may be tagged as a PER “汤姆/Tom”; the LOC “不来梅/Bremen” may be tagged as a LOC “来梅/Laimei”, which is a meaningless word; the ORG “华为公司/Huawei Corporation” may be tagged as an ORG “华为/Huawei”. The reasons for these errors are both complicated and varied. However, some of them are related to linguistic knowledge.
- **Common nouns are incorrectly tagged as entities:** For example, the two common nouns “现代数学/Modern Mathematics” and “格兰士微波炉/Galanz Microwave Oven” may be improperly tagged as a LOC and an ORG. Some tagging errors could be easily rectified. Take the erroneous ORG “市委组织, /City Committee Organizes,” for example, intuitively it is not an ORG since an entity cannot span any punctuation.

3.2.4 MLNs as Error Correction Model

We model the linguistic and structural information in Chinese named entity composition. We exploit a variety of domain knowledge which can capture essential characteristics of Chinese named entities into MLNs, a powerful combination of first-order logic and probability, to (1) validate and correct errors made in the base system and (2) find and extract new entity candidates. These domain knowledge is easy to obtain and can be well and concisely formulated in first-order logic and incorporated into MLNs.

MLNs conduct *relational learning* by incorporating first-order logic into probabilistic graphical models under a single coherent framework [69]. Traditional first-order logic is a set of hard constraints in which a world violates even one formula has zero probability. The advantage of MLNs is to soften these constraints so that when the fewer formulae a world violates, the more probable it is. MLNs have been applied to tackle the problems of gene interaction discovery from biomedical texts and citation entity resolution from citation texts with state-of-the-art performance ([70], [79]).

3.2.5 Domain Knowledge

We incorporate various kinds of domain knowledge via MLNs to predict the newly extracted NE candidates from CRF hypotheses. We extract 165 location salient words and 843 organization salient words from Wikipedia² and the LDC Chinese-English bi-directional NE lists compiled from Xinhua News database, as shown in Table 3.1. We also make a punctuation list which contains 18 items and some stopwords which Chinese NEs cannot contain. The stopwords are mainly conjunctions, auxiliary and functional words. We extract new NE candidates from the CRF results according to the following consideration:

- Definitely, if a chunk (a series of continuous characters) occurs in the training data as a PER or a LOC or an ORG, then this chunk should be a PER or a LOC or an ORG in the testing data. In general, a unique string is defined as a PER, it cannot be a LOC somewhere else.
- Obviously, if a tagged entity ends with a location salient word, it is a LOC. If a tagged entity ends with an organization salient word, it is an ORG.
- If a tagged entity is close to a subsequent location salient word, probably they should be combined together as a LOC. The closer they are, the more likely that they should be combined.
- If a series of consecutive tagged entities are close to a subsequent organization salient word, they should probably be combined together as an ORG because an ORG may contain multiple PERs, LOCs and ORGs.
- Similarly, if there exists a series of consecutive tagged entities and the last one is tagged as an ORG, it is likely that all of them should be combined as an ORG.
- Entity length restriction: all kinds of tagged entities cannot exceed 25 Chinese characters.
- Stopword restriction: intuitively, all tagged entities cannot comprise any stopword.
- Punctuation restriction: in general, all tagged entities cannot span any punctuation.
- Since all NEs are proper nouns, the tagged entities should end with noun words.

²<http://en.wikipedia.org/wiki/>.

Table 3.1: Domain knowledge for Chinese NER

Location Salient Word	Organization Salient Word
自治区/Municipality	百货公司/Department Store
火车站/Railway Station	理工学院/Technical Institute
宾馆/Hotel	旅行社/Travel Agency
公园/Park	出版社/Press
高原/Plateau	人事部/Personnel Department
省/Province	银行/Bank
镇/Town	大学/University
市/City	市委/City Committee
Stopword	Punctuation
仍然/still	。
但是/but	？
非常/very	，
的/of	；
等/and so on	：
那/that	！

- The CRF model tags each token (Chinese character) with a conditional probability. A low probability implies a low-confidence prediction. For a chunk with low conditional probabilities, all the above assumptions are adopted (The marginal probabilities are normalized, and probabilities lower than the user-defined threshold are regarded as low conditional probabilities).

All the above domain knowledge can be formulated as first-order logic to construct the structure of MLNs. And all the extracted chunks are accepted as new NE candidates (or common nouns). We train an MLN to recognize them.

3.2.6 First-order Logic Construction

We declared 14 *predicates* (person(candidate), location(candidate), organization(candidate), endwith(candidate, salientword), closeto(candidate, salientword), containstopword(candidate), containpunctuation(candidate), etc) and specified 15 first-order formulas (See Table 7.5 for some examples) according to the domain knowledge described in Section 3.2.5. For example, we used person(candidate) to specify whether a candidate is a PER. *Formulas* are recursively constructed from atomic formulas using logical connectives and quantifiers. They are constructed using four types of symbols: *constants*, *variables*, *functions*, and *predicates*. *Constant* symbols represent objects in the domain of interest (e.g., “北京/Beijing” and

Table 3.2: Examples of NE candidates and first-order formulas

Mis-tagged NEs	New NE Candidates	First-order Logic
希拉里[common noun]	希拉里	$\text{occurperson}(p) \Rightarrow \text{person}(p)$
凡尔赛[PER]	凡尔赛	$\text{occurlocation}(p) \Rightarrow \text{location}(p)$
一汽集团[common noun]	一汽集团	$\text{occurorganization}(p) \Rightarrow \text{organization}(p)$
乌市[ORG]	乌市	$\text{endwith}(r, p) \wedge \text{locsalientword}(p) \Rightarrow \text{location}(r)$
英政府[LOC]	英政府	$\text{endwith}(r, p) \wedge \text{orgsalientword}(p) \Rightarrow \text{organization}(r)$
北海[LOC]花园	北海花园	$\text{closeto}(r, p) \wedge \text{locsalientword}(p) \Rightarrow \text{location}(r)$
瑞士[LOC]联邦	瑞士联邦	$\text{closeto}(r, p) \wedge \text{orgsalientword}(p) \Rightarrow \text{organization}(r)$
市区的酒店[LOC]	市区的酒店	$\text{containstopword}(p) \Rightarrow \neg(\text{person}(p) \vee \text{location}(p) \vee \text{organization}(p))$
“百帮”服务中心[ORG]	“百帮”服务中心	$\text{containpunctuation}(p) \Rightarrow \neg(\text{person}(p) \vee \text{location}(p) \vee \text{organization}(p))$

“上海/Shanghai” are LOCs). *Variable* symbols (e.g., r and p) range over the objects in the domain. To reduce the size of ground Markov Network, variables and constants are *typed*; for example, the variable r may range over candidates, and the constant “北京/Beijing” may represent a LOC. *Function* symbols represent mappings from tuples of objects to objects. *Predicate* symbols represent relations among objects (e.g., person) in the domain or attributes of objects (e.g., endwith). A *ground atom* is an atomic formula all of whose arguments are ground terms (terms containing no variables). For example, the ground atom $\text{location}(\text{北京市})$ conveys that “北京市/Beijing City” is a LOC.

For example in Table 7.5, “乌市/Wu City” is mis-tagged as an ORG by the CRF model, but it contains the location salient word “市/City”. So it is extracted as a new entity candidate, and the corresponding formula $\text{endwith}(r, p) \wedge \text{locsalientword}(p) \Rightarrow \text{location}(r)$ means if r ends with a location salient word p , then it is a LOC. Besides the formulas listed in Table 7.5, we also specified logic such as $\text{person}(p) \Rightarrow \neg(\text{location}(p) \vee \text{organization}(p))$, which means a candidate p can only belong to one class.

We assume that the relational database contains only binary relations. Each extracted NE candidate is represented by one or more strings appearing as arguments of ground atoms in the database. The goal of NE prediction is to determine whether the candidates are entities and the types of entities (query predicates), given the evidence predicates and other relations that

can be deterministically derived from the database. As we will see, despite their simplicity and consistency, these first-order formulas incorporate the essential features for NE prediction.

3.2.7 Implementation and Model Development

We use CRF++ toolkit (version 0.48) [46] for the base model in our implementation. We find that setting the cut-off threshold f for the features not only decreases the training time, but improves the NER performance. CRFs can use the features that occurs no less than f times in the given training data. We set $f = 5$ in our system. We use the Alchemy system (Beta version) [43] for the error correction model, which is a software package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation. To avoid over-fitting for the CRF model, we penalized the log-likelihood by the commonly used zero-mean Gaussian prior over the parameters. Also, the MLNs were trained using a Gaussian prior with zero mean and unit variance on each weight to penalize the pseudo-likelihood, and with the weights initialized at the mode of the prior (zero). We performed holdout methodology to develop both the base and error correction models.

3.3 Experiments on People’s Daily Corpus

3.3.1 Data

We used People’s Daily corpus (January-Jun, 1998) in our experiments, which contains approximately 357K sentences, 156K PERs, 219K LOCs and 87K ORGs, respectively. We did some modifications on the original data to make it cleaner. We enriched some tags so that the abbreviation proper nouns are well labeled. We preprocessed some nested names to make them in better form. We also processed some person names. We enriched tags for different kinds of person names (e.g., Chinese and transliterated names) and separated consecutive person names. To reduce the training time, we use one-month corpus for training and 9-day corpus for testing.

3.3.2 The Baseline NER System

We use CRFs to build a character-based Chinese NER system, with features described in Section 3.2.2. We do not take the advantage of using the

golden-standard word segmentation and POS tagging provided in the original corpus, since such information is hardly available in real text. Instead, we use an off-the-shelf Chinese lexical analysis system, the open source ICT-CLAS [103], to segment and POS tag the corpus. This module employs a hierarchical hidden Markov model (HHMM) and provides word segmentation, POS tagging (labels Chinese words using a set of 39 tags) and unknown word recognition. It performs reasonably well, with segmentation precision recently evaluated at 97.58%. The recall of unknown words using role tagging is over 90%.

3.3.3 Experimental Results

To test the effectiveness of our proposed model, we extract all the NEs (19,879 PERs, 25,661 LOCs and 11,590 ORGs) from the training corpus, and then convert them to the first-order logic representation according to the domain knowledge. An MLN training database, which consists of 14 predicates, 16,620 constants and 97,992 ground atoms was built. We also extract new entity candidates from CRF results and construct MLN testing database in the same way. During MLN learning, each formula is converted to Conjunctive Normal Form (CNF), and a weight is learned for each of its clauses. The weight of a clause is used as the mean of a Gaussian prior for the learned weight. These weights reflect how often the clauses are actually observed in the training data.

We extract 529 entity candidates to construct the MLN testing database, which contains 2,543 entries and these entries are used as evidence for inference. Inference is performed by grounding the minimal subset of the network required for answering the query predicates. We employed 3 MCMC algorithms: Gibbs sampling (GS), Simulated Tempering (ST) as well as MC-SAT, and the MAP/MPE algorithm for inference and the comparative NER results are shown. The probabilistic graphical models greatly outperform the CRF model stand-alone by a large margin. It can be seen from Table 3.3 and Table 3.4, the probabilistic graphical models integrating first-order logic improve the precision and recall for all kinds of entities, thus boosting the overall F-measure. We achieve a 23.75% relative error reduction (RER) on F-measure by using 3 MCMC algorithms and a 20.54% RER by using MAP/MPE algorithm, over an already competitive CRF baseline. We obtained the same results using GS, ST and MC-SAT algorithms. MCMC algorithms yields slightly better results than the MAP/MPE algorithm.

3.3.4 Significance Test

Ideally, comparisons among NER systems would control for feature sets, data preparation, training and test procedures, parameter tuning, and estimate the statistical significance of performance differences. Unfortunately, reported results sometimes leave out details needed for accurate comparisons.

We give statistical significance estimates using McNemar’s paired tests³ [30] on labeling disagreements for CRF model and graphical probabilistic models that we evaluated directly.

Table 3.5 summarizes the correctness of the labeling decisions between the models with a 95% confidence interval (CI). These tests suggest that the graphical probabilistic models are significantly more accurate and confirm that the gains we obtained are statistically highly significant.

Table 3.5: McNemar’s tests on labeling disagreements

Null Hypothesis	95% CI	p-value
Proposed Model (GS) vs. CRFs	5.71-9.52	$< 1 \cdot 10^{-6}$
Proposed Model (ST) vs. CRFs	5.71-9.52	$< 1 \cdot 10^{-6}$
Proposed Model (MC-SAT) vs. CRFs	5.71-9.52	$< 1 \cdot 10^{-6}$
Proposed Model (MAP/MPE) vs. CRFs	4.50-7.37	$< 1 \cdot 10^{-6}$

3.4 Official Results in SIGHAN-6

Recently, we have participated in the Chinese named entity recognition (NER) task for the fourth SIGHAN Chinese language processing bakeoff (SIGHAN-6) [39], which provides large-scale benchmark data for evaluation. We submitted results for the open track of the NER task. Among all the groups participating the official evaluation, we obtained the best performance on the CityU corpus and the fourth place on the MSRA corpus. Moreover, we were the only group that obtained consistently over 90 F-measure on all the benchmark corpora in the NER open track.

3.4.1 Data and Preprocessing

The training corpora provided by the SIGHAN bakeoff organizers were in the CoNLL two column format, with one Chinese character per line and hand-annotated named entity chunks in the second column. The CityU corpus was

³Most researchers refer to statistically significant as $p < 0.05$ and statistically highly significant as $p < 0.001$.

Table 3.6: Statistics of SIGHAN official NER training and testing corpora.

Corpus	Training NEs	PERs/LOCs/ORGs	Testing NEs	PERs/LOCs/ORGs
CityU	66255	16552/36213/13490	13014	4940/4847/3227
MSRA	37811	9028/18522/10261	7707	1864/3658/2185

NEs: number of named entities; PERs: number of person names;
 LOCs: number of location names; ORGs: number of organization names.

Table 3.7: OOV Rate of NER testing corpora.

Corpus	Overall (IVs/OOVs/OOV-R)	PER (IVs/OOVs/OOV-R)	LOC (IVs/OOVs/OOV-R)	ORG (IVs/OOVs/OOV-R)
CityU	6660/6354/0.4882	1062/3878/0.7850	3947/900/0.1857	1651/1576/0.4884
MSRA	6056/1651/0.2142	1300/564/0.3026	3343/315/0.0861	1413/772/0.3533

IVs: number of IV (named entities in vocabulary); OOVs: number of OOV (named entities out of vocabulary); OOV-R: ratio of named entities out of vocabulary.

traditional Chinese. We converted this corpus to simplified Chinese and we used UTF-8 encoding in all the experiments so that all the resources (e.g., word dictionary and named entity dictionary) are compatible in our system.

Table 3.6 shows the statistics of NER training and testing corpora and Table 3.7 shows the OOV (Out of Vocabulary) rate of NER testing corpora⁴. The number of NEs in CityU corpus is almost twice as many as that in MSRA corpus. The OOV rate in CityU corpus is much higher than in MSRA corpus for PERs, LOCs and ORGs. These numbers indicate that NER on CityU corpus is much more difficult to handle.

3.4.2 Features and Model Development

We use similar features and domain knowledge described in Section 3.2.2 and Section 3.2.5 for our model. For word segmentation and POS features, we train our own model for conducting Chinese word segmentation and POS tagging. We employ a unified framework to integrate cascaded Chinese word segmentation and POS tagging tasks by joint decoding that guards against violations of those hard-constraints imposed by segmentation task based on dual-layer CRFs introduced by [76]. We separately train the Chinese word segmentation and POS tagging CRF models using 8-month and 2-month PKU 2000 corpus, respectively. The original PKU 2000 corpus contains more than 100 different POS tags. To reduce the training time for POS tagging experiment, we merge some similar tags and obtain only 42 tags finally. For example, {ia, ib, id, in, iv}→i. We use the same features as

⁴The NER on the PKU corpus was cancelled by the organizer due to the tagging inconsistency of this corpus.

described in [76], except that we do not use the HowNet features for word segmentation. Instead, we use max-matching segmentation features based on a word dictionary. This dictionary contains 445456 words which are extracted from People’s Daily corpus (January-June, 1998), CityU, MSRA, and PKU word segmentation training corpora in SIGHAN-6. For decoding, we first perform individual decoding for each task. We then set 10-best segmentation and POS tagging results for reranking and joint decoding in order to find the most probable joint decodings for both tasks. For dictionary features, we obtain a named entity dictionary extracted from People’s Daily 1998 corpus and PKU 2000 corpus, which contains 68305 PERs, 28408 LOCs and 55596 ORGs. We use the max-matching algorithm to search whether a string exists in this dictionary. Besides the unigram feature template, CRFs also allow bigram feature template. With this template, a combination of the current output token and previous output token (bigram) is automatically generated.

We extend the BIO representation for the chunk tag which was employed in the CoNLL-2002 and CoNLL-2003 evaluations. We use the BIOES representation, which is more informative and yields better results than BIO representation. We performed holdout methodology to develop our model. We randomly selected 5000 sentences from CityU training corpus for development testing and the rest for training. We did the same thing for MSRA training corpus. We found an optimal value for the parameter c^5 for CRFs. Using held-out data, we tested all c values, $c \in [0.2, 2.2]$, with an incremental step of 0.4. Finally, we set $c = 1.8$ for CityU corpus and $c = 1.0$ for MSRA corpus.

3.4.3 Official Results

Table 3.8 and Table 3.9 show the top 5 systems in SIGHAN NER open track on CityU and MSRA corpus, respectively. Our results are consistently good: we obtained the first place on the CityU open track (90.33 overall F-measure) and fourth place on the MSRA open track (92.88 overall F-measure) respectively. The lower F-measure obtained on CityU corpus can be attributed to the higher OOV rate of this corpus.

⁵This parameter trades the balance between over-fitting and under-fitting. With larger c value, CRF tends to overfit to the give training corpus. The results will significantly be influenced by this parameter

Table 3.8: Top 5 systems in SIGHAN NER open track on CityU corpus

ID	R	P	F	R_{PER}	P_{PER}	F_{PER}	R_{LOC}	P_{LOC}	F_{LOC}	R_{ORG}	P_{ORG}	F_{ORG}
23	87.43	93.42	90.33	95.26	97.21	96.23	93.42	92.35	92.88	66.44	88.05	75.73
02	85.79	91.79	88.69	88.22	94.49	91.25	93.36	90.99	92.16	70.72	88.52	78.62
28	88.26	88.26	88.26	91.68	89.47	90.56	93.29	89.42	91.32	75.46	84.11	79.55
24	89.75	86.16	87.92	94.74	91.53	93.11	93.89	89.66	91.73	75.89	72.74	74.28
39	71.63	80.00	75.59	71.80	81.94	76.53	83.89	78.45	81.08	52.96	79.86	63.69

23: The Chinese University of Hong Kong (our group); 02: City University of HK; 28: State Key Laboratory of Machine Perception, Peking University; 24: France Telecom R&D Beijing, Co. Ltd; 39: Language Computer Corporation.

Table 3.9: Top 5 systems in SIGHAN NER open track on MSRA corpus

ID	R	P	F	R_{PER}	P_{PER}	F_{PER}	R_{LOC}	P_{LOC}	F_{LOC}	R_{ORG}	P_{ORG}	F_{ORG}
24	99.95	99.82	99.88	1	99.89	99.95	99.97	99.75	99.86	99.86	99.86	99.86
02	99.61	99.56	99.58	1	1	1	99.92	99.29	99.60	98.76	99.63	99.20
01	93.77	96.03	94.89	96.57	95.74	96.15	95.93	97.69	96.80	87.78	93.38	90.49
23	91.11	94.71	92.88	94.58	98.33	96.42	93.36	93.97	93.66	84.39	92.80	88.40
18	91.35	93.21	92.27	95.60	96.01	95.81	92.21	93.88	93.04	86.27	89.59	87.90

24: France Telecom R&D Beijing, Co. Ltd; 02: City University of HK; 01: Chinese Academy of Science; 23: The Chinese University of Hong Kong (our group); 18: Institute of Computational Linguistics, Peking University. **Note:** Group 24 and Group 02 obtained extremely high F-measures close to 100, because they used corpora which contain the SIGHAN official testing set, to train their models.

3.5 Conclusion and Discussion

As a well-established task, Chinese NER has been studied extensively and a number of techniques for this task have been reported in the literature. Most recently, the trend in Chinese NER is to use improved machine learning approaches, or to integrate various kinds of useful evidences, features, or resources.

[27] presented a lexicalized HMM-based approach to unifying unknown word identification and NER as a single tagging task on a sequence of known words. Although lexicalized HMMs was shown to be superior to standard HMMs, this approach has some disadvantages: it is a purely statistical model and it suffers from the problem of data sparseness. And the model fails to tag some complicated NEs (e.g., nested ORGs) correctly due to lack of domain adaptive techniques. The F-measures of LOCs and ORGs are only 87.13 and

83.60, which show that there is still a room for improving.

A method of incorporating heuristic human knowledge into a statistical model was proposed in [90]. Here Chinese NER was regarded as a probabilistic tagging problem and the heuristic human knowledge was used to reduce the searching space. However, this method assumes that POS tags are golden-standard in the training data and heuristic human knowledge is often ad hoc. These drawbacks make the method unstable and highly sensitive to POS errors; and when golden-standard POS tags are not available (this is often the case), it may degrade the performance.

[18] proposed a semi-Markov model which combines a Markovian, HMM-like extraction process and a dictionary component. This process is based on sequentially classifying segments of several adjacent words. However, this technique requires that entire segments have the same class label, while our technique does not. Moreover, compared to a large-scale dictionary, our domain knowledge is much easier to obtain.

However, all the above models treat NER as classification or sequence labeling problem. We first view and formulate Chinese NER as a *statistical relational learning* problem and propose a new framework incorporating probabilistic graphical models and first-order logic for Chinese NER which achieves state-of-the-art performance. We incorporate domain knowledge to capture the essential features of the NER task via MLNs, a unified framework for SRL which produces a set of weighted first-order clauses to predict new NE candidates. To the best of our knowledge, this is the first attempt at using MLNs for the NER problem in the NLP community. And our proposed framework can be extendable to language-independent NER, due to the simplicity of the domain knowledge we could access. Despite our promising empirical results, this two-stage framework is a simple integration of sequence labeling and logic model. Directions for future work include developing a new framework founded on better theoretically motivated models.

