Improving Discriminative Sequential Learning by Discovering Important Association of Statistics

XUAN-HIEU PHAN, LE-MINH NGUYEN, YASUSHI INOGUCHI, and TU-BAO HO
Japan Advanced Institute of Science and Technology
and
SUSUMU HORIGUCHI
Tohoku University

Discriminative sequential learning models like Conditional Random Fields (CRFs) have achieved significant success in several areas such as natural language processing or information extraction. Their key advantage is the ability to capture various nonindependent and overlapping features of inputs. However, several unexpected pitfalls have a negative influence on the model's performance; these mainly come from a high imbalance among classes, irregular phenomena, and potential ambiguity in the training data. This article presents a data-driven approach that can deal with such difficult data instances by discovering and emphasizing important conjunctions or associations of statistics hidden in the training data. Discovered associations are then incorporated into these models to deal with difficult data instances. Experimental results of phrase-chunking and named entity recognition using CRFs show a significant improvement in accuracy. In addition to the technical perspective, our approach also highlights a potential connection between association mining and statistical learning by offering an alternative strategy to enhance learning performance with interesting and useful patterns discovered from large datasets.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning; G.3 [**Probability and Statistics**]

General Terms: Algorithms; Experimentation

Additional Key Words and Phrases: Discriminative sequential learning, feature selection, association rule mining, information extraction, text segmentation

1. INTRODUCTION

Discriminative models like Maximum Entropy (MaxEnt) [Berger et al. 1996], Discriminative HMMs [Collins 2002], Maximum Entropy Markov Models

This research was supported in part by JSPS Grant-in-Aid for Scientific Research.
Author's address: X.-H. Phan, School of Information Science, Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa 923-1292; email: hieuxuan@jaist.ac.jp.

(MEMMs) [McCallum et al. 2000], and CRFs [Lafferty et al. 2001] have achieved significant success in many labeling and segmenting tasks for sequence data such as POS tagging [Ratnaparkhi 1996], text shallow parsing [Peng et al. 2004; Sha and Pereira 2003], information extraction [Kristjansson et al. 2004; Pinto et al. 2003], object detection in computer vision [Torralba et al. 2004], image analysis and labeling [Kumar and Hebert 2003; He et al. 2004], and biological sequence modeling [Yeo and Burge 2003]. The noticeable advantage of these models is their flexibility in integrating a variety of arbitrary, overlapping, and nonindependent features at different levels of granularity from the observed data.

However, applications employing these models with fixed and hand-built feature templates usually generate a huge number of features, up to millions, for example, in Sha and Pereira [2003]. This is because one usually utilizes complex templates, including conjunctions of atomic statistics, for instance, n-gram of words or POS tags, to cover as many combinations of statistics as possible without eliminating irrelevant ones. As a result, models using long and fixed conjunction templates are heavily overfitting and time consuming to train because they contain many teacher-specific and redundant features. To reduce these drawbacks, McCallum [2003] proposed a likelihood-driven feature induction for CRFs that is based on a famous-feature inducing strategy for exponential models [Pietra et al. 1997]. This method iteratively adds the conjunctions of atomic observational tests that increase conditional log-likelihood into the model until some stopping criteria are reached. In spite of attaining a trade-off between the number of used features and model accuracy, this strategy may ignore rare but sensitive conjunctions with smaller likelihood gains that are still critical to model performance. Also, when the number of atomic statistics is large, the number of conjunctions becomes explosive, and thus ranking all conjunctions by likelihood gain is very expensive.

In this article, we propose a data-driven approach that can identify and emphasize rare but important conjunctions or co-occurrences of statistics[1] hidden in training data in order to improve prediction accuracy for difficult data instances. The main motivation and the underlying idea of this approach are based on the fact that sequence data, such as natural language or biological information, potentially contain the following phenomena that can be the major sources of prediction errors

—*Ambiguous data instances* usually contain unclear contextual clues that may result in misleading predictions. For instance, it is quite difficult for a phrase-chunker to determine whether the word *plans* in the text *the trip plans for Japan* is a singular verb or a plural noun.