Table 3.3: Chinese NER by CRF model

	Precision	Recall	$F_{\beta=1}$
Character features			
PER	92.88%	79.42%	85.62
LOC	90.95%	82.88%	86.73
ORG	88.16%	83.86%	85.96
Overall	90.92%	82.07%	86.27
Character+Word			
PER	93.27%	82.99%	87.83
LOC	91.49%	85.16%	88.21
ORG	88.94%	84.79%	86.82
Overall	91.48%	84.46%	87.83
Character+Word+POS			
PER	92.17%	90.64%	91.40
LOC	90.56%	89.74%	90.15
ORG	89.15%	85.19%	87.12
Overall	90.76%	89.13%	89.94
All features			
PER	92.12%	90.57%	91.34
LOC	90.62%	89.74%	90.18
ORG	89.72%	85.44%	87.53
Overall	90.89%	89.16%	90.02

Table 3.4: Chinese NER by graphical models with logic

	Precision	Recall	$F_{\beta=1}$	RER
CRF Baseline				
PER	92.12%	90.57%	91.34	
LOC	90.62%	89.74%	90.18	
ORG	89.72%	85.44%	87.53	
Overall	90.89%	89.16%	90.02	
Graphical Models (GS Inference)				
PER	93.52%	93.32%	93.42	
LOC	93.19%	91.91%	92.55	
ORG	90.16%	90.71%	90.43	
Overall	92.70%	92.09%	92.39	23.75%
Graphical Models (ST Inference)				
PER	93.52%	93.32%	93.42	
LOC	93.19%	91.91%	92.55	
ORG	90.16%	90.71%	90.43	
Overall	92.70%	92.09%	92.39	23.75%
Graphical Models (MC-SAT Inference)				
PER	93.52%	93.32%	93.42	
LOC	93.19%	91.91%	92.55	
ORG	90.16%	90.71%	90.43	
Overall	92.70%	92.09%	92.39	23.75%
Graphical Models (MAP/MPE Inference)				
PER	92.87%	93.15%	93.01	
LOC	93.15%	91.61%	92.37	
ORG	90.56%	89.10%	89.82	
Overall	92.57%	91.58%	92.07	20.54%

Chapter 4

Bidirectional Integration of Pipeline Models

4.1 A Brief Introduction

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 which is highly ineffective and suffers from inherent inferiority such as brittle accumulation of errors, thus the overall performance is limited and upper-bounded [67, 110]. 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 [83]; the number of uncertain hypotheses is severely limited [88]; subtasks are only loosely coupled [111, 102]; or the approach is feed-forward or top-down integrated and it only allows information to flow in one direction [25]. 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 [82]. Let \mathcal{G} be a factor graph defining a probability distribution over a set of output variables 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 [47] 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.

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. And parameter estimation can be performed efficiently using evidences from multiple subtasks, such that they aid each other to boost the performance. We then propose a strong *bidirectional* algorithm based on efficient Markov chain Monte Carlo (MCMC) sampling to enable tractable inference, which allows information to flow bidirectionally and mutual benefits from different subtasks can be well exploited. 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.

4.2 Model

Let \mathbf{X} be a document containing N observation sequences: $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$, each \mathbf{X}_i consists of p tokens: $\mathbf{X}_i = \{x_{i1}, \dots, x_{ip}\}$. Let $\mathbf{S}_i = \{s_{i1}, \dots, s_{iq}\}$ 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. We describe this framework formally as follows, and the parameter optimization will be discussed in the next section.

4.2.1 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 $\Phi(\mathbf{S}^j, \mathbf{X}) = \exp\{\sum_l^L \sum_i^I \sum_k^K \lambda_k g_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{X}_l)\}$. Similar to semi-CRFs [72], the value of segment feature function $g_k(\cdot)$ depends on the current segment $s_{l,i}^j$, the previous segment $s_{l,i-1}^j$, and the whole observation \mathbf{X}_l . And transitions within a segment can be non-Markovian. The joint factor $\Phi^\nabla(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \exp\{\sum_l^L \sum_i^I \sum_k^K \mu_k r_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{R}^j, \mathbf{X}_l) + \sum_l^L \sum_i^I \sum_k^K \nu_k q_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{S}^{j-1}, \mathbf{X})\}$. 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:

$$\begin{aligned} \Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) &= \Phi(\mathbf{S}^j, \mathbf{X}) \cdot \Phi^\nabla(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) \\ &= \exp \left\{ \sum_l^L \sum_i^I \sum_k^K \lambda_k g_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{X}_l) + \sum_l^L \sum_i^I \sum_k^K \mu_k r_k \right. \\ &\quad \left. (s_{l,i-1}^j, s_{l,i}^j, \mathbf{R}^j, \mathbf{X}_l) + \sum_l^L \sum_i^I \sum_k^K \nu_k q_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{S}^{j-1}, \mathbf{X}) \right\} \end{aligned} \quad (4.1)$$

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

$$P(\mathbf{S}^j | \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) = \frac{1}{Z(\mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})} \prod \Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) \quad (4.2)$$

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.

4.2.2 Relation

In the j -th iteration, the traditional relation factor $\Phi(\mathbf{R}^j, \mathbf{X})$ in this component is written as $\exp\{\sum_{m,n}^M \sum_k^K \theta_k f_k(e_m, e_n, r_{mn}^j, \mathbf{X}) + \sum_{m,t,n}^M \sum_k^K \xi_k w_k(r_{mt}^j, r_{nt}^j, r_{mn}^j, \mathbf{X})\}$ to model relations r_{mn}^j 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 $\exp\{\sum_{m,n}^M \sum_k^K \gamma_k h_k(e_m, e_n, r_{mn}^j, \mathbf{S}^{j-1}, \mathbf{X})\}$, 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:

$$\begin{aligned} \Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) &= \Phi(\mathbf{R}^j, \mathbf{X}) \cdot \Phi^\nabla(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) \\ &= \exp \left\{ \sum_{m,n}^M \sum_k^K \theta_k f_k(e_m, e_n, r_{mn}^j, \mathbf{X}) + \sum_{m,t,n}^M \sum_k^K \xi_k w_k \right. \\ &\quad \left. (r_{mt}^j, r_{nt}^j, r_{mn}^j, \mathbf{X}) + \sum_{m,n}^M \sum_k^K \gamma_k h_k(e_m, e_n, r_{mn}^j, \mathbf{S}^{j-1}, \mathbf{X}) \right\} \end{aligned} \quad (4.3)$$

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

$$P(\mathbf{R}^j | \mathbf{X}, \mathbf{S}^{j-1}) = \frac{1}{Z(\mathbf{X}, \mathbf{S}^{j-1})} \prod \Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1}) \quad (4.4)$$

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.

4.2.3 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 overfitting, 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

$$\mathcal{L} = \log [\Phi(\mathbf{S}^j, \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})] - \log [Z(\mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})] - \sum_k \frac{\delta_k^2}{2\sigma^2} \quad (4.5)$$

To simplify the expression, let $c_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{R}^j, \mathbf{S}^{j-1}, \mathbf{X})$ 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:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \delta_k} &= \sum_l^L \sum_i^I c_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{R}^j, \mathbf{S}^{j-1}, \mathbf{X}) - \sum_l^L \sum_i^I c_k(s_{l,i-1}^j, \\ & s_{l,i}^j, \mathbf{R}^j, \mathbf{S}^{j-1}, \mathbf{X}) \times P(\mathbf{S}^j | \mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1}) - \sum_k^K \frac{\delta_k}{\sigma^2} \end{aligned} \quad (4.6)$$

Let $b_k(e_m, e_n, r_{mt}^j, r_{nt}^j, r_{mn}^j, \mathbf{S}^{j-1}, \mathbf{X})$ 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

$$\mathcal{L}' = \log [\Phi(\mathbf{R}^j, \mathbf{X}, \mathbf{S}^{j-1})] - \log [Z(\mathbf{X}, \mathbf{S}^{j-1})] - \sum_k \frac{\eta_k^2}{2\sigma^2} \quad (4.7)$$

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

follows

$$\begin{aligned} \frac{\partial \mathcal{L}'}{\partial \eta_k} = & \sum_{m,t,n}^M b_k(e_m, e_n, r_{mt}^j, r_{nt}^j, r_{mn}^j, \mathbf{S}^{j-1}, \mathbf{X}) - \sum_{m,t,n}^M b_k(e_m, e_n, \\ & r_{mt}^j, r_{nt}^j, r_{mn}^j, \mathbf{S}^{j-1}, \mathbf{X}) \times P(\mathbf{R}^j | \mathbf{X}, \mathbf{S}^{j-1}) - \sum_k^K \frac{\eta_k}{\sigma^2} \end{aligned} \quad (4.8)$$

Both of functions \mathcal{L} and \mathcal{L}' are concave, which follows from the convexity of functions of the form $f(\mathbf{x}) = \log \sum_i \exp(x_i)$. Convexity is extremely helpful for parameter estimation, because it means that every local optimum is also a global optimum. Adding regularization ensures that \mathcal{L} and \mathcal{L}' are strictly concave, which implies that they have exactly one global optimum.

Both of functions \mathcal{L} and \mathcal{L}' can therefore be efficiently maximized by standard techniques such as stochastic gradient and L-BFGS algorithms which make approximate use of second-order information. L-BFGS uses the Broyden-Fletcher-Goldfarb-Shanno update to approximate the Hessian matrix. It is particularly well suited for optimization problems with a large number of dimensions. This is because L-BFGS never explicitly forms or stores the Hessian matrix, which can be quite expensive when the number of dimensions becomes large. Instead, L-BFGS maintains a history of the past m updates of the position and the gradient, where generally the history m can be short, often less than 10. These updates are used to implicitly do operations requiring the Hessian (or its inverse).

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. If relation hypotheses are not available, it can work without such information. The relation component benefits from the segmentation component by using the segmentation and labeling results explicitly in its feature functions. If segmentation results are not available, it can also work without such information. 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. And the performance of both components can be boosted in this optimization procedure.

4.2.4 Markov chain Monte Carlo

The Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on constructing a Markov

Algorithm 1: The collaborative parameter estimation algorithm

Input: A set of training data \mathcal{D} , feature sets c_k and b_k , and iteration number \mathbb{I} .

Output: Trained segmentation component with optimized weights δ_k^* , and trained relation component with optimized weights η_k^* .

begin

Train the segmentation component defined by

$P(\mathbf{S}^0|\mathbf{X}) = \frac{1}{z(\mathbf{X})} \prod \Phi(\mathbf{S}^0, \mathbf{X})$ where

$\Phi(\mathbf{S}^0, \mathbf{X}) = \exp\{\sum_l^L \sum_i^I \sum_k^K \lambda_k g_k(s_{l,i-1}^j, s_{l,i}^j, \mathbf{X}_l)\}$.

for $j = 1, 2, \dots, \mathbb{I}$ **do**

 Compute the most likely (Viterbi) segmentation assignment

$\mathbf{S}^{(j-1)*} = \arg \max_{\mathbf{S}^{j-1}} P(\mathbf{S}^{j-1}|\mathbf{X});$

 Train the relation component to maximize $\log P(\mathbf{R}^j|\mathbf{X}, \mathbf{S}^{j-1})$ given $\mathbf{S}^{(j-1)*}$, yielding weights η_k^j ;

 Compute the most likely (Viterbi) relation assignment

$\mathbf{R}^{j*} = \arg \max_{\mathbf{R}^j} P(\mathbf{R}^j|\mathbf{X}, \mathbf{S}^{j-1});$

 Train the segmentation component to maximize

$\log P(\mathbf{S}^j|\mathbf{X}, \mathbf{R}^j, \mathbf{S}^{j-1})$ given \mathbf{R}^{j*} , yielding weights δ_k^j ;

end

return

Segmentation component with optimized weights δ_k^ defined as*

$P(\mathcal{S}^j|\mathbf{X}, \mathbf{R}^j, \mathcal{S}^{j-1}) = \frac{1}{z(\mathbf{X}, \mathbf{R}^j, \mathcal{S}^{j-1})} \prod \Phi(\mathcal{S}^j, \mathbf{X}, \mathbf{R}^j, \mathcal{S}^{j-1}),$ and

relation component with optimized weights η_k^ defined as*

$P(\mathbf{R}^j|\mathbf{X}, \mathcal{S}^{j-1}) = \frac{1}{z(\mathbf{X}, \mathcal{S}^{j-1})} \prod \Phi(\mathbf{R}^j, \mathbf{X}, \mathcal{S}^{j-1}).$

end

chain that has the desired distribution as its stationary distribution. In practice, it means that we can construct a Markov chain whose states are the objects we wish to sample. The state space \mathcal{S} includes all possible segmentations and relations of the entire dataset in our case. And the transition probabilities are specified via a scheme guaranteed to converge to the desired distribution. We can walk the Markov chain, occasionally outputting samples. These samples are likely to be in high probability areas, increasing our chances of finding the maximum. After the chain has run long enough for it to approach its stationary distribution, the expectation $E_\pi[f]$ of any function $f(S)$ of the state S will be approximated by the average of that function over the set of sample states produced by the MCMC algorithm.

4.2.5 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. This means that we can find the optimal segmentation assignments given the relation assignments in the document, and the optimal relation assignments conditioned on the segmentation assignments in the document.

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 [56][33] 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. And the transition probabilities are specified via a scheme guaranteed to converge to the desired distribution. 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 . In other words, the transition probability of the Markov chain is the conditional distribution of the attribute (boundary and label) at the segment S_i given the rest of the segmentation sequence. 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:

$$P(S^{t+1}|S^t, R^t, \mathbf{X}) = P(S_i^{t+1}|S_{-i}^t, R^t, \mathbf{X}) \quad (4.9)$$

where $S_i = (S_{i1} \cdots S_{ik})$, S_{-i} is all segments except S_i , and $S_{-i}^t = S_{-i}^{t+1}$. 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 4.9. 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

$$P(S^{t+1}|S^t, R^t, \mathbf{X}) = \frac{P((S_{i1} \cdots S_{ik})^{t+1}, S_{-i}^t, R^t, \mathbf{X})}{\sum_{y_{ij} \in \mathcal{Y}} P((S_{i1} \cdots S_{ik})^{t+1}, S_{-i}^t, R^t, \mathbf{X})} \quad (4.10)$$

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 . As described above, the (random) sampling is still inefficient, since the state space is extremely non-convex. Moreover, to perform global optimization, a more principled strategy is to adopt simulated annealing [42] in the MH algorithm, and the acceptance probability $\mathcal{A}(S^t, \hat{S})$ is written as

$$\mathcal{A}(S^t, \hat{S}) = \min \left\{ 1, \frac{P^{1/c_t}(\hat{S}|R^t, \mathbf{X})Q(S^t|\hat{S}, R^t, \mathbf{X})}{P^{1/c_t}(S^t|R^t, \mathbf{X})Q(\hat{S}|S^t, R^t, \mathbf{X})} \right\} \quad (4.11)$$

where c_t is a decreasing cooling schedule with $\lim_{t \rightarrow \infty} c_t = 0$. As $c_t \rightarrow 0$ the distribution becomes sharper, and when $c_t = 0$ the distribution places all of its mass on the maximal outcome, having the effect that the Markov chain always climbs uphill. Thus if we gradually decrease c_t from 1 to 0, the Markov chain increasingly tends to go uphill. 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 4.10, and $Q(S^t|\hat{S}, R^t, \mathbf{X})$ can also be easily computed as

$$Q(S^t|\hat{S}, R^t, \mathbf{X}) = P(S_i^t|\hat{S}_{-i}, R^t, \mathbf{X}) = \frac{P(S_i^t, \hat{S}_{-i}, R^t, \mathbf{X})}{\sum_{y_i \in \mathcal{Y}} P(S_i^t, \hat{S}_{-i}, R^t, \mathbf{X})} \quad (4.12)$$

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. 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 N , it draws samples \hat{S} from $P(S^{t+1}|R^t, \mathbf{X})$ ($0 \leq t < N$) 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. [24], 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. [25], 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.

Chapter 5

An Integrated Discriminative Probabilistic Approach

5.1 A Brief Introduction

Recently, probabilistic graphical models for sequence data have become the predominant formalism for IE, achieving the state-of-the-art performance [82]. Typically, probabilistic graphical models can deal well with uncertainty, but they are less expressive and flexible than logical or symbolic systems [80][23]. More specifically, a major disadvantage of probabilistic graphical models, is the limited expressiveness of attribute-value representation of features. Attribute-value vectors have the same level of expressiveness as propositional formalisms, that is, they only allow the representation of atomic propositions and constants. While attribute-value representation is suitable for statistical machine learning approaches, they can hardly handle IE problems involving complex knowledge which requires richer representational power facilitating logical inference or reasoning.

Another limitation is that, a unique representation for all examples is needed, resulting in a quite sparse data representation. The problem of data sparseness increases as more knowledge is exploited and this can cause problems for large scale real-world tasks. Finally, in this kind of representation, equivalent features may have to be restricted to distinct identifiers. For example, the sentence "John and Bob show Kate a picture." contains a coordinate subject, namely, John and Bob. To capture the interaction between subject and action, two features, namely `subj1-verb1`, `subj2-verb1` are required. However, each feature typically is assigned a unique identifier. They will be treated as two independent pieces of information by graphical models.

First-order logic [28], on the other hand, is a powerful paradigm to rep-

resent a wide variety of knowledge. It is a more expressive formalism and allows the representation of variables and n -ary predicates, i.e., domain and relational knowledge. It can also capture complex and implicit properties through rich expression of conditions. Therefore, dependency and deeper relations can be more adequately described. First-order formalisms allow a generic predicate to be created for every possible example, relating two or more elements [80][12]. For example, the predicate `work_for(person, company)` could have several instantiations such as `work_for(John, Microsoft)` and `work_for(Bob, IBM)`, etc. While highly expressive, this type of model lacks a sophisticated treatment of degrees of uncertainty and fuzziness, which permeates real-world domains, especially the ones usually associated with intelligence. Clearly, probabilistic graphical models and first-order logic offer complementary strengths and weaknesses for sequence data, and the integration of both is highly desirable.

However, incorporating logic into probabilistic models is generally quite challenging. This is because probabilistic graphical models operate on the token level, and they are incapable of incorporating entity-level commonly found domain knowledge. There is a fundamental mismatch in representation resulting in difficulty in the integration. We solve this problem by relaxing probabilistic graphical models with the introduction of segments, and the labels of tokens inside a segment are assumed to be the same. Given the segments in observation sequences, various kinds of relational or domain knowledge can be easily and concisely formulated into first-order logic, and logical learning can be conducted.

Inspired by this motivation, in this chapter we combine the advantages of both probabilistic graphical models for sequence data and first-order logic in a principled way, resulting in an integrated discriminative probabilistic framework which models both segmentations in sequence data and relations of different segments simultaneously for IE tasks. This integrated model offers a great flexibility to capture uncertainty for sequence modeling, as well as a variety of first-order knowledge. We illustrate the benefits of this model for mining implicit relations and new relation discovery, and capturing substructures in named entities. We propose the Metropolis-Hastings [56], [33], an approximate Markov chain Monte Carlo (MCMC) algorithm to enable efficient and tractable inference for this model. This algorithm performs efficient sampling from segmentations via Markov chains, and it is guaranteed to converge. Joint parameter estimation in this model can be too expensive or even intractable. We perform parameter estimation somewhat separately for this integrated model.

5.2 Motivating Examples

High-level information extraction requires both probabilistic modeling and complex and deeper knowledge representation. In this section, we show the shortcomings of sophisticated probabilistic approaches to sequence modeling, and illustrate the advantages of the integrated model incorporating probability with first-order logic for two real-world IE tasks.

5.2.1 Implicit Relation Extraction and New Relation Discovery

A large number of engineered systems were developed for identifying relations of interest. However, reliably extracting relations between entities in text is still a difficult and unsolved problem. Traditional probabilistic systems extract relations assuming that entities are already known or extracted from text. They rely on the assumption that entity extraction has been solved without errors. Unfortunately, such assumption is not valid in practice.

Another major limitation is that, implicit relations can hardly be discovered in these models. Implicit relations are those that do not have direct contextual evidence and generally exist in different paragraphs, or even across documents. They require additional knowledge to be detected. Notably, they are ubiquitous and are the sorts of relations that are likely to have significant impact on performance. Unfortunately, extracting implicit relations is challenging even for current state-of-the-art relation extraction models.

In particular, consider the following 4 sentences:

1. *Rosemary Forbes* is the mother of *John Kerry*.
2. *Rosemary Forbes* has a sibling *James Forbes*.
3. *James Forbes's* son is *Stuart Forbes*.

4. *John Kerry* celebrated with *Stuart Forbes* on the graduation ceremony. State-of-the-art relation extraction systems may be able to detect the *son* and *sibling* relations (as shown in Figure 5.1) from local contextual clues. However, the *cousin* relation is only implied by the text and requires additional knowledge to be extracted.

We show that our approach can enable this technology. First-order formalism allows the representation of deep and relational knowledge. By employing the logic $\text{parent}(x,z) \wedge \text{sibling}(z,w) \wedge \text{child}(w,y) \Rightarrow \text{cousin}(x,y)$, the implicit relation can be easily extracted and new relation can be discovered ultimately. Since these formulas will not always hold, we would like

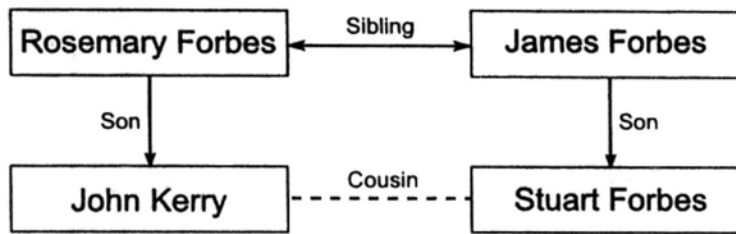


Figure 5.1: An example of implicit relation extraction. The real lines show explicit (general) relations and the dashed line shows an implicit relation.

to handle them probabilistically by estimating the confidence of each formula. Our approach can incorporate rich dependencies between entities. It can also exploit *relational autocorrelation*, a widely observed characteristic of relational data in which the value of a variable for one instance is highly correlated with the value of the same variable on another instance.

5.2.2 Modeling Sub-structures in Named Entities

Structured data are widely prevalent in the real world. Observation sequences tend to have distinct internal sub-structure and indicate predictable relationships between individual class labels. For example, in the named entity recognition task, many named entities have particular characteristics in their composition and human beings usually use prior knowledge to recognize them. A location name can optionally end with a location salient word (such as *City*, *Province*, etc.), but cannot end with any organization salient word (such as *Government*, *University*, etc.). A complex, nested organization name may be composed of a person name, a location name, or even another organization name. All entities cannot comprise any stopword or punctuation. These complex and expressive structures can largely influence predictions. For example, the organization name *U.S. Government* consists of a location name *U.S.* and an organization salient word *Government* that implies the organization class label. This characteristic is crucial, and without modeling it may lead to an incorrect location label for *U.S. Government*. However, the efficiency of probabilistic sequence models, such as the linear chain CRF-based [47] approach heavily depends on its first-order Markov property — given the observation, the label of a token is assumed to depend only on the labels of its adjacent tokens. The CRF approach models the transitions between class labels to enjoy advantages of both generative and

discriminative methods, thus capturing external dynamics, but unfortunately it lacks the ability to represent internal sub-structure.

Fortunately, these sub-structures can be modeled well by first-order logic in the integrated model. For example, the logic formulas $\text{endwith}(r,p) \wedge \text{locs_salient}(p) \Rightarrow \text{loc}(r)$ and $\text{endwith}(r,p) \wedge \text{orgs_salient}(p) \Rightarrow \text{org}(r)$ convey that if an entity candidate ends with a location salient word, then it is a location name; if it ends with an organization salient word, then it is an organization name. $\text{containstop}(p) \Rightarrow \text{non_entity}(p)$ and $\text{containpunc}(p) \Rightarrow \text{non_entity}(p)$ illustrate stopword and punctuation restrictions in named entities. These logic formulas can be designed easily and concisely to capture and model the essential sub-structures.

5.3 Model

Let \mathbf{X} be an observation sequence of tokens and $|\mathbf{X}|$ be the length of the sequence (i.e., number of tokens). Let $S = \langle S_1, S_2, \dots, S_L \rangle$ be a segmentation of the input sequence \mathbf{X} , and each entry is a segment which is a triple $S_i = \langle t_i, \mu_i, y_i \rangle$, with t_i as a start position, μ_i as an end position, and y_i as the label of this segment. $y_i \in \mathcal{Y}$ where \mathcal{Y} is the label set. Thus, a segment S_i means that the label y_i is assigned to all the observations between the start position t_i and the end position μ_i in the observation sequence \mathbf{X} . It is reasonable to assume that segments have positive lengths and adjacent segments touch, that is, $0 \leq t_i \leq \mu_i \leq |\mathbf{X}|$ and $t_{i+1} = \mu_i + 1$. S can essentially model entity candidates to be considered. Let $R = \langle R_1, R_2, \dots, R_M \rangle$ be a first-order logic possible world of segment relations expressed as a set of ground predicates R_i with truth value assigned. When only one segment candidate appears in the arguments of R_i , it represents a particular segment constraint (e.g., sub-structure in its composition). When more than one segment candidate appears in the arguments of R_i , it represents relations of segments.

We now describe in detail our proposed model. We jointly consider segmentations S in observation sequence \mathbf{X} and possible worlds of segment relations R . Therefore, an assignment of all the variables in the integrated model is a pair $\langle R, S \rangle$. A valid assignment $\langle R, S \rangle$ must satisfy the condition that both of the two assignments are optimized, that is, the assignments of the segments and the assignments of the relations of segments are maximized simultaneously. Let \hat{R} and \hat{S} denote the most likely relation assignment and segmentation assignment, respectively. By applying chain rule, \hat{R} and \hat{S} can

be obtained as follows:

$$\begin{aligned} \langle \hat{R}, \hat{S} \rangle &= \arg \max_{R,S} p(\langle R, S \rangle | \mathbf{X}) \\ &= \arg \max_{R,S} p(R|S, \mathbf{X}) \cdot p(S|\mathbf{X}) \end{aligned} \quad (5.1)$$

Clearly, our model consists of two sub-structures — segmentations S conditioned on observation sequence \mathbf{X} , represented by $p(S|\mathbf{X})$, and relations R of segments given a segmentation S and observation sequence \mathbf{X} , represented by $p(R|S, \mathbf{X})$. Note that this model is quite general, and has potential to integrate a variety of probabilistic segmentation and logic models.

In particular, we investigate the use of undirected, discriminatively-trained probabilistic graphical models, known as Semi-Markov conditional random fields (Semi-CRFs) [72], to effectively model segmentations S in sequence data. Besides the modeling flexibility similar to conventional CRFs, Semi-CRFs are capable of measuring properties of segments, and transitions within a segment can be non-Markovian. For segment relations, we employ the idea of Markov logic networks (MLNs) [69], a recently introduced framework for first-order logic, to model relations R of segments. An MLN is a set of first-order knowledge base (KB) with a real-valued weight assigned to each formula. Together with a finite set of constants representing objects in the domain, it defines a ground Markov network containing one feature for each possible grounding of a first-order formula in the KB, with the corresponding weight. The weights associated with the formulas in an MLN jointly determine the probabilities of those formulas (and vice versa) via a log-linear model.

The KB is a set of general constraints $C = \{C_1, C_2, \dots, C_N\}$ expressed as first-order logic formulas. Each C_i contains some predicates representing constraints on elements in the domain. When only one segment variable appears in C_i , it represents certain attribute or characteristic of that type of segment. When more than one segment variable appears in C_i , it represents certain relations among segments. Some C_i may not contain segment variable modeling relations or characteristics of non-segment objects in the domain. Formulas are constructed using constants, variables, functions, and predicates. *Constant* symbols represent objects in the domain of interest (e.g., people such as **John** and **Bob**). *Variable* symbols range over the objects in the domain (e.g., x and y). *Function* symbols represent mappings from tuples of objects to objects and predicate symbols represent relations among objects (e.g., **person**) in the domain or attributes of objects (e.g., **endwith**). Variables and constants are *typed*, in which case variables range only over objects of the corresponding type, and constants can only represent objects

of the corresponding type, to reduce the size of ground Markov network. For example, the variable x may range over people (e.g., John, Bob, etc.) and the constant `Microsoft` may represent a company. Formulas are recursively constructed from atomic formulas (predicates applied to a tuple of terms) using logical connectives and quantifiers. The formulas in a KB are implicitly conjoined. A *ground atom* is an atomic formula all of whose arguments are *ground terms* (terms containing no variables).

Markov logic is a highly expressive language to specify the connectivity and template of a Markov network. The nodes in the network structure of an MLN are atomic formulas, and the edges are the logical connectives used to construct the formula. Each formula is considered to be a clique, and the Markov blanket is the set of formulas in which a given atom appears. However, atomic formulas do not have a truth value until they are ground atoms with a Herbrand interpretation. Thus, an MLN becomes a Markov network only with respect to a specific grounding and interpretation, and the resulting Markov network is called the ground Markov network. Given different sets of constants, it will produce different networks. In the graphical structure of a ground Markov network, the nodes are ground atoms. There is an edge between two nodes iff the corresponding ground atoms appear together in at least one grounding of one formula in the KB. The atoms in each ground formula form a clique in the ground Markov network. Each state of the ground Markov network represents a possible world. Under several reasonable assumptions [69], the set of possible worlds is finite, and the ground Markov network represents a unique, well-defined probability distribution over those worlds, irrespective of the interpretation and domain.

The segments from the Semi-CRF sub-structure are considered to be entity candidates. Given these entity candidates and the first-order logic KB, the ground Markov network can be constructed to learn the relations between them. The MLN sub-structure attempts to provide logical inference on the entity candidates. It is constructed via grounding the first-order logical formulas associated with entity candidates. It also consists of a graphical structure derived from the formulas instantiated with the data and the weights. Consequently, the entities can be identified and the entity relations can be learned from the integrated model.

Let \mathbf{B} be the ground predicates generated from the input sequence \mathbf{X} . In other words, \mathbf{B} contains atomic formulas whose arguments are not variables. Given a particular segmentation S , *Evidence* predicates are a set of ground atoms with known truth values. Take the NER task for example, for the first-order KB listed in Table 7.1, some predicates such as `per`, `loc`, `org`, and `non_entity` are entity or query predicates. These predicates are used to predict whether an entity candidate is a PER, a LOC, an ORG, or

a non-entity. The remaining predicates such as `endwith(r,p)` are *Evidence* predicates. $p(R|S, \mathbf{X})$ in Equation 5.1 can be computed efficiently by calculating $p(R|Evidence, \mathbf{B}, S)$. Therefore, Equation 5.1 can be rewritten as:

$$\begin{aligned} \langle \hat{R}, \hat{S} \rangle &= \arg \max_{R,S} p(R|Evidence, \mathbf{B}, S) \cdot p(S|\mathbf{X}) \end{aligned} \quad (5.2)$$

$$= \arg \max_{R,S} \left(\frac{p(R, Evidence|\mathbf{B}, S)}{p(Evidence|\mathbf{B}, S)} \cdot p(S|\mathbf{X}) \right) \quad (5.3)$$

$$= \arg \max_{R,S} p(R, Evidence|\mathbf{B}, S) \cdot p(S|\mathbf{X}) \quad (5.4)$$

$$= \arg \max_{R,S} p(W_R|\mathbf{B}, S) \cdot p(S|\mathbf{X}) \quad (5.5)$$

where W_R is a set of segments, a set of functions, and a set of relations of segments; together with an interpretation, they determine the truth value of each ground atom. Equation 5.3 can be inferred to Equation 5.4, due to the fact that for a constructed ground Markov network, $p(Evidence|\mathbf{B}, S)$ is constant.

As described above, the model consists of two types of sub-structures: (1) a semi-Markov chain on the segmentations S conditioned on the sequences \mathbf{X} ; (2) an undirected graph constructed via grounding the first-order KB associated with segments (entity candidates). As shown in Figure 5.2, the nodes in this graph are ground atoms with a possible world or Herbrand interpretation assigning a truth value to each node. For the node $F_1(A, C)$, A and C are segments from observation sequence \mathbf{X} and F_1 is a predicate. Therefore, our model can be formally derived as follows:

$$\begin{aligned} \langle \hat{R}, \hat{S} \rangle &= \arg \max_{R,S} \frac{1}{Z_S} \exp \left(\sum_{i=1:|S|} \lambda_i g(i, \mathbf{X}, S) \right) \cdot \frac{1}{Z_W} \exp \left(\sum \theta_i n_i(W_R) \right) \\ &= \arg \max_{R,S} \frac{1}{Z_S} \left(\prod_{i=1:|S|} \psi_i(i, \mathbf{X}, S) \right) \cdot \frac{1}{Z_W} \left(\prod \phi_i(W_{R\{i\}})^{n_i(W_R)} \right) \end{aligned} \quad (5.6)$$

where $g = \langle g^1, g^2, \dots, g^K \rangle$ is a vector of segment feature functions. Each g^k depends on the current segment, the whole observation, and the label of previous segment, that is, $g^k(i, \mathbf{X}, S) = g^k(y_{i-1}, y_i, t_i, \mu_i, \mathbf{X})$. $\psi_i(i, \mathbf{X}, S) = \exp(\lambda_i g(i, \mathbf{X}, S))$ is the potential function conditioned on features of \mathbf{X} for segmentations. $n_i(W_R)$ is the number of true groundings of a formula in the i -th first-order logic formula, $\phi_i(W_{R\{i\}}) = e^{\theta_i}$ is the potential function for

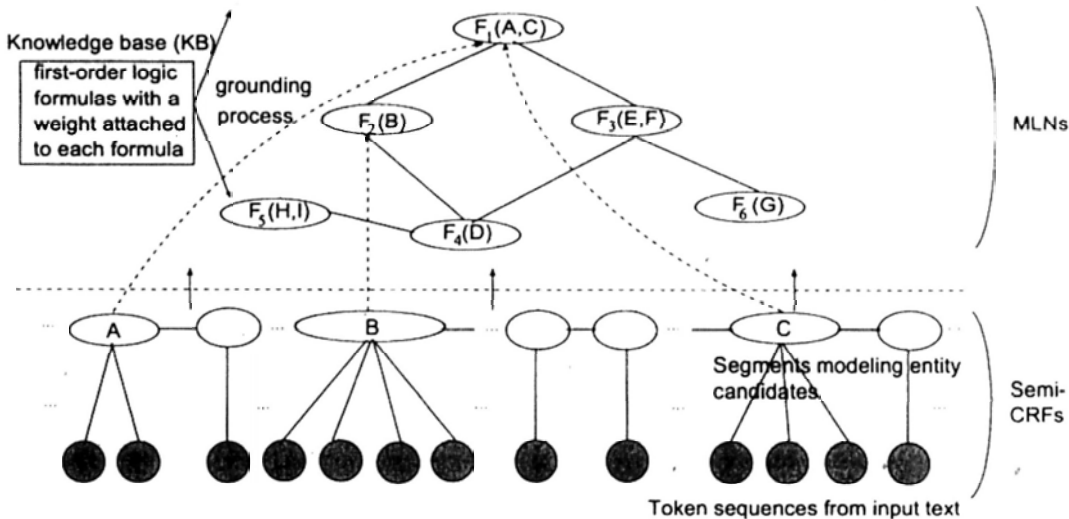


Figure 5.2: Graphical representation of the integrated discriminative probabilistic model consisting of two types of repeated sub-structures: (1) a semi-Markov chain on the segmentations conditioned on the observation sequences, the segments from this structure are considered to be entity candidates; (2) given these entity candidates and the first-order logic KB, an undirected graph is constructed via grounding process for learning relations. The nodes in this graph are ground atoms with a possible world or Herbrand interpretation.

the i -th logical formula. A potential function is associated to each formula, and takes the value of 1 when the formula is true, and 0 when it is false. $W_{R\{i\}}$ is the truth value of the grounded predicate appearing in the formulas. $\lambda = \langle \lambda_1, \lambda_2, \dots, \lambda_K \rangle$ and $\theta = \langle \theta_1, \theta_2, \dots, \theta_L \rangle$ are parameter vectors of the two sub-structures respectively. Z_S and Z_W are normalization factors of the Semi-CRF and MLN sub-structures respectively. As can be seen, this model offers a sound theoretical foundation for uncertainty, and has the advantage of combining the expressiveness of first-order logic.

5.4 Inference and Training

We discuss in detail the inference and training algorithms for our model in this section. We propose the Metropolis-Hastings (MH) algorithm, which consists of efficient sampling from segmentations, to enable approximate and

tractable inference for this model. Joint parameter estimation in this model is prohibitively intractable, and we perform parameter estimation separately for each of the two sub-structures.

5.4.1 Inference

The objective of inference is to find the most likely assignments of variables in the integrated model, that is, the pair $\langle \hat{R}, \hat{S} \rangle$ that has the maximum posterior probability. Unfortunately, exact inference by computing the posterior probability of all possible segmentation assignments S and world of relation assignments R is generally intractable, as evaluating the normalization factors Z_S and Z_W for this distribution requires summing over all possible segmentations for the entire dataset and evaluating all possible world of relations.

Consequently, approximate inference becomes an alternative by relaxing the requirement (e.g., computing the distribution explicitly) of exact inference. We propose the Metropolis-Hastings algorithm, a specific Markov chain Monte Carlo (MCMC) approach in which a Markov chain is used to sample from the segmentations $(p(S|\mathbf{X}))$ (as shown in Figure 5.3), to perform the maximum a posteriori (MAP) estimates for the inference of this model. Since MCMC and the bidirectional MH algorithm are discussed in Section 4.2.4 and Section 4.2.5, here we only focus on how to sample segmentation sequences efficiently from the conditional distribution defined by the Semi-CRF sub-structure. This algorithm runs similar procedure to the bidirectional MH algorithm, and the difference is that this algorithm does not iteratively sample from both segmentations and relations, it only draws samples from segmentations.

We adopt similar notations and formulations in Section 4.2.5. Note that the length of all segments at state $S^{(t+1)}$ should not exceed the upper bound L . From Equation 4.9 we know that the segment S_i may be divided into several sub-segments, and this may change the boundary (length) of adjacent segments S_{i-1} and S_{i+1} . The number of possible sub-segments can be large for long segment S_i . Instead of exhaustively enumerating the list of all possible sub-segments, we restrict our targets to limited ($k \leq 3$) sub-segments to enable efficient computation, based on the assumption that for a long segment, it is more likely that this segment be separated into a small number of sub-segments than into a large number of sub-segments. Note that it is possible that the total number of segments in $S^{(t+1)}$ is less than that of $S^{(t)}$ if $S_{i1}^{(t+1)}$ or $S_{ik}^{(t+1)}$ is merged with the neighboring segments in $S^{(t)}$. We can walk the Markov chain to loop through segment S_i from $i = 1$ to $i = L$, and the attribute (boundary and label) of every segment can be changed

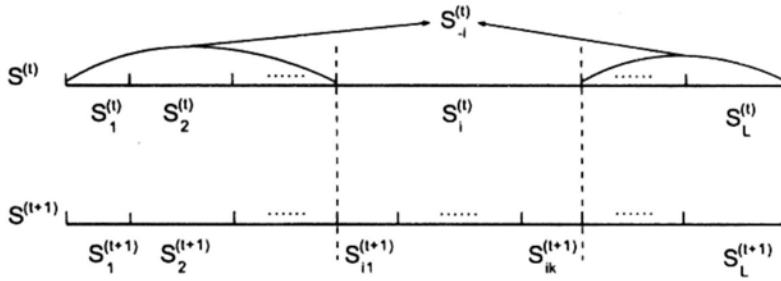


Figure 5.3: The transition probability of the Markov chain from state $S^{(t)}$ to state $S^{(t+1)}$ is the conditional distribution of the attribute (boundary and label) at the segment $S_i^{(t)}$ ($1 \leq i \leq L$) given the rest of the segmentation sequence $S_{-i}^{(t)}$. $S_i^{(t)}$ is divided into k sub-segments $(S_{i1}S_{i2} \cdots S_{ik})^{(t+1)}$ in the state $S^{(t+1)}$.

dynamically. And for each one, we re-sample the state at segment S_i from distribution given in Equation 4.9 mentioned above.

For each sample drawn from $p(S|\mathbf{X})$, the most probable relation assignment R that maximizes the probability $p(R|S, \mathbf{X})$ can be efficiently computed by the MC-SAT algorithm [66]. Thus, this approximated procedure will maximize the joint probability $p(\langle R, S \rangle | \mathbf{X})$ of the integrated model. After many samples are drawn, it will converge to $\langle \hat{R}, \hat{S} \rangle$, that is, the most likely segmentation assignment \hat{S} and corresponding most likely relation assignment \hat{R} can be obtained.

5.4.2 Parameter Estimation

Given annotated data, training can be performed to estimate the parameters in the integrated model. Each training sample in $D = \{(y^i, x^i)\}_{i=0}^N$ is a pair $(\langle R, S \rangle^i, x^i)$ and the log-likelihood function is

$$\mathcal{L}(\lambda, \theta) = \sum_{i=0}^N \log p(\langle R, S \rangle^i | x^i, \lambda, \theta), \quad (5.7)$$

where λ and θ are parameter vectors of the two sub-structures, respectively. Substituting the distribution in Equation 5.1 into the log-likelihood and we get

$$\mathcal{L}(\lambda, \theta) = \sum_{i=0}^N \log p(R^i | x^i, S^i, \lambda, \theta) + \sum_{i=0}^N \log p(S^i | x^i, \lambda, \theta) \quad (5.8)$$

Ideally, we would perform parameter estimation by numerically climbing the gradient of the full, joint likelihood. However, optimizing parameters λ and θ simultaneously is too expensive or even intractable for large-scale IE problems. Previous research indicated that learning parameters by maximizing a product of local marginals provides equal or superior accuracy to stochastic gradient ascent on an approximation of the full joint likelihood [55], [88], [111].

Following this idea, we assume that λ and θ are independent. Therefore we train each sub-structure of the model separately, either as structured pseudo-likelihood, or simply independently, and existing algorithms are sufficient for the training procedure. In all cases, estimation is iterative, consisting of dynamic programming algorithms to maximize the log-likelihood of the correct segmentation sequence [72] and limited-memory BFGS on optimizing the pseudo-log-likelihood of relations of segments [69]. To avoid over-fitting, the Gaussian prior with mean $\mu = 0$ and variance matrix $\Sigma = \delta^2 I$ is used to penalize the log-likelihood (or pseudo-likelihood) when training each part of the model.

Chapter 6

Joint Models Incorporating Logic

6.1 A Brief Introduction

To address problems occurred in pipeline approaches for higher-level IE tasks, there has been increasing interest in integrated and joint probabilistic models to explore mutual benefits and perform multiple tasks simultaneously, showing feasibility and promise in information extraction [9][88][111][102][109][67][93][50].

While several previous researchers have taken steps in this direction, they have various disadvantages: the number of uncertain hypotheses is severely limited for the full distribution of large output space [88], the subtasks are only loosely coupled [67], or the approach is feed-forward or top-down integrated and it only allows information to flow in one direction [111][102]. Exploring bi-directional information is intuitively appealing, especially during the inference procedure [93]. For example, correct coreference of a messy citation with a clean citation provides the opportunity to help the model correctly segment the messy one, and entity resolution can also benefit from correctly segmented citations.

Inspired by this motivation and to address the above problems, we propose a *joint* information extraction paradigm based on discriminatively-trained, undirected probabilistic graphical models for all relevant subtasks simultaneously. This framework models both segmentations in sequence data and relations of different segments jointly, allowing tight interactions between segmentations and relations of segments, and it combines the advantages of both probabilistic graphical models for sequence data and first-order logic in a principled way. More specifically, we make the following major contributions in this chapter:

1. First, we incorporate the advantages of both probabilistic sequence models and first-order logic, offering a great flexibility to capture uncertainty for sequence modeling, as well as a variety of domain knowledge which can be concisely and easily formulated by first-order logic. This paradigm offers a natural way for information extraction which requires uncertainty modeling as well as dependency and deeper knowledge representation.
2. Second, we propose a discriminative framework defining a joint probability distribution for both segmentations in sequence data and possible worlds of segment relations in the form of an exponential family. This joint model has several advantages over previous probabilistic graphical models.
3. Third, since exact parameter estimation in this model can be too expensive or even intractable, we propose a structured variational inference algorithm [73][89] to conduct approximate learning for the model's parameters. The variational inference method provides a fast, deterministic approximation to otherwise unattainable posteriors. Also its convergence time is independent of dimensionality [87]. Moreover, we propose a highly-coupled, bi-directional Metropolis-Hastings (MH) sampling algorithm [56][33] to enable efficient and tractable inference for this model, which allows information to flow in both directions and explores mutual benefits.

Our earlier work includes the cascaded approach (chapter 3) and the integrated discriminative approach (chapter 5). In chapter 3, we combined entity extraction and qualitative domain knowledge in a two-stage pipeline model: the first stage is a base model and the second stage is used to validate and correct the errors made in the base model. In [97], we applied Markov logic networks (MLNs) [69] for relation extraction, assuming that golden-standard entities are already known. In chapter 5, we integrated semi-CRFs [72] and MLNs for information extraction. However, this model is only loosely-coupled in a "top-down" architecture, since the parameter estimation is performed independently for the two components. For inference, the information can only flow in one direction. This chapter proposes a joint paradigm for IE and it is a major extension of the integrated model in chapter 5 in that parameters for all subtasks are optimized simultaneously via structured variational approximation to capture deep interactions between different subtasks. Moreover, the inference is strongly bi-directional, thus information can flow in both directions to exploit mutual benefits.