—*Irregular instances* are recognized as exceptions to the common statistics or decisions. For example, a POS tagger may mark *walk* as a noun in the sentence *The disabled walk very slowly* because of a regular sequential

[1]In this article, terms like "(atomic) context predicates", "(singleton) statistics", or "(atomic) observational tests" are used interchangeably to refer to particular kinds of contextual information observed from the training data

dependency that a noun should follow an adjective. However, the correct interpretation is that *The disabled* is the subject and *walk* is a plural verb rather than a noun.

—*Unbalanced data* occurs when the distribution of classes in the training data is unbalanced. For example, the number of English noun phrases is much larger than that of other phrase types, for example, adjective phrases. This may lead to low prediction accuracy for minor classes due to the dominance of major ones.

—*Frequently vs. less-observed data*, for instance, a named entity recognizer may identify *New York University* as a location, while it is in fact an organization. This is because *New York* is observed more frequently than *New York University*.

—*Long dependencies in sequence data*. Several kinds of sequential training data contain long dependencies among observations at different positions in a sequence. The problem is that one can not always use a large sliding window to capture such useful clues because it would generate too many irrelevant features.

Data instances falling into the these situations should be difficult data examples. Thus, the prediction of their labels does not usually obey the frequently observed statistics. In other words, the simple aggregation of singleton statistics may lead to misleading predictions because the common statistics always overwhelm uncommon ones. To overcome this pitfall, a model should rely on higher-order features that are based on special conjunctions of singleton statistics to win the dominance of common decisions. Although those conjunctions may only occur several times in a whole dataset, their appearance is an important source of evidence to deal with difficult data instances.

In spite of their benefit, searching such conjunctions from big datasets is challenging because the number of candidates is prohibitively large. Fortunately, we find that recent association-rule mining techniques are very useful for discovering such patterns. In our method, those conjunctions belong to a subset of rare but highly confident association rules discovered from the training data. Selected conjunctions are then integrated into the discriminative models in three ways to improve their prediction accuracy: (a) conjunctions as normal features, (b) conjunctions as weighted features, and (c) conjunctions as constraints for the inference process.

The rest of this article is organized as follows. Section 2 briefly introduces linear-chain CRFs, a typical sequential learning model. Section 3 presents the framework for discovering important associations of statistics from training data. Section 4 describes how to learn sequential models with discovered associations. Section 5 describes and discusses the experimental results. Section 6 reviews related work. Finally, conclusions and future work are given in Section 7.

2. DISCRIMINATIVE SEQUENTIAL LEARNING

The goal of labeling/tagging for sequence data is to learn to map observation sequences to their corresponding label sequences, for instance, the sequence

of POS tags for words in a sentence. Discriminative HMMs [Collins 2002], MEMMs [McCallum et al. 2000], and CRFs [Lafferty et al. 2001] were intentionally designed for such sequential learning applications. In contrast to generative models like HMMs [Rabiner 1989], these models are discriminative, that is, trained to predict the most likely label sequence given the observation sequence. In this article, CRFs are referred to as the undirected linear-chain of model states, that is, conditionally-trained finite state machines (FSMs) that obey the first-order Markov independence assumption. The strength of CRFs is that they can combine both the sequential property of HMMs and the maximum entropy principle as well as global normalization that can avoid the label-bias problem. In our work, CRFs were used to conduct all experiments.

2.1 Conditional Random Fields

Let $\mathbf{o} = (o_1, o_2, \ldots, o_T)$ be some observed data sequence. Let $\boldsymbol{S}$ be a set of FSM states, each of which is associated with a label, $l \in \boldsymbol{L}$. Let $\mathbf{s} = (s_1, s_2, \ldots, s_T)$ be some state sequence; Lafferty et al. [2001] define CRFs as the conditional probability of a state sequence given an observation sequence as

$$p_\theta(\mathrm{s}|\mathrm{o}) = \frac{1}{Z(\mathrm{o})} \exp\left[\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(s_{t-1}, s_t, \mathrm{o}, t)\right], \tag{1}$$