6.2 A Joint Model

6.2.1 Preliminaries and Notations

Let \mathbf{X} be an observation sequence of tokens and $|\mathbf{X}|$ be the length of the sequence (i.e., number of tokens). Let $S = \langle S_1, S_2, \dots, S_L \rangle$ be a segmentation of the input sequence \mathbf{X} , and each entry is a segment which is a triple $S_i = \langle t_i, \mu_i, y_i \rangle$, with t_i as a start position, μ_i as an end position, and y_i as the label of this segment. $y_i \in \mathcal{Y}$ where \mathcal{Y} is the label set. Thus, a segment S_i means that the label y_i is assigned to all the observations between the start position t_i and the end position μ_i in the observation sequence \mathbf{X} . It is reasonable to assume that segments have positive lengths and adjacent segments touch, that is, $0 \leq t_i \leq \mu_i \leq |\mathbf{X}|$ and $t_{i+1} = \mu_i + 1$. A segment S_i can be a non-entity (e.g., a punctuation), it can also be an entity (e.g., the person name *Nancy Hanks* in Figure 1.1). S can essentially model entity candidates to be considered, since they are not the final entity output in our model. We will consider possible worlds of relations between them, as described below.

Our model allows the user to specify or construct a set of first-order logic formulas [28] known as a knowledge base (\mathbf{KB}). More specifically, the \mathbf{KB} in our model is a set of general constraints $C = \{C_1, C_2, \dots, C_N\}$ expressed as standard first-order logic formulas. Each C_i contains some predicates representing constraints on elements in the domain. When only one segment variable appears in C_i , it represents certain attribute or characteristic of that type of segment. When more than one segment variable appears in C_i , it represents certain relations among segments. Some C_i may not contain segment variable modeling relations or characteristics of non-segment objects in the domain. Formulas are constructed using constants, variables, functions, and predicates. Constant symbols represent objects in the domain of interest (e.g., people such as *John* and *Bob*). Variable symbols range over the objects in the domain (e.g., x and y). Function symbols represent mappings from tuples of objects to objects and predicate symbols represent relations among objects (e.g., *person*) in the domain or attributes of objects (e.g., *endwith*). Variables and constants are *typed*, in which case variables range only over objects of the corresponding type, and constants can only represent objects of the corresponding type, to reduce the size of ground Markov network. For example, the variable x may range over people (e.g., *John*, *Bob*, etc.) and the constant *Microsoft* may represent a company. Formulas are recursively constructed from atomic formulas (predicates applied to a tuple of terms) using logical connectives and quantifiers. The formulas in a \mathbf{KB} are implicitly conjoined. Some sample formulas are given in Table 7.4.

A *ground atom* (or *ground predicate*) is an atomic formula all of whose arguments are *ground terms* (terms containing no variables). A *possible world* (or a *Herbrand interpretation*) determines a truth value assignment to each ground predicate. Let $R = \langle R_1, R_2, \dots, R_M \rangle$ be a first-order logic possible world of segment relations expressed as a set of ground predicates R_i with truth value assigned. R allows a variety of relations and dependencies, and it is built upon the segmentation S which models entity candidates. When only one segment candidate appears in the arguments of R_i , it represents a particular segment constraint (e.g., sub-structure in its composition). When more than one segment candidate appears in the arguments of R_i , it represents relations of segments.

Let $Y = \{R, S\}$ be the pair of segmentations S and possible worlds in first-order logic of segment relations R for an observation sequence \mathbf{X} . Therefore, an assignment of all the variables is a pair Y . A valid assignment Y must satisfy the condition that both of the two assignments are optimized, that is, the assignments of the segments and the assignments of the relations of segments are maximized simultaneously. We formally define the task of joint information extraction as follows:

Definition 4 (*Joint Optimization of Information Extraction*): *Given an observation sequence \mathbf{X} , the goal of joint information extraction is to find the assignment $Y^* = \{R^*, S^*\}$ that has the maximum a posteriori (MAP) probability*

$$Y^* = \arg \max_Y P(Y|\mathbf{X}), \quad (6.1)$$

where R^* and S^* denote the most likely relation assignment and segmentation assignment, respectively.

Note that this definition is different from traditional two-stage pipeline models performing segmentation and relation in sequential order, that is, optimizing $P(S|\mathbf{X})$ and $P(R|S)$ independently without capturing interactions between them.

6.2.2 Model Formulation

We now describe in detail our proposed model. We define a joint conditional distribution for segmentations S in observation sequence \mathbf{X} and possible worlds of segment relations R in undirected, probabilistic graphical models (also known as Markov random fields or Markov networks). Let \mathcal{G} be a factor graph [45] defining a probability distribution over a set of output variables \mathbf{Y} conditioned on observation sequences \mathbf{X} . A factor φ_i computes a scalar value over the subset of variables \mathbf{Y}_i and \mathbf{X}_i that are neighbors

of φ_i in the graph \mathcal{G} . Usually this real-valued function is defined as the exponential family of an inner product over sufficient statistics $\{f_{ik}(\mathbf{X}_i, \mathbf{Y}_i)\}$ and corresponding parameters $\{\mu_{ik}\}$ as $\varphi_i = \exp\{\sum_k \mu_{ik} f_{ik}(\mathbf{X}_i, \mathbf{Y}_i)\}$. Let $Z(\mathbf{X})$ be the normalization function, then the probability distribution over \mathcal{G} can be written as:

$$P(Y|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_{\varphi_i \in \mathcal{G}} \exp \left\{ \sum_k \mu_{ik} f_{ik}(\mathbf{X}_i, \mathbf{Y}_i) \right\}. \quad (6.2)$$

For factor graphs we can group several factors of similar nature using the same parameter and this is called *parameter tying*. The nature of our modeling enables us to partition the factors of \mathcal{G} into two sets of factors $\mathcal{C}_s = \{\Phi_c(\mathbf{S}_c, \mathbf{X}_c)\}$ and $\mathcal{C}_r = \{\Psi_d(\mathbf{R}_d, \mathbf{S}_d, \mathbf{X}_d)\}$, \mathbf{X}_c is a set of input variables and \mathbf{S}_c is a set of output variables, and they are arguments to the non-negative potential functions Φ_c . Similarly, \mathbf{X}_d and \mathbf{S}_d are sets of input variables, \mathbf{R}_d is a set of output variables. These are arguments to Ψ_d . In other words, \mathcal{C}_s is a collection of cliques, and \mathbf{X}_c and \mathbf{S}_c are sets of variables corresponding to the nodes in the clique c . According to the celebrated Hammersley-Clifford theorem [5], the joint conditional distribution P is factorized as a product of potential functions over cliques in the graph \mathcal{G} as:

$$P(Y|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_{c \in \mathcal{C}_s} \Phi_c(\mathbf{S}_c, \mathbf{X}_c) \prod_{d \in \mathcal{C}_r} \Psi_d(\mathbf{R}_d, \mathbf{S}_d, \mathbf{X}_d), \quad (6.3)$$

where $Z(\mathbf{X}) = \sum_Y \prod_{c \in \mathcal{C}_s} \Phi_c(\mathbf{S}_c, \mathbf{X}_c) \prod_{d \in \mathcal{C}_r} \Psi_d(\mathbf{R}_d, \mathbf{S}_d, \mathbf{X}_d)$ is the normalization factor over all states for the observation sequence \mathbf{X} .

We assume the potential functions factorize according to a set of features and a corresponding set of real-valued weights. More specifically, the potential $\Phi_c(\mathbf{S}_c, \mathbf{X}_c)$ factorizes over a set of features $g(i, \mathbf{X}_c, \mathbf{S}_c)$ and weight vector $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_K\}$, one weight for each feature, as:

$$\Phi_c(\mathbf{S}_c, \mathbf{X}_c) = \exp \left\{ \sum_{i=1}^{|\mathbf{S}_c|} \lambda g(i, \mathbf{X}_c, \mathbf{S}_c) \right\}, \quad (6.4)$$

where $g = \{g^1, g^2, \dots, g^K\}$ is a vector of segment feature functions. To effectively capture properties of segmentations S in sequence data and inspired by the idea of semi-CRFs [72], we relax the first-order Markov assumption to semi-Markov such that each g^k depends on the current segment, the whole observation, and the label of previous segment, that is, $g^k(i, \mathbf{X}_c, \mathbf{S}_c) = g^k(y_{i-1}, y_i, t_i, \mu_i, \mathbf{X}_c)$. Also transitions within a segment can be non-Markovian. $\Phi_c(\mathbf{S}_c, \mathbf{X}_c)$ is the potential conditioned on features for

segmentations \mathbf{S}_c , and it defines a semi-Markov chain over the input sequence \mathbf{X}_c .

The potential $\Psi_d(\mathbf{R}_d, \mathbf{S}_d, \mathbf{X}_d)$ factorizes according to the number of true groundings of a formula in the j -th first-order logic formula, $f_j(W_{\mathbf{R}_d})$, and the corresponding weight vector $\theta = \{\theta_1, \theta_2, \dots, \theta_L\}$ as:

$$\Psi_d(\mathbf{R}_d, \mathbf{S}_d, \mathbf{X}_d) = \exp \left\{ \sum_j \theta_j f_j(W_{\mathbf{R}_d}) \right\}, \quad (6.5)$$

$W_{\mathbf{R}_d}$ determines possible worlds of segment relations \mathbf{R}_d . And it is a set of segments, a set of functions, and a set of relations of segments; together with an interpretation. They determine the truth value assignment of each possible ground predicate of segment relations. The network contains one feature for each possible grounding of each formula, and the feature takes the value of 1 when the ground formula is true, and 0 when it is false.

Note that this formulation shares some resemblance with Markov logic networks (MLNs) [69], which can be used for constructing the graphical model for the potential Ψ_d and segment relations \mathbf{R}_d . Given a finite set of constants representing objects in the domain, we can construct a Markov network via specific *grounding process* and interpretation, and the resulting Markov network is called the *ground Markov network*. A ground Markov network contains one feature for each possible grounding of a first-order formula in the \mathbf{KB} , with the corresponding weight. The atoms in each ground formula form a clique in the ground Markov network. Each state of the ground Markov network represents a possible world. Here we make several assumptions: the set of possible worlds is finite, and the ground Markov network represents a unique, well-defined probability distribution over those worlds, irrespective of the interpretation and domain. These assumptions are reasonable for most problems, and significantly simplify the model complexity.

Now we can formally define the proposed joint discriminative probabilistic model as follows:

Definition 5 (A Joint Model for Information Extraction): *Given observation sequences \mathbf{X} and a set of first-order logic knowledge base (\mathbf{KB}). Let $\{g(i, \mathbf{X}, S)\}$ be a set of feature functions and $\{f(W_R)\}$ be the number of true groundings in the \mathbf{KB} , and $\Theta = \{\lambda_1, \lambda_2, \dots, \lambda_K, \theta_1, \theta_2, \dots, \theta_L\} \in \mathfrak{R}^{K,L}$ be a set of real-valued weights. Then the joint conditional distribution P for segmentations S and possible worlds of segment relations R is in the form of*

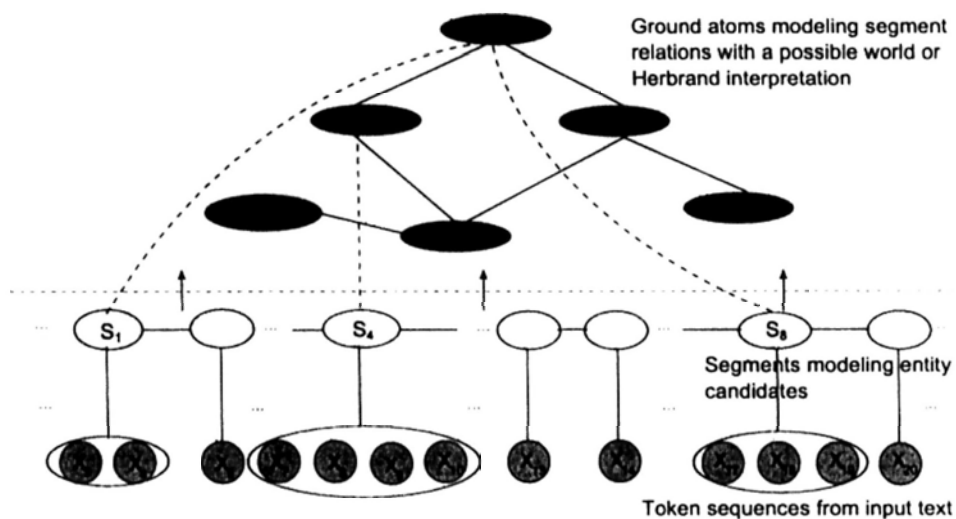


Figure 6.1: An instance of graphical representation of the joint discriminative probabilistic model for segmentations in observation sequence and possible worlds of segment relations. It consists of a semi-Markov chain on the segmentations S conditioned on the observation sequences \mathbf{X} , and a ground Markov network constructed via grounding the first-order logic \mathbf{KB} associated with segments (entity candidates).

an exponential family if and only if

$$\begin{aligned}
 P(Y|\mathbf{X}) &= \frac{1}{Z(\mathbf{X})} \prod_{c \in \mathcal{C}_s} \exp \left\{ \sum_{i=1}^{|\mathcal{S}|} \lambda g(i, \mathbf{X}, S) \right\} \prod_{d \in \mathcal{C}_r} \exp \left\{ \sum_j \theta_j f_j(W_R) \right\} \\
 &= \frac{1}{Z(\mathbf{X})} \exp \left[\sum_{i=1}^{|\mathcal{S}|} \lambda g(y_{i-1}, y_i, t_i, \mu_i, \mathbf{X}) + \sum_j \theta_j f_j(W_R) \right] \quad (6.6)
 \end{aligned}$$

where $Z(\mathbf{X}) = \sum_Y \exp \left[\sum_{i=1}^{|\mathcal{S}|} \lambda g(y_{i-1}, y_i, t_i, \mu_i, \mathbf{X}) + \sum_j \theta_j f_j(W_R) \right]$ is the normalization factor of the joint model.

An instance of the graphical representation of the joint model is shown in Figure 6.1. The gray nodes represent token sequences from input observations \mathbf{X} . The yellow nodes (e.g., the node S_1 , S_4 , and S_8) express segments S_i which model entity candidates to be considered. Given the first-order logic \mathbf{KB} , the red (darkest) nodes in this graph are ground atoms with a possible world or Herbrand interpretation assigning a truth value to each node. For example, consider the node $R_1(S_1, S_8)$, S_1 and S_8 are segments

from observation sequence \mathbf{X} and R_1 is a predicate. It represents certain relation between segment S_1 and S_8 . It is also possible that some nodes contain only one segment (e.g., the node $R_2(S_4)$). It means that there is no relation between this segment and other segments.

As mentioned above, the formulation of our model shares some similarity with semi-CRFs [72] and MLNs [69] in that we use semi-Markov chains to model segmentations and first-order logic \mathbf{KB} to represent possible worlds of segment relations. However, several major elements make our model different. The semi-CRF model cannot capture long-distance dependencies, nor expressive knowledge representation. The MLN model can conduct relation learning between entities, however, this model cannot be applied to token-level learning since all entities are unknown in our task. As can be seen, our model offers a sound theoretical foundation for uncertainty, and has the advantage of combining the expressiveness of first-order logic. By modeling both segmentations in sequence data and relations of segments simultaneously, our proposed model offers a natural way for joint information extraction, avoiding the problems such as error propagation occurred in decoupled approaches. Using first-order logic formalism, this model can capture a rich class of relations and dependencies (e.g., long-distance dependencies). Using tractable substructures in the structured variational inference approximation, deep interactions between entities and relations can be captured. Moreover, the bi-directional MCMC inference allows information to flow in both directions, and makes use of mutual benefits from different tasks to boost the performance.

6.2.3 Exact Parameter Estimation

Given a set of training data $\mathcal{D} = \{(\mathbf{X}^l, Y_o^l)\}_{l=0}^N$ and a first-order logic \mathbf{KB} containing k formulas, where \mathbf{X}^l is the l -th sample, and Y_o^l is the corresponding label and $Y_o^l = \{R_o^l, S_o^l\}$, the objective of learning is to estimate $\Theta = \{\lambda_1, \lambda_2, \dots, \lambda_K, \theta_1, \theta_2, \dots, \theta_L\}$ which is the vector of model's parameters. Without loss of generality, R_o^l and S_o^l are observed variables (labels), while R and S are viewed as hidden variables. Assume that the samples are independent and identically distributed (IID) and the log-likelihood of the data is:

$$\begin{aligned} \mathcal{L}(\Theta) &= \sum_{l=0:N} \log P(Y_o^l | \mathbf{X}^l) = \sum_{l=0:N} \log \left[\sum_Y P(Y, Y_o^l | \mathbf{X}^l) \right] \\ &= \sum_{l=0:N} \log \left[\sum_{S,R} P(S, R, S_o^l, R_o^l | \mathbf{X}^l) \right]. \end{aligned} \quad (6.7)$$

To reduce over-fitting, the Gaussian prior with mean $\mu = 0$ and variance matrix $\Sigma = \sigma^2 I$ is used to penalize the log-likelihood (or pseudo-log-likelihood). After substituting in the joint model (Equation 6.6) into the log-likelihood (Equation 6.7), we obtain the following expression:

$$\begin{aligned} \mathcal{L}(\Theta) = & \sum_{l=0:N} \left[\sum_{i=1}^{|S^l|} \lambda g(y_{i-1}^l, y_i^l, t_i, \mu_i, \mathbf{X}^l) + \sum_j \theta_j f_j(W_R) \right] \\ & - \sum_{l=0:N} \log Z(\mathbf{X}^l) - \sum_{k=1}^K \frac{\lambda_k^2}{2\sigma_\lambda^2} - \sum_{j=1}^J \frac{\theta_j^2}{2\sigma_\theta^2} \end{aligned} \quad (6.8)$$

The derivative of this function with respect to parameter λ_k is

$$\begin{aligned} \frac{\partial \mathcal{L}(\Theta)}{\partial \lambda_k} = & \sum_{l=0:N} \sum_{i=1}^{|S^l|} g_k(y_{i-1}^l, y_i^l, t_i, \mu_i, \mathbf{X}^l) - \sum_{l=0:N} \sum_{i=1}^{|S^l|} \sum_{S, S'} g_k(S, S', \mathbf{X}^l) \\ & \times P(S, S' | \mathbf{X}^l) - \sum_{k=1}^K \frac{\lambda_k}{\sigma_\lambda^2} \end{aligned} \quad (6.9)$$

Similarly, the partial derivative of the log-likelihood with respect to parameter θ_j is

$$\begin{aligned} \frac{\partial \mathcal{L}(\Theta)}{\partial \theta_j} = & \frac{\partial}{\partial \theta_j} \left(\sum_{l=0:N} \log P(\{S^l, R^l = W_R\} | \mathbf{X}^l) \right) \\ = & \sum_{l=0:N} \sum_j f_j(W_R) - \sum_{l=0:N} \sum_j \sum_{W_R, W'_R} f_j(W'_R) P(S^l, W_R, W'_R | \mathbf{X}^l) - \sum_j \frac{\theta_j}{\sigma_\theta^2} \end{aligned} \quad (6.10)$$

Computing the expectations in the log-likelihood can be very expensive, since counting the number of true groundings of a formula in a domain is intractable. Instead we can take the derivative of the pseudo-log-likelihood as

$$\begin{aligned} \frac{\partial \mathcal{L}^*(\Theta)}{\partial \theta_j} = & \frac{\partial}{\partial \theta_j} \left(\sum_{l=0:N} \log P^*(\{S^l, R^l = W_R\} | \mathbf{X}^l) \right) = \frac{\partial}{\partial \theta_j} \left(\sum_{c=1}^n \log P(X_{c=x_c} | MB_{W_R}(\mathbf{X}_c)) \right) \\ = & \sum_j \sum_{c=1}^n \left[f_j(W_R) - P(S, W_{R_{c=0}} | MB_{W_R}(\mathbf{X}_c)) f_j(W_{R_{c=0}}) \right. \\ & \left. - P(S, W_{R_{c=1}} | MB_{W_R}(\mathbf{X}_c)) f_j(W_{R_{c=1}}) \right] - \sum_j \frac{\theta_j}{\sigma_\theta^2} \end{aligned} \quad (6.11)$$

where $MB_{WR}(\mathbf{X}_c)$ is the state of the Markov blanket of \mathbf{X}_c in the training data.

However, gradient-based exact parameter estimation requires computing the marginal probabilities (e.g., $P(S, S'|\mathbf{X}^t)$) and the normalization constant $Z(\mathbf{X})$, which are prohibitively intractable in our model. In that case, approximate techniques can be applied to compute the gradient. In the following, we will discuss approximate parameter estimation in our model.

6.2.4 Approximate Parameter Estimation via Structured Variational Inference

As we shown, working directly with the above function using the maximum likelihood (ML) estimation is typically precluded by the need to compute the normalization constant $Z(\mathbf{X})$ of the model. Unlike ML learning, the basic idea of variational inference [40][37][87] is to reformulate the computation of a marginal or conditional probability in terms of a simplified optimization problem, especially in the context of the exponential family. Solving this problem gives an approximation of probabilities of interest.

Let $Q(Y|Y_o, \mathbf{X})$ be the *variational distribution* and it serves as an approximation of $P(Y|Y_o, \mathbf{X})$. Under the IID assumption, we ignore the summation operator $\sum_{l=0:N}$ in the log-likelihood during the following derivations since there is no essential difference between one sample and a set of samples. According to the variational mean field theory, the optimal solution is the distribution that has the minimum Kullback-Leibler (KL) divergence between the two distributions Q and P , where the KL divergence is defined as follows:

$$\begin{aligned} KL(Q||P) &= \sum_Y Q(Y|Y_o, \mathbf{X}) \log \frac{Q(Y|Y_o, \mathbf{X})}{P(Y|Y_o, \mathbf{X})} \\ &= \sum_{S,R} Q(S, R|S_o, R_o, \mathbf{X}) \log \frac{Q(S, R|S_o, R_o, \mathbf{X})}{P(S, R|S_o, R_o, \mathbf{X})}. \end{aligned} \quad (6.12)$$

Given the non-negativity property of the KL divergence, and take $\log P(Y|Y_o, \mathbf{X}) = \log P(Y, Y_o|\mathbf{X}) - \log P(Y_o|\mathbf{X})$ into Equation 6.7, we can easily ob-

tain:

$$\begin{aligned}\mathcal{L}(\Theta) &= \log P(S_o, R_o | \mathbf{X}) - KL(Q || P) \\ &= \sum_{S,R} Q(S, R | S_o, R_o, \mathbf{X}) \left[-\log Q(S, R | S_o, R_o, \mathbf{X}) \right. \\ &\quad \left. + \log P(S, R, S_o, R_o | \mathbf{X}) \right] \end{aligned} \quad (6.13)$$

$$= \mathbb{H}(Q) + \mathbb{E}_Q \left\{ \log P(S, R, S_o, R_o | \mathbf{X}) \right\} \quad (6.14)$$

$$\leq \mathcal{L}(\Theta) \quad (6.15)$$

where $\mathbb{H}(Q) = -\sum_{S,R} Q(S, R | S_o, R_o, \mathbf{X}) \log Q(S, R | S_o, R_o, \mathbf{X})$ is the entropy of the variational distribution, and $\mathbb{E}_Q \{ \log P(S, R, S_o, R_o | \mathbf{X}) \} = \sum_{S,R} Q(S, R | S_o, R_o, \mathbf{X}) \log P(S, R, S_o, R_o | \mathbf{X})$ is the expectation with respect to $Q(S, R | S_o, R_o, \mathbf{X})$. Clearly, $\mathcal{L}(\Theta)$ is the lower bound of the log-likelihood $\mathcal{L}(\Theta)$. Thus by maximizing $\mathcal{L}(\Theta)$ we will always recover the log-likelihood of the data $\mathcal{L}(\Theta^*) = \log P(S_o, R_o | \mathbf{X}) - 0$.

In statistical physics, $\mathcal{F} = -\mathcal{L}(\Theta)$ is call the *variational free energy* and the lower bound $\mathcal{L}(\Theta)$ can be expressed as the difference of two free energies as $\mathcal{L}(\Theta) = \mathcal{F}_\infty - \mathcal{F}_0$, where $\mathcal{F}_\infty = -\log Z(\mathbf{X})$ is the free energy when we use model distribution with all variables free, and \mathcal{F}_0 is the free energy when we use data distribution with observable labels clamped to their values. Intuitively, to optimize the lower bound, we take derivatives of $\mathcal{L}(\Theta)$ with respect to λ_k :

$$\begin{aligned}\frac{\partial \mathcal{L}(\Theta)}{\partial \lambda_k} &= \frac{\partial}{\partial \lambda_k} \left\langle \log P(S, R, S_o, R_o | \mathbf{X}) \right\rangle_{Q(S, R | S_o, R_o, \mathbf{X})} \\ &= \sum g(y_{i-1}, y_i, t_i, \mu_i, \mathbf{X}) + \frac{\partial \mathcal{F}_\infty}{\partial \lambda_k}, \end{aligned} \quad (6.16)$$

where $\langle \cdot \rangle_Q$ is the expectation under distribution Q . Similarly, the derivatives with respect to θ_k is:

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \theta_k} = \sum_k f_k(W_R) + \frac{\partial \mathcal{F}_\infty}{\partial \theta_k} \quad (6.17)$$

Unfortunately, in both Equation 6.16 and 6.17, the derivatives of the free energy \mathcal{F}_∞ are prohibitively intractable in our model.

Approximate inference methods rely on additional structure in the joint distribution beyond what is already explicated by the graph. For example, the probability model corresponding to a fully connected graph may factor into a product of pairwise potential functions depending only the variables

associated with each undirected edge. Note that we can easily collect together the edge potentials into larger clique potentials. Mean field and other approximate inference algorithms heavily exploit this type of additional factorization structure (the clique potentials or conditional probabilities may also possess useful additional parametric structure, other than factorization discussed above).

It is important to choose a family of Q such that we can tractably optimize the KL divergence in Equation 6.12. The simplest naïve variational mean field theory assumes a fully factorized distribution (the interacted variables are independent and the joint distribution is a product of single variable marginal probabilities) and leads to computational tractability as

$$\begin{aligned} Q(S, R|S_o, R_o, \mathbf{X}) &= \sum_{i,j} Q(S_i, R_j|S_o, R_o, \mathbf{X}) \\ &= \sum_i Q(S_i|S_o, R_o, \mathbf{X}) \sum_j Q(R_j|S_o, R_o, \mathbf{X}) \end{aligned} \quad (6.18)$$

However, the assumption of a completely factorized distribution is a very strong one, and it may not yield sufficiently accurate results [37]. The essential principles underlying the mean field approach are not limited to fully factorized distributions, and a natural approach to improving over this simple mean field method is to combine it with exact probabilistic calculations. More generally, we can consider classes of tractable distributions that incorporate additional substructure which could be readily handled with exact methods. In this structured mean field approach, exact probability calculations on tractable substructures are combined with variational methods to capture the interactions between substructures [73][89][37].

Let s be an instantiation of the variable set S , and r be an instantiation of the variable set R , respectively. Recall that the target distribution of our model $P(Y|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_i \Omega_i(d_i)$, where $\Omega_i(d_i) = \exp [\sum_{i=1}^{|S|} \lambda g(y_{i-1}, y_i, t_i, \mu_i, \mathbf{X}) + \sum_j \theta_j f_j(W_R)]$, and d_i is the projection of the instantiations s and r to the variables in $KL_i \subseteq \{R, S\}$, and the subsets $\{KL_{i=1}^T\}$ can be overlapped. Suppose V_1, \dots, V_M are subsets (clusters) of variables $\{R, S\}$, and v_m is the projection of the instantiation $\{r, s\}$ to the variables in V_m . The entropy $\mathbb{H}(Q)$ in Equation 6.12 can be rewritten as the following well-known form:

$$\begin{aligned} \mathbb{H}(Q) &= - \sum_{V_m} Q(v_m) \log Q(v_m) \\ &\quad - \sum_{V_m} Q(v_m) \sum_n \sum_{V_n \setminus V_m} Q(v_n|v_m) \log Q(v_n|v_m) \end{aligned} \quad (6.19)$$

Similarly, the expectation $\mathbb{E}_Q\{\log P(S, R|S_o, R_o, \mathbf{X})\}$ is written as

$$\begin{aligned} \mathbb{E}_Q\{\log P(S, R|S_o, R_o, \mathbf{X})\} = \\ \sum_{v_m} Q(v_m) \sum_i \sum_{KL_i \setminus v_m} Q(d_i|v_m) \log \Omega_i(d_i) - \log(Z(\mathbf{X})) \end{aligned} \quad (6.20)$$

Therefore, the KL divergence defined in Equation 6.12 can be rewritten as

$$\begin{aligned} KL(Q||P) &= -\mathbb{H}(Q) - \mathbb{E}_Q\{\log P(S, R|S_o, R_o, \mathbf{X})\} \\ &= \sum_{v_m} Q(v_m) \log Q(v_m) - \sum_{v_m} Q(v_m) \left[- \sum_n \sum_{v_n \setminus v_m} Q(v_n|v_m) \right. \\ &\quad \times \log Q(v_n|v_m) + \sum_i \sum_{KL_i \setminus v_m} Q(d_i|v_m) \log \Omega_i(d_i) \left. \right] + \log(Z(\mathbf{X})) \\ &= \sum_{v_m} Q(v_m) \log \frac{Q(v_m)}{\Upsilon_m(v_m)} + \log(Z(\mathbf{X})) \end{aligned} \quad (6.21)$$

where $\Upsilon_m(v_m) = \exp \left[- \sum_n \sum_{v_n \setminus v_m} Q(v_n|v_m) \log Q(v_n|v_m) + \sum_i \sum_{KL_i \setminus v_m} Q(d_i|v_m) \log \Omega_i(d_i) \right]$.

With a little abuse of notations, we use $Q(S, R)$ to denote $Q(S, R|S_o, R_o, \mathbf{X})$. To find a distribution Q minimizing the KL divergence between Q and P , we assume that it can be factorized as $Q(S, R) = Q(S)Q(R)$, and we further assume $Q(S)$ to be of the form $Q(S) = \frac{1}{Z_{Q_S}} \prod_j \phi_j(u_j)$ and $Q(R)$ to be $Q(R) = \frac{1}{Z_{Q_R}} \prod_k \psi_k(w_k)$. Z_{Q_S} and Z_{Q_R} are two local normalization factors. Suppose U_1, \dots, U_J are possibly overlapped subsets of the variable S , and W_1, \dots, W_K are possibly overlapped subsets of the variable R , then u_j is the projection of the instantiation s to the variables in U_j and w_k is the projection of the instantiation r to the variables in W_k , respectively. We also denote the un-normalized distributions as $\tilde{Q}(S) = \prod_j \phi_j(u_j)$ and $\tilde{Q}(R) = \prod_k \psi_k(w_k)$, $\tilde{Q}(S) \propto Q(S)$ and $\tilde{Q}(R) \propto Q(R)$. We define $Q(u_m|u_j) = \frac{1}{|U_m \setminus U_j|}$ for instantiations $U_j = u_j$ for which $Q(u_j) = 0$. Thus, all terms in the equality $Q(u_m, u_j) = Q(u_j)Q(u_m|u_j)$ are well defined. And $\sum_{u_m \setminus U_j} Q(u_m|u_j) = 1$.

We propose an iterative converging algorithm to find distributions $Q(S)$ and $Q(R)$ based on [89], such that in each iteration the KL divergence between Q and P decreases until Q reaches an equilibrium state. This structured variational inference algorithm is shown in Algorithm 2: for $Q(S)$, it iterates over all clusters $U_j (1 \leq j \leq J)$ and corresponding instantiations u_j via Equations 6.22 and 6.23 to update potentials $\phi_j(u_j)$, where h_{mj} and e_{ij} are two indicators, and they are defined as:

$$h_{mj} = \begin{cases} 0, & U_m \cap U_j = \emptyset, \\ 1, & \text{otherwise.} \end{cases}, \quad e_{ij} = \begin{cases} 0, & KL_i \cap U_j = \emptyset, \\ 1, & \text{otherwise.} \end{cases}$$

At each step, this algorithm uses an inference procedure to compute conditional probabilities $Q(u_m|u_j)$ and $Q(d_i|u_j)$ from an un-normalized distribution \tilde{Q} where $\tilde{Q} = \prod_j \phi_j(u_j)$. This is accomplished by using any inference algorithms such as the sum-product algorithm [45]. For example, to compute $Q(u_m|u_j)$, the algorithm first computes $\tilde{Q}(u_m, u_j)$ and then $\tilde{Q}(u_j)$. $Q(u_m|u_j)$ is the ratio of these two quantities since the normalization factor cancels. More importantly, for \tilde{Q} the calculation of these conditional probabilities is not affected by multiplying any $\phi_j(u_j)$ by a constant. For the distribution $Q(R)$, similar algorithm can be performed to iterate over all clusters $W_k (1 \leq k \leq K)$ to update potentials $\psi_k(w_k)$, where h'_{nk} and e'_{ik} are two indicators, analogous to h_{mj} and e_{ij} , respectively. Similar algorithms described in [91][92] can be used to generate the clusters $U_j (1 \leq j \leq J)$ and $W_k (1 \leq k \leq K)$ for our model.

Note that the structured mean field algorithm described above generalizes the naïve mean field algorithm, which is just a special case of this algorithm in that each U_j or W_k contains only one variable. Note that $\Upsilon_m(v_m)$ does not depend on potentials $\phi_j(u_j)$ or $\psi_k(w_k)$ being optimized. This algorithm computes $\phi_j(u_j)$ and $\psi_k(w_k)$ hence decreases the KL divergence in each iteration by improving $\phi_j(u_j)$ and $\psi_k(w_k)$ while holding all other potentials fixed. Consequently, it converges to an equilibrium state of the KL divergence between Q and P among all distributions Q of the given form $Q(S) = \frac{1}{Z_{Q_S}} \prod_j \phi_j(u_j)$ and $Q(R) = \frac{1}{Z_{Q_R}} \prod_k \psi_k(w_k)$. This shows that the algorithm is theoretically sound and correct.

Now, we summarize the whole parameter estimation procedure as follows: first the distribution $Q(S)$ is computed iteratively via Equations 6.22 and 6.23, and $Q(R)$ is updated via Equations 6.24 and 6.25. We can obtain the variational distribution indexed by a set of *variational parameters* as $Q_{\lambda_v, \theta_v}(S, R)$. Then, the EM or gradient-based optimization algorithms can be applied to update these variational parameters in the model. In particular, we exploit the limited memory quasi-Newton (L-BFGS) algorithm [49] in the learning procedure since this algorithm is efficient and works well for many optimization problems.

6.2.5 Bidirectional MCMC Sampling for Inference

For inference, the objective is to find the most likely segmentation assignments S^* and corresponding most likely relation assignments R^* , that is,

$Y^* = \{R^*, S^*\} = \arg \max_Y P(Y|\mathbf{X})$. Unfortunately, the exact inference for Y^* is general intractable, since we need to search a huge number of possible segmentation assignments S and worlds of relation assignments R . Consequently, approximate inference becomes an alternative. We propose a bi-directional MCMC sampling algorithm to find the maximum a posteriori (MAP) assignment of all the variables of this model. This algorithm is strongly coupled to joint inference based on efficient Metropolis-Hastings (MH) sampling [56][33] from both semi-Markov chains and ground Markov networks in an iterative manner to find an approximate solution for Y^* . It allows information to flow in both directions, such that evidences from segmentations and relations can be well exploited. It is a theoretically well-founded MCMC algorithm, and is guaranteed to converge.

The MCMC methods are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its stationary distribution. Let $S^{(t)}$ be the current state of segmentation sequence S and $S^{(t+1)}$ be the next state of S . We assume that the current relation sample $R^{(t)}$ has already been drawn and we keep it unchanged. To draw segmentation samples from $Q(S|R^{(t)}, \mathbf{X})$ 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:

$$Q(S^{(t+1)}|S^{(t)}, R^{(t)}, \mathbf{X}) = Q((S_{i1} \cdots S_{ik})^{(t+1)}|S_{-i}^{(t)}, R^{(t)}, \mathbf{X}) \quad (6.26)$$

where S_{-i} is all segments except S_i , $S_{-i} = (S_1, \dots, S_{i-1}, S_{i+1}, \dots, S_L)$ and $S_{-i}^{(t)} = S_{-i}^{(t+1)}$. 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 $i = L$, 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 6.26. This distribution is easy to compute in the semi-Markov chains. 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 L segments, we can sample the whole segmentation sequences from the conditional distribution

$$Q(S^{(t+1)}|S^{(t)}, R^{(t)}, \mathbf{X}) = \frac{Q((S_{i1} \cdots S_{ik})^{(t+1)}, S_{-i}^{(t)}, R^{(t)}, \mathbf{X})}{\sum_{y_{ij} \in \mathcal{Y}} Q((S_{i1} \cdots S_{ik})^{(t+1)}, S_{-i}^{(t)}, R^{(t)}, \mathbf{X})} \quad (6.27)$$

To avoid low grade samples, an MH step of the target distribution $Q(S^{(t+1)}|R^{(t)}, \mathbf{X})$ and the proposal distribution $\Pi(\hat{S}|S^{(t)}, R^{(t)}, \mathbf{X})$ involves sampling a

candidate sequence \hat{S} given the current value $S^{(t)}$ according to $\Pi(\hat{S}|S^{(t)}, R^{(t)}, \mathbf{X})$, and uses an acceptance/rejection scheme to define a transition kernel with $Q(S^{(t+1)}|S^{(t)}, R^{(t)}, \mathbf{X})$. 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 [42] in the MH algorithm, and the acceptance probability $\mathcal{A}(S^{(t)}, \hat{S})$ is written as

$$\mathcal{A}(S^{(t)}, \hat{S}) = \min \left\{ 1, \frac{Q^{1/c_t}(\hat{S}|R^{(t)}, \mathbf{X})\Pi(S^{(t)}|\hat{S}, R^{(t)}, \mathbf{X})}{Q^{1/c_t}(S^{(t)}|R^{(t)}, \mathbf{X})\Pi(\hat{S}|S^{(t)}, R^{(t)}, \mathbf{X})} \right\} \quad (6.28)$$

where c_t is a decreasing cooling schedule with $\lim_{t \rightarrow \infty} c_t = 0$. As $c_t \rightarrow 0$ the distribution becomes sharper, and when $c_t = 0$ the distribution places all of its mass on the maximal outcome, having the effect that the Markov chain always climbs uphill. Thus if we gradually decrease c_t from 1 to 0, the Markov chain increasingly tends to go uphill. The proposal distribution $\Pi(\hat{S}|S^{(t)}, R^{(t)}, \mathbf{X})$ can be computed via Equation 6.27, and $\Pi(S^{(t)}|\hat{S}, R^{(t)}, \mathbf{X})$ can also be easily computed as

$$\begin{aligned} \Pi(S^{(t)}|\hat{S}, R^{(t)}, \mathbf{X}) &= Q(S_i^{(t)}|\hat{S}_{-i}, R^{(t)}, \mathbf{X}) \\ &= \frac{Q(S_i^{(t)}, \hat{S}_{-i}, R^{(t)}, \mathbf{X})}{\sum_{y_i \in \mathcal{Y}} Q(S_i^{(t)}, \hat{S}_{-i}, R^{(t)}, \mathbf{X})} \end{aligned} \quad (6.29)$$

After we obtain the segmentation sample $S^{(t+1)}$, we can draw relation samples from $Q(R|S^{(t+1)}, \mathbf{X})$, which is defined by the ground Markov network in the model. And similar MH procedure can also be exploited. The basic step consists of sampling one ground atom given its Markov blanket. The Markov blanket of a ground atom is the set of ground atoms that appear in some grounding of a formula with it. Let $f_i^t = f_i(R_l = r_l, MB^{(t)})$ be the truth value of the feature corresponding to the i -th ground formula when $R_l = r_l$ and its Markov blanket $MB^{(t)}$ at state t . Let $f_i^1 = f_i(R_l = 1, MB^{(t)})$ and $f_i^0 = f_i(R_l = 0, MB^{(t)})$. The probability of a ground atom R_l for $MB^{(t)}$ is

$$Q(R_l = r_l|MB^{(t)}) = \frac{\exp(\sum \theta_i f_i^r)}{\exp(\sum \theta_i f_i^1) + \exp(\sum \theta_i f_i^0)}. \quad (6.30)$$

When all ground atoms are sampled, we can obtain the relation sample at state t as $R^{(t)}$. Sampling the next state $t + 1$ can be done according to the transition probability $Q(R^{(t+1)}|R^{(t)}, S^{(t+1)}, \mathbf{X})$, which is defined by changing the truth value assignment of one ground atom while keeping others the

same as before. Similarly, a candidate relation sample \hat{R} is accepted with probability $\mathcal{A}'(R^{(t)}, \hat{R})$ as

$$\mathcal{A}'(R^{(t)}, \hat{R}) = \min \left\{ 1, \frac{Q^{1/c'_t}(\hat{R}|S^{(t+1)}, \mathbf{X})\pi(R^{(t)}|\hat{R}, S^{(t+1)}, \mathbf{X})}{Q^{1/c'_t}(R^{(t)}|S^{(t+1)}, \mathbf{X})\pi(\hat{R}|R^{(t)}, S^{(t+1)}, \mathbf{X})} \right\} \quad (6.31)$$

and rejected with probability $1 - \mathcal{A}'(R^{(t)}, \hat{R})$. c'_t is another decreasing cooling schedule similar to c_t .

Algorithm 3 summarizes the bi-directional MH inference procedure, which performs efficient sampling for both segmentations and relations bi-directionally and iteratively to capture mutual benefits between them. We run this algorithm to sample enough number of segmentations and relations, and this algorithm is guaranteed to converge to its stationary distribution. Thus, it will produce an approximated solution of the most likely pair $Y^* = \{R^*, S^*\}$.

Algorithm 2: The structured variational inference for distributions $Q(S)$ and $Q(R)$

Input: A set of potentials $\Omega_i(d_i)$ defining the target distribution $P(Y|\mathbf{X})$, a set of clusters U_j ($1 \leq j \leq J$) with initial potentials $\phi_j(u_j)$, and a set of clusters W_k ($1 \leq k \leq K$) with initial potentials $\psi_k(w_k)$.

Output: Revised sets of potentials $\phi_j(u_j)$ and $\psi_k(w_k)$ defining distributions $Q(S) = \frac{1}{Z_{Q_S}} \prod_j \phi_j(u_j)$ and $Q(R) = \frac{1}{Z_{Q_R}} \prod_k \psi_k(w_k)$, such that $Q(S, R) = Q(S)Q(R)$ is an equilibrium state of the KL divergence $KL(Q||P)$.

while not converge do

foreach instantiation u_j of cluster U_j **do**

$$\begin{aligned} \alpha_j(u_j) \leftarrow & - \sum_{\{m:h_{mj}=1\}} \sum_{U_m \setminus U_j} Q(u_m|u_j) \log Q(u_m|u_j) \\ & + \sum_{\{i:e_{ij}=1\}} \sum_{KL_i \setminus U_j} Q(d_i|u_j) \log \Omega_i(d_i) \end{aligned} \quad (6.22)$$

$$\phi_j(u_j) \leftarrow \exp [\alpha_j(u_j)] \quad (6.23)$$

end

foreach instantiation w_k of cluster W_k **do**

$$\begin{aligned} \beta_k(w_k) \leftarrow & - \sum_{\{n:h'_{nk}=1\}} \sum_{W_n \setminus W_k} Q(w_n|w_k) \log Q(w_n|w_k) \\ & + \sum_{\{i:e'_{ik}=1\}} \sum_{KL_i \setminus W_k} Q(d_i|w_k) \log \Omega_i(d_i) \end{aligned} \quad (6.24)$$

$$\psi_k(w_k) \leftarrow \exp [\beta_k(w_k)] \quad (6.25)$$

end

end

Algorithm 3: The bi-directional Metropolis-Hastings sampling inference algorithm

Input: Initialized segmentation and relation assignments S^0 and R^0 , iteration \mathbb{T} as sample size.

Output: Approximated most likely pair $\{R^*, S^*\}$ taken from $S^0 \dots S^{\mathbb{T}}$ and $R^0 \dots R^{\mathbb{T}}$.

for $t = 0, 1, 2, \dots, \mathbb{T}$ **do**

 Draw $\hat{S} \sim Q(S|R^{(t)}, \mathbf{X})$

 Compute $\mathcal{A}(S^{(t)}, \hat{S})$

 With probability $\mathcal{A}(S^{(t)}, \hat{S})$ set $S^{(t+1)} = \hat{S}$,
 otherwise set $S^{(t+1)} = S^{(t)}$

 Draw $\hat{R} \sim Q(R|S^{(t+1)}, \mathbf{X})$

 Compute $\mathcal{A}'(R^{(t)}, \hat{R})$

 With probability $\mathcal{A}'(R^{(t)}, \hat{R})$ set $R^{(t+1)} = \hat{R}$,
 otherwise set $R^{(t+1)} = R^{(t)}$

end

Chapter 7

Experiments

In this chapter, we perform experimental study by applying our proposed models, including the bidirectional integrated models (**bidirectional models**) in chapter 4, the integrated discriminative probabilistic models (**integrated models**) in chapter 5, and the joint models incorporating logic (**joint models**) in chapter 6, to three well-investigated IE tasks — Chinese named entity recognition (NER), entity identification and relation extraction from Wikipedia’s encyclopedic articles, and citation matching. Empirical results on real-world datasets show that our proposed models achieve substantial improvements over current state-of-the-art probabilistic models, illustrating the promise of our approaches. We compare and discuss the merits of our models against others. Several interesting issues, such as the superiority of the bidirectional MH inference algorithm, are also investigated.