where $Z(\mathrm{o}) = \sum_{\mathrm{S}}, \exp(\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(s'_{t-1}, s'_t, \mathrm{o}, t))$ is a normalization summing over all label sequences. f_k denotes a feature function in the language of maximum entropy modeling and λ_k is a learned weight associated with feature f_k. Each f_k is either a *per-state* or a *transition* feature:

$$f_k^{(per\text{ - }state)}(s_t, \mathrm{o}, t) = \delta(s_t, l)x_k(\mathrm{o}, t) \tag{2}$$

$$f_k^{(transition)}(s_{t-1}, s_t, t) = \delta(s_{t-1}, l')\delta(s_t, l), \tag{3}$$

where δ denotes the Kronecker-δ. A per-state feature (2) combines the label l of current state s_t and a statistic or context predicate, that is, the binary function $x_k(\mathbf{o}, t)$ that captures a particular property of the observation sequence $\mathbf{o}$ at time position t. For example, the current label is JJ (adjective) and the current word is sequential. A transition feature (3) represents sequential dependencies by combining the label l' of the previous state s_{t-1} and the label l of the current state s_t, such as the previous label $l' =$ JJ and the current label $l =$ NN (noun).

2.2 Inference in Conditional Random Fields

Inference in CRFs is to find the most likely state sequence $\mathbf{s}^*$ given the observation sequence $\mathbf{o}$,

$$\mathrm{s}* = \arg\max_{\mathrm{S}} p_\theta(\mathrm{s}|o) = \arg\max_{\mathrm{S}} \left\{\exp\left[\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(s_{t-1}, s_t, \mathrm{o}, t)\right]\right\}. \tag{4}$$

In order to find $\mathbf{s}^*$, one can apply a dynamic programming technique with a slightly modified version of the original Viterbi algorithm for HMMs [Rabiner 1989]. To avoid an exponential time search over all possible settings of $\mathbf{s}$, Viterbi

stores the probability of the most likely path up to time t, which accounts for the first t, observations and ends in state s_i. We denote this probability to be $\varphi_t(s_i)$ $(0 \leq t \leq \mathrm{T}\text{ - }1)$ and $\varphi_0(s_i)$ to be the probability of starting in each state s_i. The recursion is given by

$$\varphi_{t+1}(s_i) = \max_{s_j} \left\{ \varphi_t(s_j) \exp \left[\sum_k \lambda_k f_k(s_j, s_i, \mathrm{o}, t) \right] \right\}. \tag{5}$$

The recursion terminates when $t = \mathrm{T}\text{ - }1$ and the biggest unnormalized probability is $p* = \arg\max_i[\varphi_T(s_i)]$. At this time, we can backtrack through the stored information to find the most likely sequence $\mathbf{s}^*$.

2.3 Training Conditional Random Fields

CRFs are trained by setting the set of weights $\theta = \{\lambda_1, \ldots\}$ to maximize the log-likelihood, $\mathbf{L}$, of a given training data set $D = \{(\mathrm{o}^{(k)}, \mathrm{l}^{(k)})\}_{k=1}^{N}$:

$$L = \sum_{j=1}^{N} \log \left(p_\theta(\mathrm{l}^{(j)} | \mathrm{o}^{(j)}) \right) - \sum_k \frac{\lambda_k^2}{2\sigma^2}, \tag{6}$$

where the second sum is a Gaussian prior over parameters with variance σ^2, which provides smoothing to deal with sparsity in the data [Chen and Rosenfeld 1999].

When the labels make the state sequence unambiguous, the likelihood function in exponential models such as CRFs is convex, thus searching the global optimum is guaranteed [McCallum 2003]. However, the optimum can not be found analytically. Parameter estimation for CRFs requires an iterative procedure. It has been shown that quasi-Newton methods, such as L-BFGS [Liu and Nocedal 1989], are most efficient [Malouf 2002; Sha and Pereira 2003]. This method can avoid the explicit estimation of the Hessian matrix of the log-likelihood by building up an approximation of it using successive evaluations of the gradient.