7.1 Chinese NER

7.1.1 Data

We used one-month data from People’s Daily (January-Jun, 1998) corpus for Chinese NER experiments, which contains 44818 sentences, with tagged entities of 19879 person, 25661 location, and 11590 organization names, respectively. See Section 3.3.1 for more details of this corpus.

7.1.2 Methodology

We used features that have been shown to be very effective for NER, namely the current character and its POS tag, several characters surrounding the current character and their POS tags, current word and several words surrounding the current word, and some clue word features which can capture

non-local dependencies. In addition, we take the advantage of using entity-level dictionary features. This gives us a rich feature set using both local and non-local information.

We extracted 165 location salient words and 843 organization salient words from Wikipedia and the LDC¹ Xinhua News database. We also made a punctuation list which contains 18 items and some stopwords that named entities cannot contain. The stopwords are mainly conjunctions, auxiliary and functional words. We introduced various types of domain knowledge which can capture essential characteristics of NER and can be well and concisely formulated in first-order logic. The considered domain knowledge is listed as follows, and the corresponding first-order logic is shown in Table 7.1. The goal of logical inference is to determine whether the candidates are entities and the types of entities by answering the query predicates (e.g., *per*, *loc*, *org*, and *non_entity*) given the *Evidence* predicates (e.g., *endwith(r,p)*) and other relations that can be deterministically derived.

- **Occur in Train:** Definitely, if an entity occurs in the training data as a PER or a LOC or an ORG, then this entity should be a PER or a LOC or an ORG in the testing data.
- **LOC+SW:** If an entity candidate ends with a location salient word, then it should be a LOC.
- **ORG+SW:** If an entity candidate ends with an organization salient word, then it should be an ORG.
- **Label Consist:** If two entity candidates are exactly the same, they should be consistently labeled to the same entity type.
- **Restriction:** Intuitively, all entity candidates cannot comprise any stopword or punctuation.
- **Noun:** Since all entities are proper nouns, each entity candidate should be a noun or a noun phrase.

We perform 10-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 $F_{\beta=1}$ which is the harmonic mean of P and R ($F_{\beta=1} = \frac{2PR}{P+R}$). We compare our approaches with three models: CRFs, Semi-CRFs, and MLNs for this task. All these models exploit standard parameter learning and inference algorithms. We set the sample size to 10000 for both MH and bidirectional MH inference algorithms for our models. To

¹The Linguistic Data Consortium, see <http://www ldc.upenn.edu/>

Table 7.1: Domain knowledge and corresponding first-order formulas for NER.

Occur in Train	$\text{occur_per}(p) \Rightarrow \text{per}(p)$ $\text{occur_loc}(p) \Rightarrow \text{loc}(p)$ $\text{occur_org}(p) \Rightarrow \text{org}(p)$
LOC+SW	$\text{endwith}(r,p) \wedge \text{locsalient}(p) \Rightarrow \text{loc}(r)$
ORG+SW	$\text{endwith}(r,p) \wedge \text{orgsalient}(p) \Rightarrow \text{org}(r)$
Label Consist	$\text{same_str}(p,q) \Rightarrow \text{same_label}(p,q)$
Restriction	$\text{containstop}(p) \Rightarrow \text{non_entity}(p)$ $\text{containpunc}(p) \Rightarrow \text{non_entity}(p)$
Noun	$\text{notnoun}(p) \Rightarrow \text{non_entity}(p)$

make accurate and fair comparison, we use the same set of features for all these models. For **CRFs** and **Semi-CRFs**, the first-order domain knowledge is transformed into binary features. For **MLNs**, all features are presented via first-order logic. The ground Markov network in our models (e.g., the **integrated models** and the **joint models**) consists of 14 predicates, 16620 constants and 97992 ground atoms. It also contains a total of 9878 tuples (i.e., there are 9878 true ground atoms).

7.1.3 Experimental Results and Analysis

The comparative performance is summarized in Table 7.2. As can be seen, our proposed models yield substantially better results than all the three baseline models. Notably, the **joint model** obtains the best performance, leading to a relative error reduction of up to 34.07% on the overall F-measure over the **CRF** model, a relative error reduction of up to 33.67% on the overall F-measure over the **Semi-CRF** model, and a relative error reduction of up to 49.31% over the **MLN** model, respectively. All the improvements are statistically significant according to McNemar’s paired tests (p -value < 0.05 with a 95% confidence interval).

It is worth noticing that our proposed models boosted the performance for all 3 entity types. For example, when compared to the **CRF** model, the improvement for the **joint model** on the F-measure is 2.91% for person, 1.98% for location, and 3.31% for organization, respectively. This can be explained by the fact that there are much more sub-structures existing in organization names than in person or location names. In that case, modeling the internal sub-structures is more helpful for organization names in the NER task. This phenomenon also demonstrates the advantage and capability of our model for effective sub-structure modeling in named entities.

The **Semi-CRF** model slightly outperforms the **CRF** model (90.08 vs.

Table 7.2: Comparative performance of our models, CRFs, Semi-CRFs, and MLN models for NER.

Entities	CRFs			Semi-CRFs			MLNs		
	P	R	F_1	P	R	F_1	P	R	F_1
person	92.12	90.57	91.34	92.10	90.67	91.38	92.85	79.99	85.94
location	90.62	89.74	90.18	90.53	89.96	90.24	91.79	84.23	87.85
organization	89.72	85.44	87.53	89.62	85.78	87.66	88.60	84.68	86.60
Overall	90.89	89.16	90.02	90.82	89.35	90.08	86.55	87.49	87.02

Entities	Bidirectional			Integrated			Joint		
	P	R	F_1	P	R	F_1	P	R	F_1
person	93.90	93.03	93.46	93.90	93.54	93.72	94.55	93.95	94.25
location	92.10	90.75	91.42	92.38	91.07	91.72	92.77	91.55	92.16
organization	90.92	89.40	90.15	91.21	89.55	90.37	91.75	89.95	90.84
Overall	92.50	92.06	92.28	92.90	92.35	92.62	93.92	92.93	93.42

90.02 on the overall F-score), since the **Semi-CRF** model captures segments instead of tokens for named entity recognition, thus entity-level dictionary features can be better exploited. It is not surprising that the **Semi-CRF** model performs better than the **MLN** model, since the NER task can be formulated as a sequence labeling problem, and can therefore be effectively modeled by probabilistic sequence segmentation approaches such as **Semi-CRFs**. The **MLN** model, however, can hardly capture the first-order Markov property in sequence data, leading to reduced performance on the NER task.

The **bidirectional model** substantially outperforms the three baseline models, however, its performance is worse than that of **integrated** and **joint** models. We analyze and explain the main reasons as follows. The power of **bidirectional model** extensively depends on the bidirectional nature of joint factors connecting variables of multiple subtasks. However, Chinese NEs have distinct linguistic characteristics or substructures in their composition. These substructures can be well modeled via first-order domain knowledge. In that case, taking into consideration these substructures will be more helpful. As can be seen, both **integrated** and **joint** models exploit substructures in Chinese NEs, resulting in enhanced performance consequently.

7.1.4 Bidirectionality

We study and test the advantages of our proposed bi-directional MH inference in the **joint model** by comparing it with the greedy, N -best list, and uni-

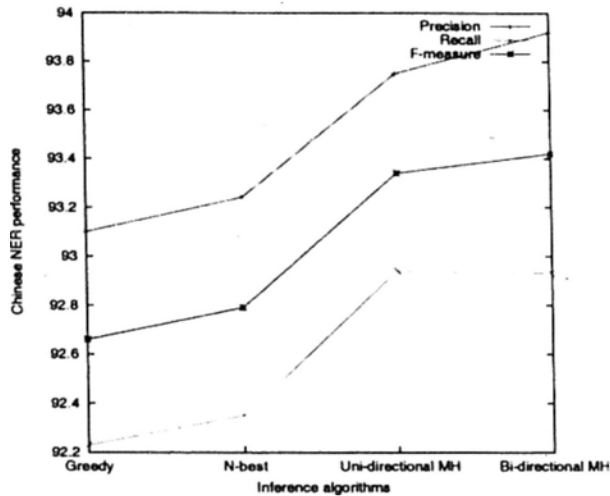


Figure 7.1: Performance comparison of different inference algorithms on Chinese NER.

directional MH sampling algorithms described in detail as follows:

- Greedy: this algorithm is a special case of the N -best list algorithm when $N = 1$, that is, it greedily takes the best output of segmentations and corresponding relations.
- N -best list: for this inference algorithm, we restrict our re-ranking targets to the N -best list $\mathbb{L} = \{L_1, L_2, \dots, L_N\}$, where $\{L_1, L_2, \dots, L_N\}$ is ranked by the conditional probability $Q(S|\mathbf{X})$. For a sequence \mathbf{X} , we maintain N -best segmentations over this sequence. For each segmentation S in this list, we can find a relation assignment R over S that maximizes the probability $Q(R|S, \mathbf{X})$. Given a particular segmentation assignment S , the most probable relation assignment R and its probability $Q(R|S, \mathbf{X})$ can be inferred. Having the N -best list of segmentation assignments and their corresponding relation assignments, the approximated solution that maximizes the joint probability $Q(\{R, S\}|\mathbf{X})$. And the most probable relation assignment along with this segmentation is our final output.
- Uni-directional MH: for this algorithm, we draw segmentation samples from $Q(S|\mathbf{X})$ and then we draw relation samples from $Q(R|S, \mathbf{X})$, given the generated segmentation samples. Note that unlike the bi-directional MH algorithm, this algorithm does not iteratively draw samples from both segmentations and relations. It only draws relation

samples based on the segmentation samples which have already been drawn. The inference information in this algorithm can only flow into one direction.

As shown in Figure 7.1, the bidirectional MH inference algorithm consistently and significantly outperforms the other three inference algorithms in the **joint model** for overall Chinese NER performance, except that compared to the uni-directional MH algorithm, the overall recall is slightly worse (92.94 vs. 92.93). But we found that this result is not statistically significant. As can be seen in Figure 7.1, the overall F-measures for the greedy, N -best list, uni-directional MH, and bidirectional MH sampling inference algorithms are 92.66, 92.79, 93.34, and 93.42, respectively. The greedy algorithm is very easy to implement, and it is computationally efficient, however, this algorithm cannot make use of the whole probability distribution as defined in the **joint model**. The N -best list and uni-directional MH algorithms can lead to useful improvement over the greedy algorithm. Here, we set $N = 10$ according to the holdout methodology. However, one disadvantage of the uni-directional MH algorithm is that it is only feed forward and information can only flow into one direction (from segmentation to relation), thus relation cannot guide segmentation for inference and decision making. The bidirectional MH algorithm achieves the best performance for capturing bidirectional information flow and sharing mutual benefits for both segmentation and relation.

7.2 Entity Identification and Relation Extraction From Wikipedia

7.2.1 Wikipedia

Wikipedia is the world's largest free online encyclopedia, representing the outcome of a continuous collaborative effort of a large number of volunteer contributors. Virtually any Internet user can create or edit a Wikipedia web page, and this "freedom of contribution" has a positive impact on both the quantity (fast growing number of articles) and the quality (potential mistakes are quickly corrected within the collaborative environment) of this online resource. Currently Wikipedia has approximately 9.25 million articles in more than 200 languages. Moreover, Wikipedia has the category hierarchy structure which is used to classify articles according to their content. All these characteristics make Wikipedia an appropriate resource for information extraction. Figure 7.2 gives a snapshot of Wikipedia Web page about the

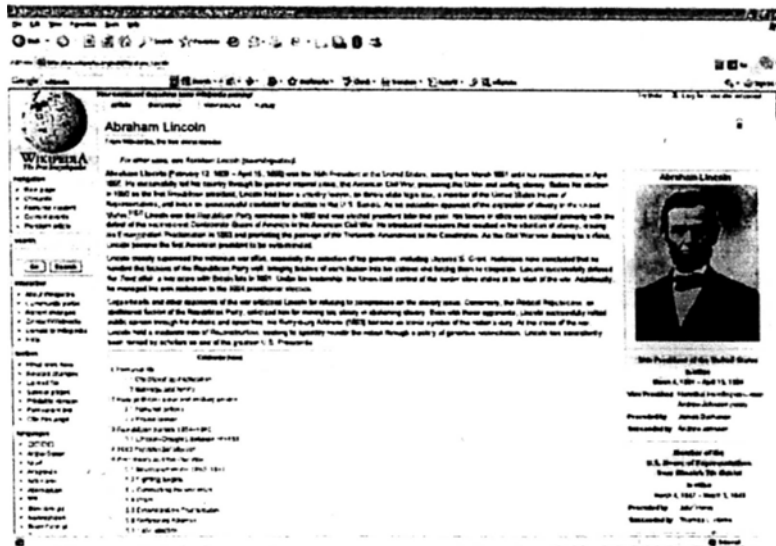


Figure 7.2: A snapshot of the encyclopedic article about Abraham Lincoln in Wikipedia.

great person Abraham Lincoln.

We investigate the problem of identifying entities and discovering semantic relationships between entity pairs from English encyclopedic articles in Wikipedia, which is a joint IE problem (see Section 1.2 for more details of this problem).

7.2.2 Data

We conducted experiments on entity identification and relation extraction from Wikipedia. The original dataset comes from [21]. However, all the entities are hyper-linked within the documents and the locations of them are already known. Another problem is that, the entity types are not classified. This dataset was used only for Wikipedia relation extraction, as in [21]. Thus, the original dataset is not directly suitable for our evaluation since our focus is on both secondary entity identification and the relation to the principal entity. To make this dataset appropriate for our experiments, we deleted all the hyper-links which define secondary entities in the documents, and we classified all secondary entities into fine-grained categories. The resulting dataset consists of 1127 paragraphs from 441 pages from the online encyclopedia Wikipedia. We labeled 7740 secondary 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

Table 7.3: Statistics of relation types and corresponding frequencies.

Relation	Freq	Relation	Freq
job_title	379	daughter	35
visited	368	husband	33
birth_place	340	religion	32
associate	326	influence	31
birth_year	287	underling	27
member_of	283	sister	20
birth_day	283	grandfather	20
opus	267	ancestor	19
death_year	210	grandson	18
death_day	199	inventor	15
education	185	cousin	13
nationality	148	descendant	11
executive	127	role	10
employer	111	nephew	9
death_place	93	uncle	6
award	86	supported_person	6
father	84	granddaughter	6
participant	81	owns	4
brother	71	great_grandson	4
son	68	aunt	4
associate_competition	58	supported_idea	3
wife	57	great_grandfather	3
superior	54	gpe_competition	3
mother	50	brother_in_law	2
political_affiliation	44	grandmother	1
friend	43	discovered	1
founder	43	Overall	4701

a founder relation between x and p , represented by $\text{founder_key}(x,p) \Rightarrow \text{founder}(x,p)$. Similarly, some morphological suffixes such as *-eer* and *-ician* of entity x may probably show a `job_title` relation to the principal entity p . Entity type is very helpful for relation extraction. If x is a location, its relation to p can only be `visited`, `birth_place`, or `death_place`. If x is an organization, its relation can only be `education`, `member_of`, or `employer`. Using first-order logic, we can also discover new relations (e.g., x 's father's father is the `grandfather`.) and handle the label consistency problem (e.g., if x and y are the same string, their relations to p should be the same.). More importantly, some formulas capture coherent interactions and dependencies between entities and relations. For example, if x is identified as a location candidate, then it has high probability of relation `visited`, `birth_place` or `death_place` to the principal person p , and vice versa. Since these formulas are not always true, our model can learn them under uncertainty by estimating the confidence of each formula.

Table 7.4: Some representative first-order formulas.

$\text{conjunction}(x,y) \Rightarrow \text{same_relation}(x,y)$
$\text{founder_key}(x,p) \Rightarrow \text{founder}(x,p)$
$\text{person}(p) \wedge \text{job_suffix}(x) \Rightarrow \text{job_title}(x,p)$
$\text{person}(p) \wedge \text{date}(x) \Rightarrow \text{birth_day}(x,p) \vee \text{death_day}(x,p)$
$\text{person}(p) \wedge \text{year}(x) \Rightarrow \text{birth_year}(x,p) \vee \text{death_year}(x,p)$
$\text{person}(p) \wedge \text{location}(x) \Rightarrow \text{visited}(x,p) \vee \text{birth_place}(x,p) \vee \text{death_place}(x,p)$
$\text{person}(p) \wedge \text{organization}(x) \Rightarrow \text{education}(x,p) \vee \text{member_of}(x,p) \vee \text{employer}(x,p)$
$\text{father}(x,y) \wedge \text{father}(y,z) \Rightarrow \text{grandfather}(x,z)$
$\text{husband}(x,y) \wedge \text{daughter}(z,x) \Rightarrow \text{mother}(y,z)$
$\text{father}(x,y) \Rightarrow \text{son}(y,x) \vee \text{daughter}(y,x)$
$\text{same_str}(x,y) \Rightarrow \text{same_relation}(x,y)$

7.2.4 Implicit Relation Extraction

Implicit relations are those that do not have direct contextual evidence. Implicit relations generally exist in different paragraphs, or even across documents. They require additional knowledge to be detected. Notably, these are the sorts of relations that are likely to have significant impact on performance. A system that can accurately discover knowledge that is implied by the text will effectively provide access to the implications of a corpus. Unfortunately, extracting implicit relations is challenging even for current state-of-the-art relation extraction models.

We show that our models can enable this technology. By employing the first-order logic formalism, the implicit relations can be easily discovered from text. Since these formulae will not always hold, we would like to handle them probabilistically by estimating the confidence of each formula.

Consider the following 2 sentences in Wikipedia articles (the principal entity is boxed and the secondary entities are in italic font):

1. On November 4, 1842 Abraham Lincoln married *Mary Todd*.
2. Abraham Lincoln had a son named *Robert Todd Lincoln* and he was born in Springfield, Illinois on 1 August 1843.

State-of-the-art extraction models may be able to detect the *wife* relation between *Mary Todd* and Abraham Lincoln, and the *son* relation between *Robert Todd Lincoln* and Abraham Lincoln successfully from local contextual clues. However, in the descriptive article of Robert Todd Lincoln

Table 7.5: Examples of first-order logic for implicit relation extraction.

$wife(x,y) \Rightarrow husband(y,x)$
$father(x,y) \Rightarrow son(y,x) \vee daughter(y,x)$
$brother(x,y) \Rightarrow brother(y,x) \vee sister(y,x)$
$husband(x,y) \wedge daughter(z,x) \Rightarrow mother(y,z)$
$father(x,y) \wedge father(y,z) \Rightarrow grandfather(x,z)$
$founder(x,y) \wedge superior(x,z) \Rightarrow employer(z,y)$
$associate(x,y) \wedge member_of(x,z) \Rightarrow member_of(y,z)$
$executive(x,y) \wedge member_of(z,y) \Rightarrow superior(x,z)$

in Wikipedia, Robert Todd Lincoln becomes the principal entity, and the *mother* relation between *Mary Todd* and Robert Todd Lincoln is only implied by the text and it is an implicit relation. First-order formalism allows the representation of deep and relational knowledge. Using the logic $wife(x,y) \wedge son(z,y) \Rightarrow mother(x,z)$, the relational knowledge in the above example can be easily captured to infer the implicit relation. These formulae are generally simple, and capture important knowledge for implicit relation extraction. Examples of first-order logic to infer implicit relations are listed in Table 7.5.

7.2.5 Methodology

We set the upper bound of the segment length L to 4 to enable efficient computation, since over 95% of the entities are within this threshold. That is, for each segment S_i , we have $1 \leq |S_i| \leq 4$. And we set the sample size to 10000 for both MH and bidirectional MH inference algorithms. The L-BFGS algorithm converged within 200 iterations for parameter learning. We perform four-fold cross-validation on this dataset, and take the average performance. Performance is evaluated by the standard measures of Precision (P), Recall (R), and $F_{\beta=1}$ for both entity identification and relation extraction tasks. We compare our approaches with two pipeline models **CRF+CRF**, **CRF+MLN**, and one joint model **Single MLN**, described in detail as follows:

- **CRF+CRF**: since we only extract relations between the principal entity and each mentioned secondary entity, this formulation allows us to view relation extraction as a sequence labeling task such as part-of-speech tagging. This model uses one linear-chain CRF [47] for entity recognition, and another linear-chain CRF for relation prediction. Here, relation extraction is viewed as a sequence labeling problem.

Each secondary entity's label is its relation to the principal entity, and we can capture the dependency between adjacent labels.

- **CRF+MLN**: this model uses Markov logic network (MLN) [69], a recently introduced framework for first-order logic, for relation extraction given the entity candidates from the CRF model.
- **Single MLN**: this model performs joint inference for both entity identification and relation extraction in a single MLN framework. We follow [67] to design some formulas capturing interactions such as "identify one entity can help to identify similar ones" in this model.

For the **CRF+CRF** model, for example, in the dataset it is common to see phrases such as "*Albert Einstein* (1879 - 1955) was born in *Germany*" for which the labels *birth_year*, *death_year*, and *birth_place* occur consecutively. Sequence models are specifically designed to handle these kinds of dependencies.

We exploit standard parameter estimation and inference algorithms for all these models. To avoid over-fitting, penalization techniques on likelihood are also performed. Note that for the second CRF in the **CRF+CRF** model, the first-order domain knowledge described in the above subsection is transformed into binary features, since CRF cannot handle first-order logic. For **Single MLN**, all features are formulated via first-order logic. Using typed variables and first-order domain knowledge described above, the total number of possible ground atoms in our **integrated** and **joint** models is 254966. The ground Markov network also contains a total of 10588 tuples (i.e., there are 10588 true ground atoms).

7.2.6 Performance of Entity Recognition

Table 7.6 shows the performance of entity identification and Table 7.7 to Table 7.8 show the performance of relation extraction of different models, respectively. From these results, we can see that our proposed models achieve the best performance for both tasks. In particular, the **joint model** obtains the overall F-measure of 94.11 on entity identification, and 68.59 on relation extraction task. For entity identification in Table 7.6, our **joint model** outperforms **CRF+CRF** and **CRF+MLN** by 4.99% on the overall F-measure, and **Single MLN** by 3.66% on the overall F-measure, respectively. For relation extraction in Table 7.7 and Table 7.8, our **joint model** outperforms **CRF+CRF** by 5.08%, **CRF+MLN** by 4.51%, and **Single MLN** by 3.62% on the overall F-measure, respectively. The improvement demonstrates the merits of our approaches by exploring tight interactions between entities and

Table 7.6: Comparative performance of our models, the CRF+CRF, CRF+MLN, and Single MLN models for entity identification from Wikipedia.

Entities	CRF+CRF			CRF+MLN			Single MLN		
	P	R	F_1	P	R	F_1	P	R	F_1
person	75.33	83.22	79.08	75.33	83.22	79.08	75.94	83.93	79.74
location	77.03	69.45	73.04	77.03	69.45	73.04	77.42	70.13	73.59
organization	53.78	47.76	50.59	53.78	47.76	50.59	54.11	47.06	50.34
date	98.54	97.53	98.03	98.54	97.53	98.03	97.79	95.68	96.72
year	97.14	99.10	98.11	97.14	99.10	98.11	98.01	99.03	98.52
time	60.00	20.33	30.37	60.00	20.33	30.37	50.00	15.38	23.53
number	98.88	60.33	74.94	98.88	60.33	74.94	100.0	66.07	79.57
miscellaneous	77.42	80.56	78.96	77.42	80.56	78.96	79.81	84.14	81.92
Overall	89.55	88.70	89.12	89.55	88.70	89.12	90.45	90.45	90.45

Entities	Bidirectional			Integrated			Joint		
	P	R	F_1	P	R	F_1	P	R	F_1
person	85.12	86.58	85.84	84.91	86.26	85.58	85.38	87.85	86.60
location	82.90	80.82	81.85	82.94	80.52	81.71	82.95	81.44	82.19
organization	65.45	65.50	65.47	64.63	65.61	65.12	72.43	63.69	67.78
date	98.60	95.98	97.27	98.60	95.98	97.27	98.90	96.24	97.55
year	98.06	99.12	98.59	97.15	99.42	98.27	97.36	99.55	98.44
time	100.0	30.00	46.15	100.0	25.00	40.00	100.0	33.00	49.62
number	100.0	65.00	78.79	100.0	60.00	75.00	100.0	65.52	79.17
miscellaneous	85.79	90.46	88.06	85.69	88.16	86.91	85.77	90.36	88.01
Overall	94.03	93.89	93.96	93.35	93.37	93.36	94.17	94.06	94.11

relations such that both of them can be optimized in a collaborative manner to aid each other, resulting in improved performance. We conducted statistical significance estimates using McNemar's paired tests and our models were found to be statistically significantly better (p -value < 0.05 with a 95% confidence interval). The improvement demonstrates the merits of our models.

As shown in Table 7.6, our **joint model** obtains the best performance on 4 entity categories, and the **bidirectional model** obtains the best performance on 2 categories (*year* and *miscellaneous*). The **CRF+CRF** and **CRF+MLN** obtain the best performance on *date*, and **Single MLN** obtains the best performance on *number*. However, for the three baseline models, the performance on these entities is slightly higher than that of our models. Unlike the Chinese NER task, it is particularly interesting that the **bidirectional model** outperforms the **integrated model** on both entity identification (93.96 vs. 93.36 on the overall F-score) and relation extraction (68.27 vs. 68.15). This is due to the bidirectional nature of this model (e.g., use joint factors to connect variables for entities and entity relations, perform collaborative parameter estimation such that entities and entity relations can help each other to boost the performance).

7.2.7 Performance of Relation Extraction

For relation extraction in Table 7.7 and Table 7.8, the performance varies widely for different relation types. All the systems perform quite well on *death_day*, *death_year*, *birth_day* and *birth_year*. This is because these relations are generally more distinctive than other types and can be easily extracted. Another reason is that, these relations are closely related to entities *date* and *year*, which can also be well recognized using contextual evidences for all models. Consequently, relation extraction benefits from good entity identification results. However, some relation types (e.g., *aunt* and *discovered*) can hardly be extracted. 19 relation types cannot be extracted by all models. This may be due to the lack of training data since these relations occur rarely in the dataset. For the 34 relation types listed in Table 7.7 and Table 7.8, the **CRF+CRF** model obtains the highest F-measure on 2 relation types, **CRF+MLN** obtains the highest F-measure on 2, **Single MLN** obtains the highest F-measure on 3. For our proposed models, the **bidirectional model**, **integrated model**, and the **joint model** achieves the highest F-measure on 8, 3, and 16 relation types, respectively. Compared to entity identification, our models perform much worse on relation extraction task (e.g., 94.11 vs. 68.59 for the **joint model**), which shows that accurately extracting relations between entities is still a difficult and open problem for future research.

7.2.8 Analysis and Discussion

Our proposed **joint model** is superior to the pipeline models **CRF+CRF** and **CRF+MLN** by modeling segmentations in sequence data for entity identification and relations of different segments for relation extraction jointly. The **CRF+CRF** model performs relation extraction sequentially without considering connections between entities. It cannot capture long-distance dependencies and may cause the label consistency problem. These disadvantages limit the ability of CRFs for relation extraction to a large extent. The **CRF+MLN** model can alleviate some of these problems by modeling relations between entities via first-order logic, however, it does not consider the mutual interaction or correlation between entities and relations. As pipeline models, **CRF+CRF** or **CRF+MLN** cannot correctly extract relations between mis-recognized entities from the CRF. For example, in our experiments, both of the two models cannot extract the *member_of* relation between the secondary entity *Republican* and the principal entity *George W. Bush*, since the organization name *Republican* is incorrectly labeled as a *miscellaneous*. Since knowing the secondary entities is helpful for their re-

lations to the principal entity and vice versa, modeling both simultaneously is highly desirable. Our proposed **joint model** can correctly label the organization name *Republican* and predict the *member_of* relation to *George W. Bush*. This modeling can incorporate rich dependencies between entities and relations, it can also exploit relational autocorrelation, a widely observed characteristic of relational data in which the value of a variable for one instance is highly correlated with the value of the same variable on another instance.

Our **joint model** combines the advantages of both probabilistic graphical models for sequence data uncertainty modeling, and a variety of first-order logic for domain knowledge. The efficiency of the purely probabilistic graphical model **CRF+CRF** heavily depends on its first-order Markov property, which is important for sequence modeling. However, our compound task requires expressive and deeper knowledge representation which **CRF+CRF** cannot handle well. As illustrated in Table 7.7, the **CRF+CRF** model performs very poorly on 5 relations: *friend*, *sister*, *grandfather*, *grandson*, and *cousin*, resulting in reduced overall F-measure. Since these relations are likely to have significant impact on performance and they require higher-level domain knowledge to be extracted. The **Single MLN** model, on the other hand, can compactly represent a wide variety of knowledge via first-order logic. This model captures correlations between entity relations, and it outperforms **CRF+CRF** and **CRF+MLN** models using joint inference. However, the power of MLN alone for modeling sequence data is limited. Limitations of first-order logic make it difficult to specify a relation factor that uses the uncertain output of segmentation [77]. 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. By modeling segmentations and relations between segments simultaneously, our model strengthens the mutual interactions between entities and relations. Note that the structured variational inference algorithm captures interactions between substructures in our model. Thus, deep interactions between entities and relations are well modeled, and they are optimized properly.

7.2.9 Comparison with Other Methods

Table 7.9 compares our results with some recently published results on the same dataset. Notably, our approaches outperform 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 models are more applicable to real-world IE tasks.

Culotta et al. [21] proposed a probabilistic model based on CRFs to integrate extraction and data mining tasks performed on biographical Wikipedia articles. Relation extraction was treated as a sequence labeling problem and relational patterns were discovered to boost the performance. However, this model extracts relations without considering dependencies between entities, and the best reported F-measure is 67.91, which is significantly lower than our systems when evaluated on the same training and testing sets.

Nguyen et al. [58] proposed a subtree mining approach to extracting relations from Wikipedia by incorporating information from the Wikipedia structure and by the analysis of Wikipedia text. In this approach, a syntactic tree that reflects the relation between a given entity pair was built, and a tree-mining algorithm was used to identify the basic elements of syntactic structure of sentences for relations. This approach mainly relies on syntactic structures to extract relations. Syntactic structures are important for relation extraction, but insufficient to extract relations accurately. The obtained F-measure was only 37.76, which shows that there is a large room for improving.

We mention some other related work. [10] presented an approach to extract relations from the Web using minimal supervision. [71] presented a method for improving semi-supervised relation extraction from the Web using corpus statistics on entities. Our work is different from these research work. We investigate supervised joint IE tasks based on probabilistic graphical models.

7.2.10 Bidirectionality

We also examine the effectiveness of our proposed bi-directional MH inference algorithm in the **joint model** and Figure 7.4 demonstrates its feasibility by comparing it with the greedy, N -best list, and uni-directional MH sampling algorithms. The detailed description of these algorithms is presented in Subsection 7.1.4. It shows that the bi-directional MH algorithm consistently outperforms other algorithms on both entity identification and relation extraction tasks from Wikipedia. The greedy algorithm is very simple, but it only makes use of 1-best list of segmentations and corresponding relations, losing much useful information. This algorithm produces the worst performance: 93.36 F_1 on entity identification and 67.76 F_1 on relation extraction. The N -best list gives useful improvements over the greedy. We set $N = 20$ according to the holdout methodology for our model. 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 the model. The uni-

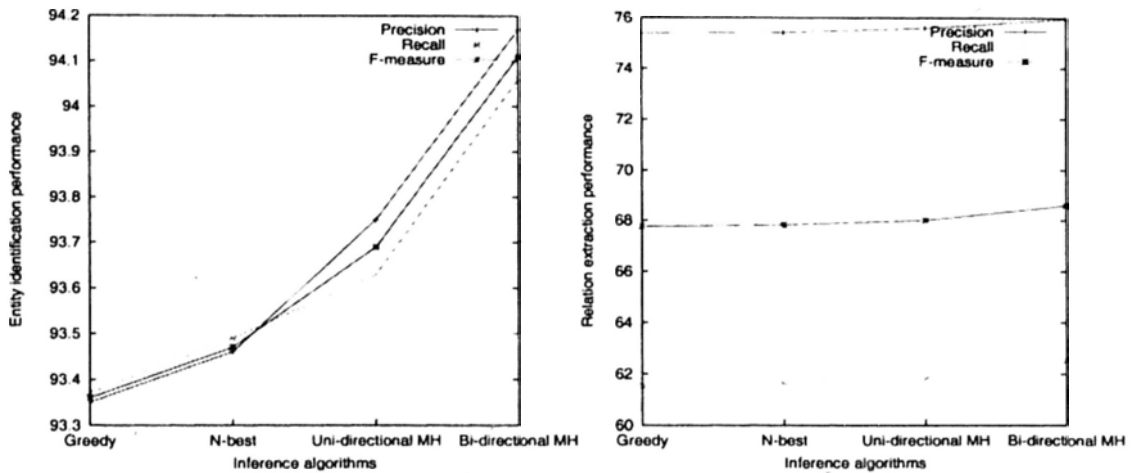


Figure 7.4: Performance comparison of different inference algorithms on entity identification (left) and relation extraction (right) from Wikipedia.

directional MH algorithm outperforms N -best list when enough samples (we set sample size = 10000) are drawn, since sampling 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. The bi-directional MH algorithm achieves the highest performance: 94.11 F_1 on entity identification and 68.59 F_1 on relation extraction. It is a bi-directional highly-coupled joint inference and enforces transitivity, thus mutual benefits from both segmentations and relations can be well captured.

7.3 Citation Matching

7.3.1 Task Description

Consider the task of citation matching in which we are given a large collection of citation strings from the “References” section of research papers. They may have different citation styles, different abbreviations, and typographical errors. Many of the citations refer to the same underlying papers. Our job is to identify the AUTHOR, TITLE, and VENUE fields of each citation (segmentation) and also find the citations referring to the same paper (coreference or entity resolution). As shown in Figure 7.5 which contains 3 citations, **C1** and **C2** are coreferent since they refer to the same paper. **C1** and **C3**, **C2** and **C3** are not coreferent.

<p>C1: [Parag Singla]_{AUTHOR} and [Pedro Domingos]_{AUTHOR}, “[Memory-Efficient Inference in Relational Domains]_{TITLE}” ([AAAI-06]_{VENUE}).</p> <p>C2: [Singla, P.]_{AUTHOR}, & [Domingos, P.]_{AUTHOR} (2006). [Memory-efficient inference in relational domains]_{TITLE}. In [Proceedings of the Twenty-First National Conference on Artificial Intelligence]_{VENUE} (pp. 500-505). Boston, MA: AAAI Press.</p> <p>C3: [H. Poon]_{AUTHOR} & [P. Domingos]_{AUTHOR}, [Sound and Efficient Inference with Probabilistic and Deterministic Dependencies]_{TITLE}, in [Proc. AAAI-06]_{VENUE}, Boston, MA, 2006.</p>
--

Figure 7.5: An example of citation matching. The notations [_{AUTHOR}], [_{TITLE}], and [_{VENUE}] denote that the fields are author, title, and venue, respectively. **C1** and **C2** are coreferent, **C1** and **C3**, **C2** and **C3** are not coreferent.

7.3.2 Data and Methodology

We use the Cora dataset to evaluate our proposed models. This dataset contains 1295 citations and 134 clusters (sets of citations that refer to the same paper), and each citation has three fields – *author*, *title*, and *venue*. The dataset is divided into the same three folds as in [67] such that they are distributed as evenly as possible and no clusters are split across different folds. We also set the iteration number T to 10000 for the bi-directional MH inference algorithm. We run three-fold cross-validation on this dataset. Segmentation is evaluated by P , R , and F_1 . For entity resolution, we measure both pairwise P , R , F_1 and cluster recall, which is the fraction of clusters that are correctly predicted.

Accurate segmentation enables features that are naturally expected to be useful to boost coreference. A wide range of rich, overlapping features can be exploited in our models. These features largely consider field-level similarity using a number of string and token-based comparison metrics (e.g., string edit distance, tfidf over tokens and n-grams, etc). We also include feature conjunctions, specialized features for *author* and *title* fields matching, and global features based on distance metrics for entire citations. In leveraging coreference to improve segmentation, we use a combination of local (e.g., contextual and morphological), layout, lexicon membership features.

For performance comparison, the **CRF+CRF** model uses first CRF for segmentation, and another CRF for resolution. The **CRF+MLN** uses MLN

for resolution. For **Single MLN**, we follow [67] to design it, using features mentioned above. In addition, we also compare the performance of our models with some recently published results on the same dataset.

7.3.3 Experimental Results and Analysis

Our experimental results on Cora are shown in Table 7.10 and Table 7.11, demonstrating the promise of our approaches with significant improvements on both segmentation and coreference, comparing with the three baseline models **CRF+CRF**, **CRF+MLN**, **Single MLN** and other previously published results. All improvements of our proposed models over the three baseline models are statistically significant using McNemar’s paired tests.

Table 7.10 shows improvements on F-measure for the segmentation task. We list both the overall performance and the performance on the three fields. Our **bidirectional model** obtains the best overall performance (98.66 on the F-score), and it outperforms earlier results, namely, **Isolated MLN** [67] and **Single MLN** [67], providing an overall relative error reduction in F_1 of 25.56% and 16.25%. Compared to our three baseline models, the error reduction is 42.74%, 42.74%, and 15.72% respectively. The performance of the **joint model** is slightly worse (by 0.03 on the F-score) than that of the **bidirectional model**. Note that the difference between our **Single MLN** model and the one in [67] is that we used different features. Although the performance of the **integrated model** is worse than **Isolated MLN** and **Single MLN**, it still performs reasonably good and outperforms the baseline systems such as **CRF+CRF** and **CRF+MLN**.

Table 7.11 compares the performance of entity resolution for different models on both metrics. Our **joint model**, which concurrently solves the citation matching task, easily outperforms previously published results in [79] and [67]. It also outperforms our three baseline models by 3.34%, 1.61%, and 1.19% in pairwise F_1 . Even though the **Single MLN** model in [67] captures interactions between segmentation and coreference, it is only a weak interaction. First, the logic formulas in [67] only examine pairs of consecutive labels, not whole fields – failing to use information from predicted field range and non-consecutive words in the field. Second, the frequency with which the **JntInfCandidate** feature appears is quite data-dependent. If the feature occurs too often, it can be harmful for coreference. As can be seen, our **joint model** achieves stronger interaction between tasks, leading to improved performance for citation matching. In addition, Table 7.11 shows that our joint approach allows cluster recall to improve substantially, resulting in an improvement of up to 9.44% compared to our **Single MLN** model. This is particularly notable given that cluster recall is more strict than the pairwise

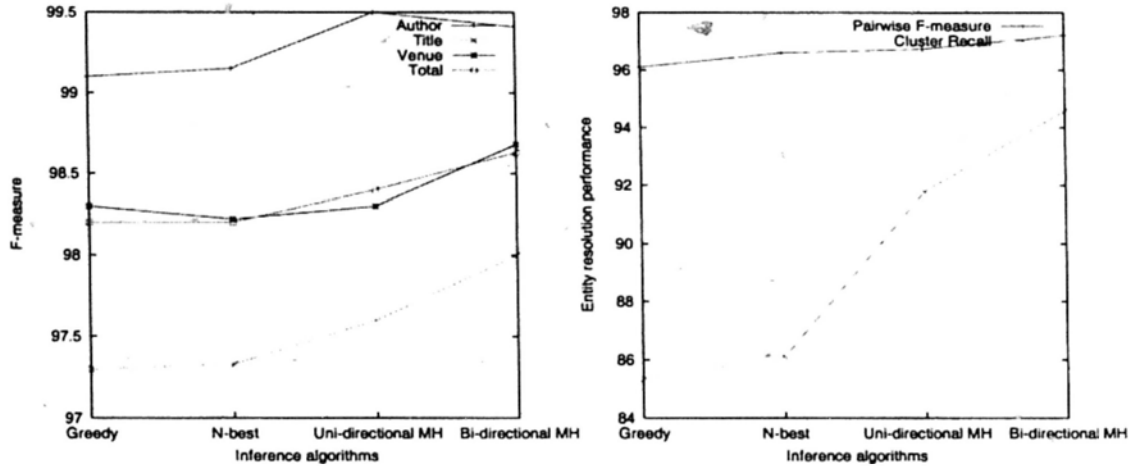


Figure 7.6: Performance comparison of different inference algorithms on segmentation (left) and entity resolution (right) for citation matching.

F_1 metric.

7.3.4 Bidirectionality

Figure 7.6 illustrates the benefits of the bi-directional MH inference algorithm over the greedy, N -best list, and uni-directional MH algorithms for citation matching. We set $N = 20$ for the N -best list and sample size to be 10000 for the uni-directional MH algorithm. For segmentation task, we compare both the overall F-measure and F-measures on the three fields for these algorithms. As shown in Figure 7.6, the bi-directional MH algorithm obtains the best performance on the fields *title* and *venue*. For *author*, its performance is slightly worse than that of uni-directional MH (99.41 vs. 99.5). For the overall performance, the greedy algorithm obtains the lowest F_1 of 98.2 and the N -best list can hardly improve it. The uni-directional MH slightly improves to 98.4, and the bi-directional MH algorithm enhances this number to 98.63 further. For entity resolution task, we compare both the pairwise F-measure and cluster recall metrics, and the bi-directional MH algorithm achieves the highest performance. Compared to the pairwise F-measure, the cluster recall is boosted substantially by this algorithm. This is particularly interesting and it shows that the bi-directional MH algorithm is much more accurate than the other three under the strict metric. This figure demonstrates the bi-directionality of our inference algorithm using segmentation to aid coreference and vice-versa, which is highly coupled and strong interactions between segmentation and resolution can be achieved.

Table 7.7: Performance of the CRF+CRF, CRF+MLN, and Single MLN models for relation extraction from Wikipedia. Here, poli_aff and ass_comp denote political affiliation and associate_competition, respectively.

Relations	CRF+CRF			CRF+MLN			Single MLN		
	P	R	F ₁	P	R	F ₁	P	R	F ₁
death_day	98.15	92.98	95.50	98.22	93.24	95.67	98.18	94.74	96.43
death_year	98.15	91.38	94.64	98.18	91.55	94.75	98.18	93.10	95.58
birth_year	93.75	91.46	92.59	93.50	91.23	92.35	93.65	91.56	92.59
birth_day	92.68	93.83	93.25	91.39	93.67	92.52	91.57	93.83	92.68
nationality	79.55	87.50	83.33	79.00	84.87	81.83	79.07	85.00	81.93
birth_place	79.76	78.82	79.29	79.03	80.22	79.62	79.17	89.41	83.98
job_title	85.71	87.27	86.49	85.33	87.98	86.63	85.09	88.18	86.61
death_place	93.33	66.67	77.78	88.91	65.35	75.33	86.67	61.90	72.22
education	72.73	85.11	78.43	71.82	83.30	77.14	70.91	82.98	76.47
father	56.67	68.00	61.82	63.77	72.98	68.06	67.86	76.00	71.70
wife	76.47	81.25	78.79	77.33	83.49	80.29	76.92	62.50	68.97
award	85.00	65.38	73.91	83.33	63.45	72.04	82.35	53.85	65.12
mother	100.0	14.29	25.00	100.0	37.76	54.82	100.0	42.86	60.00
poli_aff	100.0	53.33	69.57	100.0	57.67	73.15	100.0	58.66	73.94
husband	83.33	50.00	62.50	83.33	50.00	62.50	83.33	50.00	62.50
visited	59.57	52.34	55.72	52.37	58.22	55.14	52.17	56.07	54.05
daughter	83.33	45.45	58.82	71.00	51.89	59.96	71.43	45.45	55.56
founder	90.00	47.37	62.07	71.17	52.39	60.35	77.78	36.84	50.00
member_of	66.07	52.11	58.27	66.27	52.09	58.33	52.31	47.89	50.00
executive	59.09	36.11	44.83	57.48	37.17	45.15	56.52	36.11	44.07
superior	75.00	31.58	44.44	63.88	34.98	45.21	63.64	36.84	46.67
brother	46.15	40.00	42.86	46.15	40.67	43.24	53.33	53.33	53.33
opus	79.55	34.31	47.95	73.88	39.19	51.21	72.41	41.18	52.50
son	61.54	34.78	44.44	70.09	34.40	46.15	80.00	34.43	48.14
associate	47.33	53.91	50.41	45.88	51.87	48.69	37.21	55.65	44.60
participant	66.67	38.10	48.48	65.42	42.33	51.40	52.94	42.86	47.37
employer	64.29	27.27	38.30	65.17	27.56	38.74	66.67	18.18	28.57
ass_comp	60.00	40.00	48.00	65.43	52.10	58.01	33.33	20.00	25.00
religion	100.0	8.33	15.38	100.0	8.67	15.96	100.0	10.33	18.73
friend	0	0	0	38.67	20.28	26.61	40.09	23.33	29.50
sister	0	0	0	25.67	9.78	14.16	24.17	12.50	16.48
grandfather	0	0	0	16.67	9.33	11.96	16.67	16.67	16.67
grandson	0	0	0	0	0	0	18.00	7.76	10.84
cousin	0	0	0	0	0	0	11.67	7.67	9.26
other types	0	0	0	0	0	0	0	0	0
Overall	70.40	57.85	63.51	69.39	59.53	64.08	68.54	61.75	64.97

Table 7.8: Performance of the bidirectional model, the integrated, and joint models for relation extraction from Wikipedia. Here, poli_aff and ass_comp denote political affiliation and associate competition, respectively.