L-BFGS is a limited memory quasi-Newton procedure for unconstrained optimization that requires the value and gradient vector of the function to be optimized. Let s_j denote the state path of training instance j in training set $\boldsymbol{D}$, then the log-likelihood gradient component of λ_k is

$$\frac{\delta L}{\delta \lambda_k} = \left[\sum_{j=1}^{N} C_k(\mathrm{s}^{(\mathrm{j})}, \mathrm{o}^{(\mathrm{j})}) \right] - \left[\sum_{j=1}^{N} \sum_s p_\theta(\mathrm{s} | \mathrm{o}^{(j)}) C_k(\mathrm{s}, \mathrm{o}^{(j)}) \right] - \frac{\lambda_k}{\sigma^2}, \tag{7}$$

where $C_k(\mathbf{s}, \mathbf{o})$ is the count of f_k given $\mathbf{s}$ and $\mathbf{o}$ equal to $\sum_{t=1}^{T} f_k(s_{t-1}, s_t, \mathrm{o}, t)$, that is, the sum of $f_k(s_{t-1}, s_t, \mathbf{o}, t)$ values for all positions, t, in the training sequence. The first two terms show the difference between the empirical and the model expected values of f_k. The last term is the first derivative of the Gaussian prior.

Table I. Unlabeled Data Among Class Labels

Label sequence	B-NP	I-NP	**O**	B-NP	I-NP	B-VP
Observation sequence	NNP Westwood	NNP Brick	**CC and**	NNP Westwood	NNP Group	VBP Are
Label sequence	B-NP	I-NP	I-NP	B-ADVP	I-ADVP	I-ADVP
Observation sequence	NN company	NNS personnel	NN policy	RB backwards	**CC and**	RB forwards
Time steps:			−2	−1	0	+1

3. MINING IMPORTANT ASSOCIATIONS OF STATISTICS IN SEQUENCE DATA

3.1 The Need of Discovering Important Associations of Statistics

As stated earlier, sequence data, such as natural language or biological information, potentially contain difficult observations that might come from highly ambiguous and unbalanced data. These observations do not always obey frequent statistics and common sequential dependencies. Thus they are the main source of prediction errors. Furthermore, a prediction error made at a particular position in a sequence can be propagated though the sequence and influence the other positions. Table I shows an example of unbalanced phenomenon in which the observation {**and**, **CC**} and the label {**O**} (of the first sequence) occur 2,472 times in the whole CoNLL2000 shared task corpus. However, the same observation was annotated with the label {**I-ADVP**} (in the second sequence) only 22 times in the same corpus.

It is common that one can rely on contextual information in the form of statistics around each data observation in order to make the prediction more discriminative. However, the singleton statistics are not discriminative and strong enough to recognize difficult data observations. Therefore, previous methods usually utilize higher-order or complex statistics, combining two or more singleton statistics, using fixed templates [Sha and Pereira 2003]. This once again encounters the problem that long fixed templates would generate too many complex statistics (due to the combination explosion) and thus leading to the overfitting problem. Our solution is to use short fixed templates in order to keep the model as simple as possible. In addition, we discover important associations/conjunctions of singleton statistics from sequential training data to deal with difficult data observations. Those associations are rare (i.e., occur only several times in the whole corpus) but confident enough to highlight difficult observations. Recall the example in Table I about the imbalanced problem. It is useful if we can discover and use the association $\{t_{-1}$:rb $\wedge$ t_0:cc $\wedge$ w_0:and $\wedge$ t_1:rb$\} \rightarrow$ **i-advp**$\}$ to predict the label for the observation {**and**, **CC**} in the second sequence. This is because the previous above association is rare (occurs only 4 or 5 times) but highly confident (100%) to win the dominance of frequent statistics. The next two sections describe how to represent data and discover such important associations from sequential training data.

3.2 Data Representation for Discovering Important Associations of Statistics

$D = \{(\mathrm{o}^{(k)}, \mathrm{l}^{(k)})\}_{k=1}^{N}$ is the sequential training dataset in which $(\mathbf{o}^{(k)}, \mathbf{l}^{(k)})$ is the k^{th} observation and label sequence. The top part of Table II shows an example

Table II. Sequential Training Data

Label sequence	B-NP	B-PP	B-NP	I-NP	B-VP	I-VP	I-VP
Observation sequence	NN Confidence	IN in	DT the	NN pound	VBZ is	RB widely	VBN expected
Time steps of sliding window	0	1	2				
	−1	0	1	2			
	−2	−1	0	1	2		
		−2	−1	0	1	2	

Table III. Transactional Database for Discovering Important Associations of Statistics

w_0:confidence w_1:in w_2:the t_0:nn t_1:in t_2:dt **b-np**}
w_{-1}:confidence w_0:in w_1:the w_2:pound t_{-1}:nn t_0:in t_1:dt t_2:nn **b-pp**
w_{-2}:confidence w_{-1}:in w_0:the w_1:pound w_2:is t_{-2}:nn t_{-1}:in t_0:dt t_1:nn t_2:vbz **b-np**
w_{-2}:in w_{-1}:the w_0:pound w_1:is w_2:widely t_{-2}:in t_{-1}:dt t_0:nn t_1:vbz t_2:rb **i-np**
w_{-2}:the w_{-1}:pound w_0:is w_1:widely w_2:expected t_{-2}:dt t_{-1}:nn t_0:vbz t_1:rb t_2:vbn **b-vp**
w_{-2}:pound w_{-1}:is w_0:widely w_1:expected w_2:to t_{-2}:nn t_{-1}:vbz t_0:rb t_1:vbn t_2:to **i-vp**
w_{-2}:is w_{-1}:widely w_0:expected w_1:to w_2:take t_{-2}:vbz t_{-1}:rb t_0:vbn t_1:to t_2:vb **i-vp**
...

of a training sequence for English phrase-chunking where each observation (in the observation sequence) consists of both a word and its part-of-speech tag, and a class label of each observation is B-NP, I-NP, etc. $A = \{A_1, A_2, \ldots, A_M\}$ is the set of M statistic templates in which each template A_i captures a particular type of contextual information about data observations. Applying all templates in $\boldsymbol{A}$ to each position (i.e., timestep) in every training sequence of the training set $\boldsymbol{D}$, we obtain a transactional database $\boldsymbol{T}$ in which each transaction consists of a list of statistics and a particular class label.

Table II shows an example of a training data sequence. For instance, we set a sliding window of size 5 and move it along the sequence, and at each timestep, we choose the current word and its POS tag (w_0 and t_0) and those of the previous two positions (w_{-2}, t_{-2}, w_{-1}, t_{-1}) and next positions (w_1, t_1, w_2, t_2) as the set of statistic templates $\boldsymbol{A}$. Scanning this set of statistic templates over all training sequences in $\boldsymbol{D}$, we obtain the corresponding transactional database. Table III shows a part of a transactional database corresponding to the sequence in Table II in which each line is a transaction that includes a list of statistics and a particular label. For instance, the statistic "w_1:the" in the second transaction (line) says that "the next word is *the*" and the statistic "t_{-1}:nn" says that "the previous word is a noun". The **b-pp** (**b**egin of **p**repositional **p**) at the end of the second line is the class label of the current observation.

3.3 Important Associations of Statistics: A Formal Definition

Let $I = \{x_1, x_2, \ldots, x_n\}$ be the set of all statistics in transactional database $\boldsymbol{T}$. Let $\boldsymbol{L}$ be the set of all class labels in $\boldsymbol{T}$. Our target is to examine every (predictive) association rule r [Agrawal and Srikant 1994] having the following form

$$X \Rightarrow l, \tag{8}$$

where the left-hand side (LHS) of r, $X = \{x_{i1} \wedge x_{i2} \wedge \ldots \wedge x_{ip}\} \subset I$, is a conjunction of p statistics in $\boldsymbol{I}$, and the right-hand side (RHS) of r, $l \in L$, is a particular class

label. The support of r, denoted as supp(r), is the number of transactions in $\boldsymbol{T}$ containing $\{l\} \cup X$, and the confidence of r, denoted as conf(r), is the conditional probability that a transaction in $\boldsymbol{T}$ has the label l given that it contains X, that is, conf(r) = supp($\{l\} \cup X$)/supp(X). In a sense, this kind of rule is similar to the associative classification rules in Li et al. [2001] and Liu et al. [1998].