Relations	Bidirectional			Integrated			Joint		
	P	R	F ₁	P	R	F ₁	P	R	F ₁
death_day	98.20	94.85	96.50	98.10	94.70	96.37	100.0	94.55	97.20
death_year	96.30	94.60	95.44	96.10	94.38	95.23	96.43	93.20	94.79
birth_year	95.06	93.90	94.48	94.88	93.55	94.21	94.94	91.66	93.27
birth_day	93.98	96.30	95.12	93.77	96.12	94.93	94.47	96.50	95.47
nationality	88.64	97.50	92.86	88.50	96.22	92.20	88.77	95.22	91.88
birth_place	88.12	88.97	88.54	88.37	89.41	88.89	87.60	86.28	86.93
job_title	86.73	89.09	87.89	86.73	89.09	87.89	87.95	88.48	88.21
death_place	93.75	71.43	81.08	93.90	72.00	81.50	94.12	77.90	85.25
education	69.49	87.23	77.36	70.00	87.25	77.68	73.23	87.25	79.63
father	75.00	84.00	79.25	75.00	83.25	78.91	75.00	84.00	79.25
wife	72.22	81.25	76.47	72.33	81.00	76.42	72.89	82.25	77.29
award	94.44	65.38	77.27	94.00	66.45	77.86	94.44	68.30	79.27
mother	86.00	40.33	54.91	85.77	41.56	55.99	85.71	42.86	57.14
poli_aff	100.0	60.00	75.00	100.0	46.67	63.64	87.80	48.87	62.79
husband	85.71	60.00	70.59	84.98	60.00	70.34	85.71	62.00	71.95
visited	66.27	51.40	57.89	66.18	51.33	57.82	68.75	51.40	58.82
daughter	76.67	63.33	69.36	77.78	63.64	70.00	70.00	64.88	67.34
founder	80.33	40.57	53.91	81.82	47.37	60.00	81.82	47.37	60.00
member_of	52.54	43.66	47.69	52.39	43.33	47.43	63.97	57.89	60.78
executive	68.00	47.22	55.74	68.87	47.33	56.10	68.66	48.37	56.76
superior	72.73	42.11	53.33	65.00	42.34	51.28	63.54	42.95	51.25
brother	60.00	60.00	60.00	53.33	53.33	53.33	46.85	40.00	43.07
opus	68.75	21.57	32.84	67.00	20.49	31.38	69.70	52.55	59.92
son	52.94	39.13	45.00	56.67	36.74	44.58	57.75	34.98	43.57
associate	48.33	46.89	47.60	45.61	45.22	45.41	43.96	43.82	43.89
participant	45.45	23.81	31.25	45.67	22.33	29.99	44.44	19.05	26.67
employer	50.00	24.24	32.65	50.00	26.78	34.88	53.63	44.84	48.84
ass_comp	66.93	53.21	59.29	50.00	46.67	48.28	40.00	40.00	40.00
religion	80.00	12.33	21.37	75.00	12.00	20.69	70.00	16.67	26.93
friend	50.00	32.33	39.27	47.76	32.33	38.56	45.50	38.90	41.94
sister	27.87	16.33	20.59	30.00	16.00	20.87	26.90	16.67	20.58
grandfather	100.0	14.29	25.00	30.00	15.50	20.44	20.50	16.67	18.39
grandson	20.00	16.67	18.18	20.00	15.00	17.14	25.00	15.33	19.00
cousin	15.00	7.67	10.15	12.00	7.67	9.36	15.00	9.50	11.63
other types	0	0	0	0	0	0	0	0	0
Overall	72.89	64.20	68.27	75.29	62.25	68.15	75.95	62.53	68.59

Table 7.9: Performance comparison with other systems on relation extraction.

System	Precision	Recall	F-measure
Culotta et al. [21]	75.53	61.69	67.91
Nguyen et al. [58]	29.07	53.86	37.76
Bidirectional	72.89	64.20	68.27
Integrated	75.29	62.25	68.15
Joint	75.95	62.53	68.59

Table 7.10: Comparative performance of different models for segmentation in citation matching.

Method	Author	Title	Venue	Total
Isolated MLN [67]	99.30	97.30	98.20	98.20
Single MLN [67]	99.50	97.60	98.30	98.40
CRF+CRF	98.77	97.02	97.56	97.66
CRF+MLN	98.77	97.02	97.56	97.66
Single MLN	99.39	97.79	98.36	98.41
Bidirectional	99.45	98.00	98.70	98.66
Integrated	98.90	97.12	97.67	97.71
Joint	99.41	98.00	98.68	98.63

Table 7.11: Comparative performance of different models for entity resolution in citation matching.

Method	P	R	F_1	Cluster Recall
Fellegi-Sunter [79]	78.00	97.70	86.70	62.70
Single MLN [67]	94.30	97.00	95.60	78.10
CRF+CRF	93.10	94.65	93.87	76.32
CRF+MLN	94.14	97.11	95.60	78.89
Single MLN	94.84	97.22	96.02	85.15
Bidirectional	95.10	97.58	96.32	85.78
Integrated	94.24	97.31	95.75	82.33
Joint	96.20	98.25	97.21	94.59

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis presents several statistical frameworks, including the cascaded model, the bidirectional integrated model, the integrated discriminative probabilistic model, and the joint model incorporating first-order logic for the problem of joint information extraction, which is generally very challenging and promising. Fundamentally, we rely on the uncertainty power of undirected, conditionally-trained probabilistic graphical models for sequence data modeling as well as the expressiveness of first-order logic formalism for deep and relational domain knowledge representation which is essential for higher-level IE tasks as we investigated. We investigate exact algorithms as well as approximate techniques where exact inference is intractable in some of these models, resulting in several efficient learning and inference formulations. For example, we propose the collaborative parameter estimation based on the L-BFGS algorithm for bidirectional model learning. We propose a structured variational inference algorithm for tractable and approximate parameter estimation for the joint model, which exploits substructures and captures interactions between them. And we propose a strongly-coupled, bidirectional MH sampling algorithm to enable efficient inference to find the MAP assignments for all the subtasks in the joint model, such that inference information can flow in both directions and mutual benefits between different subtasks can be well exploited.

As shown in this thesis, our approaches has several theoretical and practical advantages over standard state-of-the-art probabilistic models, offering a natural way for joint information extraction tasks. The cascaded framework considers entity extraction and qualitative domain knowledge. This architecture captures a variety of linguistic characteristics in Chinese NEs as domain

knowledge which can be easily and concisely formulated via first-order logic. The bidirectional framework is highlighted by introducing joint factors to connect variables of relevant subtasks capturing tight interactions between them, such that evidences from multiple subtasks can be shared and they aid each other to enhance the performance. The integrated framework combines the advantages of both probabilistic graphical models for sequence data and first-order logic in a principled way. We emphasize its capability of mining implicit relations and new relation discovery, and capturing sub-structures in named entities. The joint model defines a joint probability distribution for both segmentations in sequence data and possible worlds of relations between segments in the form of an exponential family, to optimize all relevant subtasks simultaneously.

We develop theoretical foundations for our approach and show a wide range of experimental applications, including Chinese NER, entity identification and relationship extraction from Wikipedia’s encyclopedic articles, and citation matching. Extensive experimental study on real-world datasets demonstrates the feasibility and effectiveness of our approaches. We analyze and discuss potential merits and advantages of our proposed models. In addition, some interesting issues, such as the superiority of the bidirectional MH inference algorithm, are also presented.

8.2 Future Work

Our proposed models allow extensive further investigation, both for parameter learning and inference algorithms. And there are several applications, extensions, and open problems for our estimated frameworks. We organize and list directions for future work in the following main areas:

- We plan to improve the scalability of our approaches and apply them to other large-scale real-world problems, especially for some novel tasks.
- We plan to conduct further theoretical analysis of these models, and propose new optimization and/or inference algorithms.
- We plan to extend our approaches to more general learning settings, such as semi-supervised or unsupervised learning.

Bibliography

- [1] The Automatic Content Extraction (ACE) Evaluation. <http://www.nist.gov/speech/tests/ace/>.
- [2] Message Understanding Conference: Systems used on Named Entity Tasks. http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_toc.html#named.
- [3] The shared task on the seventh Computational Natural Language Learning (CoNLL-2003). <http://www.cnts.ua.ac.be/conll2003/ner/>.
- [4] The shared task on the sixth Computational Natural Language Learning (CoNLL-2002). <http://www.cnts.ua.ac.be/conll2002/ner/>.
- [5] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–236, 1974.
- [6] Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what’s in a name. *Machine Learning*, 34(1-3):211–231, February 1999.
- [7] Andrew Borthwick. *A Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University, September 1999.

- [8] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.
- [9] Razvan Bunescu and Raymond J. Mooney. Collective information extraction with relational Markov networks. In *Proceedings of ACL-04*, Barcelona, Spain, 2004.
- [10] Razvan C. Bunescu and Raymond J. Mooney. Learning to extract relations from the Web using minimal supervision. In *Proceedings of ACL-07*, pages 576–583, Prague, Czech Republic, June 2007.
- [11] Peter Carbonetto, Jacek Kiszyński, O De Freitas, and David Poole. Nonparametric Bayesian logic. In *Proceedings of UAI-05*, pages 85–93, 2005.
- [12] Ki Chan, Wai Lam, and Xiaofeng Yu. Coreference resolution using expressive logic models. In *Proceedings of CIKM-08*, pages 1373–1374, Napa Valley, California, USA, 2008.
- [13] Shing-Kit Chan, Wai Lam, and Xiaofeng Yu. A cascaded approach to biomedical named entity recognition using a unified model. In *Proceedings of ICDM-07*, pages 93–102, Omaha NE, USA, 2007.
- [14] Shing-Kit Chan, Wai Lam, and Xiaofeng Yu. An online cascaded approach to biomedical named entity recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-08)*, pages 595–600, Hyderabad, India, 2008.
- [15] Aitao Chen, Fuchun Peng, Roy Shan, and Gordon Sun. Chinese named entity recognition with conditional probabilistic models. In *5th*

SIGHAN Workshop on Chinese Language Processing, Australia, July 2006.

- [16] Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. Chinese named entity recognition with conditional random fields. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.
- [17] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL-03*, 2003.
- [18] William W. Cohen and Sunita Sarawagi. Exploiting dictionaries in named entity extraction: Combining semi-Markov extraction processes and data integration methods. In *Proceedings of ACM-SIGKDD 2004*, 2004.
- [19] Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39:80–91, 1996.
- [20] Aron Culotta and Andrew McCallum. Confidence estimation for information extraction. In *Proceedings of HLT/NAACL-04*, pages 109–112, 2004.
- [21] Aron Culotta, Andrew McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of HLT/NAACL-06*, pages 296–303, New York, 2006.
- [22] Aron Culotta, Michael Wick, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *Proceedings of HLT/NAACL-07*, pages 81–88, 2007.

- [23] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. A survey of first-order probabilistic models. In *Innovations in Bayesian Networks*, pages 289–317. 2008.
- [24] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of ACL-05*, pages 363–370, Ann Arbor, Michigan, 2005.
- [25] Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of EMNLP-06*, pages 618–626, Sydney, Australia, 2006.
- [26] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *Proceedings of IJCAI-99*, pages 1300–1309, 1999.
- [27] Guohong Fu and Kang-Kwong Luke. Chinese named entity recognition using lexicalized HMMs. *ACM SIGKDD Explorations Newsletter*, 7:19–25, June 2005.
- [28] Michael R. Genesereth and Nils J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1987.
- [29] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, UK, 1996.
- [30] L. Gillick and Stephen Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of ICASSP-89*, pages 532–535, 1989.

- [31] Asela Gunawardana, Milind Mahajan, Alex Acero, and John C. Platt. Hidden conditional random fields for phone classification. In *Proceedings of INTERSPEECH-05*, pages 1117–1120, 2005.
- [32] Joseph Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.
- [33] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [34] Kristy Hollingshead and Brian Roark. Pipeline iteration. In *Proceedings of ACL-07*, pages 952–959, Prague, Czech Republic, 2007.
- [35] Tuyen N. Huynh and Raymond J. Mooney. Discriminative structure and parameter learning for Markov logic networks. In *Proceedings of ICML-08*, pages 416–423, 2008.
- [36] Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of COLING-02*, pages 1–7, Taipei, Taiwan, 2002.
- [37] Tommi S. Jaakkola. Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, pages 129–159. MIT Press, 2000.
- [38] Manfred Jaeger. Relational Bayesian networks. In *Proceedings of UAI-97*, pages 266–273, 1997.
- [39] Guangjin Jin and Xiao Chen. The fourth international Chinese language processing bakeoff: Chinese word segmentation, named entity recognition and Chinese POS tagging. In *Proceedings of Sixth SIGHAN Workshop on Chinese Language Processing*, pages 69–81, 2008.

- [40] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical methods. *Machine Learning*, 37:183–233, 1999.
- [41] Kristian Kersting and Luc De Raedt. Bayesian logic programming: Theory and tool. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [42] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [43] Stanley Kok, Parag Singla, Matthew Richardson, and Pedro Domingos. The Alchemy system for statistical relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2005. <http://www.cs.washington.edu/ai/alchemy>.
- [44] Terry Koo and Michael Collins. Hidden-variable models for discriminative reranking. In *Proceedings of HLT/EMNLP-05*, pages 507–514, Vancouver, British Columbia, Canada, 2005.
- [45] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.
- [46] Taku Kudo. CRF++: Yet another CRF tool kit. <http://crfpp.sourceforge.net/>, 2005.
- [47] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*, pages 282–289, 2001.

- [48] Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26:119–134, 2007.
- [49] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.
- [50] Ping Luo, Fen Lin, Yuhong Xiong, Yong Zhao, and Zhongzhi Shi. Towards combining Web classification and Web information extraction: a case study. In *Proceedings of KDD-09*, pages 1235–1244, Paris, France, 2009.
- [51] Milind Mahajan, Asela Gunawardana, and Alex Acero. Training algorithms for hidden conditional random fields. In *Proceedings of ICASSP-06*, pages 273–276, 2006.
- [52] Enzo Marinari and Giorgio Parisi. Simulated Tempering: A new Monte Carlo scheme. *Europhysics Letters*, 19:451–458, 1992.
- [53] Andrew McCallum and David Jensen. A note on the unification of information extraction and data mining using conditional-probability, relational models. In *Proceedings of the IJCAI-03 Workshop on Learning Statistical Models from Relational Data*, 2003.
- [54] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL-03*, 2003.
- [55] Andrew McCallum and Ben Wellner. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of NIPS-03*, pages 905–912. MIT Press, 2003.

- [56] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [57] Brian Milch, Bhaskara Marthi, and Stuart Russell. BLOG: Relational modeling with unknown objects. In *Proceedings of the ICML-04 Workshop on Statistical Relational Learning and its Connections to Other Fields*, pages 67–73, Banff, Canada, 2004.
- [58] Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from Wikipedia using subtree mining. In *Proceedings of AAAI-07*, pages 1414–1420, Vancouver, British Columbia, Canada, 2007.
- [59] Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. Identity uncertainty and citation matching. In *Proceedings of NIPS-03*, pages 1401–1408, Washington, DC, 2003.
- [60] Hanna Pasula and Stuart Russell. Approximate inference for first-order probabilistic languages. In *Proceedings of IJCAI-01*, pages 741–748, 2001.
- [61] Judea Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [62] Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of COLING-04*, pages 562–568, 2004.

- [63] Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *Proceedings of HLT-NAACL 2004*, pages 329–336, 2004.
- [64] David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft. Table extraction using conditional random fields. In *Proceedings of ACM SIGIR-03*, pages 235–242, 2003.
- [65] Thierry Poibeau and Leila Kosseim. Proper name extraction from non-journalistic texts. *Computational Linguistics in the Netherlands*, 14:144–157, 2001.
- [66] Hoifung Poon and Pedro Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proceedings of AAAI-06*, Boston, Massachusetts, July 2006. The AAAI Press.
- [67] Hoifung Poon and Pedro Domingos. Joint inference in information extraction. In *Proceedings of AAAI-07*, pages 913–918, Vancouver, British Columbia, Canada, 2007.
- [68] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [69] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.
- [70] Sebastian Riedel and Ewan Klein. Genic interaction extraction with semantic and syntactic chains. In *Proceedings of the Learning Language in Logic Workshop (LLL-05)*, pages 69–74, 2005.

- [71] Benjamin Rosenfeld and Ronen Feldman. Using corpus statistics on entities to improve semi-supervised relation extraction from the Web. In *Proceedings of ACL-07*, pages 600–607, Prague, Czech Republic, June 2007.
- [72] Sunita Sarawagi and William W. Cohen. Semi-Markov conditional random fields for information extraction. In *Proceedings of NIPS-04*, 2004.
- [73] Lawrence K. Saul and Michael I. Jordan. Exploiting tractable substructures in intractable networks. In *Proceedings of NIPS-96*, pages 486–492, Cambridge, MA, 1996.
- [74] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, Geneva, Switzerland, 2004.
- [75] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of HLT/NAACL-03*, pages 213–220, 2003.
- [76] Yanxin Shi and Mengqiu Wang. A dual-layer CRFs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of IJCAI-07*, pages 1707–1712, Hyderabad, India, 2007.
- [77] Sameer Singh, Karl Schultz, and Andrew McCallum. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Proceedings of ECML/PKDD-09*, pages 414–429, Bled, Slovenia, 2009.
- [78] Parag Singla and Pedro Domingos. Discriminative training of Markov logic networks. In *Proceedings of AAAI-05*, pages 868–873, 2005.

- [79] Parag Singla and Pedro Domingos. Entity resolution with Markov logic. In *Proceedings of ICDM-06*, pages 572–582, 2006.
- [80] Lucia Specia, Mark Stevenson, and Maria das Graças V. Nunes. Learning expressive models for word sense disambiguation. In *Proceedings of ACL-07*, pages 41–48, Prague, Czech Republic, 2007.
- [81] Maosong Sun, Changning Huang, Haiyan Gao, and Jie Fang. Identifying Chinese names in unrestricted texts. *Journal of Chinese Information Processing*, 1995.
- [82] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.
- [83] Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8:693–723, 2007.
- [84] Charles Sutton, Khashayar Rohanimanesh, and Andrew McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of ICML-04*, Banff, Alberta, Canada, 2004.
- [85] Ben Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Proceedings of UAI-02*, pages 485–492, 2002.

- [86] Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. Joint learning improves semantic role labeling. In *Proceedings of ACL-05*, pages 589–596, 2005.
- [87] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.
- [88] Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of UAI-04*, pages 593–601, Banff, Canada, 2004.
- [89] Wim Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings of UAI-2000*, pages 626–633, San Francisco, CA, 2000.
- [90] Youzheng Wu, Jun Zhao, Bo Xu, and Hao Yu. Chinese named entity recognition based on multiple features. In *Proceedings of HLT-EMNLP 2005*, 2005.
- [91] Eric P. Xing, Michael I. Jordan, and Stuart Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of UAI-03*, pages 583–591, 2003.
- [92] Eric P. Xing, Michael I. Jordan, and Stuart Russell. Graph partition strategies for generalized mean field inference. In *Proceedings of UAI-04*, pages 602–610, 2004.
- [93] Chunyu Yang, Yong Cao, Zaiqing Nie, Jie Zhou, and Ji-Rong Wen. Closing the loop in Webpage understanding. *IEEE Transactions on Knowledge and Data Engineering*, Forthcoming, 2010.

- [94] Xiaofeng Yu. Chinese named entity recognition with cascaded hybrid model. In *Proceedings of HLT/NAACL-07*, pages 197–200, Rochester, New York, 2007.
- [95] Xiaofeng Yu, Marine Carpuat, and Dekai Wu. Boosting for Chinese named entity recognition. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.
- [96] Xiaofeng Yu and Wai Lam. Hidden dynamic probabilistic models for labeling sequence data. In *Proceedings of AAAI-08*, pages 739–745, Chicago, Illinois, 2008.
- [97] Xiaofeng Yu and Wai Lam. An integrated probabilistic and logic approach to encyclopedia relation extraction with multiple features. In *Proceedings of COLING-08*, pages 1065–1072, Manchester, United Kingdom, 2008.
- [98] Xiaofeng Yu and Wai Lam. Bidirectional integration of pipeline models. In *Proceedings of AAAI-10*, 2010. To appear.
- [99] Xiaofeng Yu and Wai Lam. Probabilistic joint models incorporating logic and learning via structured variational approximation for information extraction. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2010. Under review.
- [100] Xiaofeng Yu, Wai Lam, and Shing-Kit Chan. A framework based on graphical models with logic for chinese named entity recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-08)*, pages 335–342, Hyderabad, India, 2008.

- [101] Xiaofeng Yu, Wai Lam, Shing-Kit Chan, Yiu Kei Wu, and Bo Chen. Chinese NER using CRFs and logic for the fourth SIGHAN bakeoff. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing (SIGHAN-6)*, pages 102–105, Hyderabad, India, 2008.
- [102] Xiaofeng Yu, Wai Lam, and Bo Chen. An integrated discriminative probabilistic approach to information extraction. In *Proceedings of CIKM-09*, pages 325–334, Hong Kong, China, 2009.
- [103] Hua Ping Zhang, Qun Liu, Xue-Qi Cheng, Hao Zhang, and Hong Kui Yu. Chinese lexical analysis using Hierarchical Hidden Markov Model. In *2nd SIGHAN Workshop on Chinese Language Processing*, volume 17, pages 63–70, 2003.
- [104] Guodong Zhou and Jian Su. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of ACL-02*, pages 473–480, Philadelphia, USA, 2002.
- [105] Junsheng Zhou, Liang He, Xinyu Dai, and Jiajun Chen. Chinese named entity recognition with a multi-phase model. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.
- [106] Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of WWW-09*, pages 101–110, Madrid, Spain, 2009.
- [107] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2D conditional random fields for Web information extraction. In *Proceedings of ICML-05*, pages 1044–1051, Bonn, Germany, 2005.
- [108] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. Simultaneous record detection and attribute labeling in Web data extrac-

- tion. In *Proceedings of KDD-06*, pages 494–503, Philadelphia, Pennsylvania, USA, 2006.
- [109] Jun Zhu, Zaiqing Nie, Bo Zhang, and Ji-Rong Wen. Dynamic hierarchical Markov random fields and their application to Web data extraction. In *Proceedings of ICML-07*, pages 1175–1182, Corvallis, Oregon, 2007.
- [110] Jun Zhu, Zaiqing Nie, Bo Zhang, and Ji-Rong Wen. Dynamic hierarchical Markov random fields for integrated Web data extraction. *Journal of Machine Learning Research*, 9:1583–1614, 2008.
- [111] Jun Zhu, Bo Zhang, Zaiqing Nie, Ji-Rong Wen, and Hsiao-Wuen Hon. Webpage understanding: an integrated approach. In *Proceedings of KDD-07*, pages 903–912, San Jose, California, USA, 2007.