Derived from the predictive association rules defined in (8) and the concepts of support and confidence factors, we present a descriptive definition of *rare-but-confident associations*.

Definition 1. Let lsupp and usupp be two integers that are much smaller than the total number of transactions in $\boldsymbol{T}$ (i.e., lsupp $\leq$ usupp $<<|\boldsymbol{T}|$), and let lconf be a real number that satisfies the condition $0 \leq$ lconf ≤ 1 and lconf ≈ 1. A predictive association rule r in (8) is called rare-but-confident if:

$$\text{lsupp} \leq \text{supp}(r) \leq \text{usupp and conf}(r) \geq \text{lconf}.$$

All predictive association rules satisfying Definition 1 are rare-but-confident. However, not all of them are important. This is based on the important observation that if most statistics in the LHS of a rare-but-confident rule r individually tend to characterize label l, then the rule r is trival. In other words, if most statistics in the LHS of r largely support label l in a separated manner, there is no need to examine the co-occurrence of all statistics in the LHS. Based on this observation, we define the concept of *important associations* as follows.

Definition 2. A rare-but-confident association rule r: $X \Rightarrow l$ is considered to be important if there exists at least another label $l' \in \boldsymbol{L}$ such that the sum of support counts for the label l' from the statistics in the LHS of r is larger than that for the label l, that is,

$$\exists l' \in L : \sum_{x \in X} \text{sup}(x \Rightarrow l') > \sum_{x \in X} \text{sup}(x \Rightarrow l).$$

Why are predictive association rules satisfying Definition 2 important? Intuitively, if such a rule, r, exists in the training data but is not discovered and emphasized, the model may predict the label l' for any data transaction holding all statistics in LHS of r, when the correct label is l. This is because most singleton statistics in the LHS of r tend to support l' rather than l. This is why the appearance of predictive association rules satisfying Definition 2 is important. There should be more sophisticated definitions and conditions of important associations. However, we choose the previous definition because of the trade-off between the rigorousness and the simplicity of calculation.

Table IV shows some examples of important associations discovered from transactional database $\boldsymbol{T}$ (Table III), using association rule-mining techniques [Agrawal and Srikant 1994; Han et al. 2000] and two filtering criteria in Definitions 1 and 2. The first column displays important associations in which the LHSs are conjunctions of statistics and the RHSs are class labels (l). The second column shows the support and confidence factors and the last column is the label l' that satisfy Definition 2.

Table IV. Examples of Important Associations of Statistics Discovered from Training Data

Important associations: $X \Rightarrow l$	Supp & Conf	Label l'
{w_1:in ∧ t_1:in ∧ t_0:vb ∧ t_{-1}:md ∧ w_0:result} ⇒ **i-vp**	(5, 100%)	**i-np**
{w_2:the ∧ t_1:in ∧ t_2:dt ∧ t_0:in ∧ t_{-1}:vbd ∧ w_0:that} ⇒ **b-sbar**	(7, 100%)	**b-np**
{w_2:the ∧ t_1:in ∧ t_2:dt ∧ t_{-1}:nns ∧ w_1:as ∧ w_0:such} ⇒ **b-pp**	(5, 100%)	**b-np**
{w_2:the ∧ t_2:dt ∧ t_0:rb ∧ t_1:vb ∧ t_{-1}:md ∧ w_0:only} ⇒ **i-vp**	(4, 100%)	**i-np**
{w_2:the ∧ t_1:in ∧ w_0:even ∧ w_1:though} ⇒ **b-sbar**	(4, 100%)	**b-advp**
{t_1:in ∧ t_2:nnp ∧ t_{-1}:nns ∧ w_1:of ∧ w_0:south} ⇒ **b-advp**	(4, 100%)	**b-np**
{w_2:the ∧ t_2:dt ∧ t_0:nns ∧ w_{-1}:for ∧ w_0:years} ⇒ **b-np**	(4, 100%)	**i-np**
{t_0:in ∧ t_{-2}:vbn ∧ t_2:cd ∧ w_0:at ∧ w_1:least} ⇒ **b-np**	(4, 100%)	**b-pp**
{t_0:in ∧ t_2:nn ∧ w_0:as ∧ w_{-2}:as ∧ w_{-1}:well} ⇒ **i-conjp**	(4, 100%)	**b-adjp**
{w_1:in ∧ t_1:in ∧ t_0:rb ∧ w_0:early} ⇒ **b-advp**	(4, 100%)	**b-np**
{t_1:in ∧ t_2:dt ∧ t_0:jj ∧ w_1:for ∧ w_0:good} ⇒ **b-adjp**	(4, 100%)	**b-np**
...		

3.4 Important Associations of Statistics in the Context of Exception Rule Mining

Discovering exception rules is one of the important directions in association rule-mining. The target is to find out interesting exception rules/patterns that decision makers can use for advantageous actions. However, finding such rules is quite difficult because it is much harder to know which of the discovered rules are really interesting. Interestingness is a relative issue since it always depends on the user's prior knowledge about the domain. Interestingness can be either user's (biased) belief or estimated relative to the commonsense rules found in the data.

An interesting exception is something that contradicts the user's common belief. Exceptions [Padmanabhan and Tuzhilin 1998; Silberchats and Tuzhilin 1996; Suzuki and Shimura 1996; Suzuki 1997] can play an important role in making critical decisions. Exceptions and commonsense rules point in opposite directions. Exceptions are usually minority, they are either not known or omitted from the normal discovery process. Intuitively, exceptions contradict the commonsense rules, and they have low support [Suzuki 1997]. Therefore, exception rules are weak in terms of support, but have high confidence similar to commonsense rules. A weak rule of low support may not be reliable. A user can specify minimum support for exceptions to ensure mining reliable exception rules. In addition, exception rules are evaluated by several other criteria, such as generality, monotonicity, reliability, search range, interpretation of the evaluation measure, use of domain knowledge, and success in real applications.

Most of the previous studies discovered exception rules/patterns for supporting data analysis or decision making. Our work, on the other hand, focuses on searching important associations of statistics in linguistic data to improve prediction accuracy in natural language learning problems. Our important associations of statistics, in a sense, follow some criteria of exception rules (e.g., low support, high confidence, contradicting commonsense rules, etc.). However, they are different in the discovery process, filtering criteria, and the reasons to discover them.

3.5 Discovering Important Associations of Statistics

To discover important associations of statistics, we first apply one of the association mining techniques to discover normal associations (of statistics). Then,

those associations are filtered using the conditions previously stated to obtain only the important ones. Mining normal associations includes two main steps: (1) discovering frequent patterns (i.e., frequent itemsets) and (2) generating associations from those frequent patterns. The first step is most challenging, while the second step, rule generation, is quite straightforward. Thus, almost all studies have focused on the first one to reduce the overall complexity of the mining process.

Frequent-pattern mining plays an essential role not only in mining associations, but also in discovering correlations, causality, sequential patterns, episodes, multidimensional patterns, partial periodicity, emerging patterns, and many other important data mining tasks [Han et al. 2004]. In principle, we can use any previous frequent-patterns mining technique to find frequent patterns for generating important associations of statistics in linguistic corpora. However, due to the characteristics of this kind of rule, such as rare and highly confident, not all of them are efficient for this task. Here, we briefly introduce one of the most efficient techniques for discovering frequent patterns, which is particularly appropriate for mining important associations.

Most previous studies in mining frequent patterns adopt an *Apriori*-like approach, which is based on the antimonotone Apriori heuristic [Agrawal and Srikant 1994]: if any length k pattern is not frequent in the database, its length $(k + 1)$ superpattern can never be frequent. The essential idea is to iteratively generate the set of candidate patterns of length $(k + 1)$ from the set of frequent patterns of length k (for $k \geq 1$), and check their corresponding occurrence frequencies in the database.

The Apriori-like heuristic achieves good performance gained by reducing the size of candidate sets. However, in situations with a large number of frequent patterns, long patterns, or quite low minimum support thresholds, an Apriori-like algorithm may suffer from the following nontrivial costs.

—It is costly to handle a huge number of candidate sets. This is very critical because the number of combinations of statistics in linguistic data is extremely large due to the fact that natural language data is diverse and sparse. For instance, if we have 10 statistic templates and each generates 2,000 statistics on average, then the number of combinations is approximately 10^{33}. Frequent-pattern mining techniques can, of course, prune unnecessary candidates using the Apriori heuristic, however this explosion of candidates strongly degrades the performance of Apriori-like algorithms.

—Important associations are rare, that is, their supports are very small in comparison with the database size (i.e., the total number of transactions $|\boldsymbol{T}|$). This is probably the biggest challenge for mining important associations. In fact, Apriori-like algorithms are infeasible for mining important associations from large linguistic corpora.

The previous challenges seem to prevent us from discovering all possible important associations in linguistic corpora. Fortunately, FP-growth [Han et al. 2000, 2004], a frequent-pattern mining algorithm without candidate generation, can discover such associations in an acceptable computational time. This

Table V. Steps for Discovering Important Associations of Statistics from Linguistic Corpora

Step	Description
1.	Converting the linguistic corpus $\boldsymbol{D}$ into a transactional database $\boldsymbol{T}$
2.	Finding frequent patterns from $\boldsymbol{T}$ with minimum support threshold lsupp using the FP-growth algorithm
3.	Generating normal associations from the above frequent patterns that satisfy the condition in Definition 1.
4.	Filtering the normal associations using the Definition 2 to obtain important associations of statistics.

is because FP-growth employs a FP-tree (an extended prefix tree structure) to store crucial, quantitative information about frequent patterns in such a way that more frequently occurring items will have better chances of sharing nodes than less frequently occurring ones. All mining operations are then performed on the FP-tree in a partitioning, recursive fashion without candidate generation. See Han et al. [2000, 2004] for a complete description of this algorithm.

Table V shows the steps necessary for discovering important associations of statistics. The first thing we have to do is to convert the original sequence training data set into its transactional form as described earlier. Next, the FP-growth algorithm will be used to discover all frequent patterns/itemsets that satisfy the minimum support threshold lsupp. Then, normal associations are generated from those frequent patterns/itemsets. This step is quite straightforward and much cheaper than the second one. See Agrawal and Srikant [1994] for a complete description of the rule generation algorithm. Finally, the resulting normal associations will be filtered using the condition in Definition 2 to obtain important ones.

Because the first, third, and the last steps are much cheaper in comparison with the second step, the computational time of the discovery process depends on that of the FP-growth algorithm. As stated earlier, because the FP-growth can discover frequent patterns without candidate generation, it can avoid the problem of combination explosion. As proved in Han et al. [2000, 2004], this algorithm is much more efficient than Apriori-like algorithms, especially when the minimum threshold is much smaller than the database size. Our empirical study also shows that FP-growth is appropriate for mining patterns in linguistic corpora even though natural language is much more diverse and sparse in comparison to traditional numerical and categorical data.

4. LEARNING CRFS WITH IMPORTANT ASSOCIATIONS OF STATISTICS

This section presents three ways to incorporate the important associations discovered from the training data into CRFs: (1) associations as normal features, (2) associations as features with emphasized feature functions, and (3) associations as constraints for the inference process.

4.1 Important Associations as Normal CRF Features

All important associations of statistics are in the form of $X \Rightarrow l$ in which $X = \{x_{i1} \wedge x_{i2} \wedge \ldots \wedge x_{ip}\} \subset I$ is a conjunction of p statistics, and $l \in \boldsymbol{L}$ is a

particular label. These associations can be integrated into CRFs in terms of normal per-state features as follows.

$$f_k^{(per\text{ - }state)}(s_t, \mathrm{o}, t) = \delta(s_t, l)\{x_{i1}(\mathrm{o}, t) \wedge \ldots \wedge x_{ip}(\mathrm{o}, t)\}.$$

These per-state features are similar to those in (2) except that they capture a co-occurrence of p singleton statistics rather than a single one. The features are treated as normal features of CRFs and are trained together.

4.2 Important Associations as Weighted CRF Features

It is noticeable that important features are infrequently observed in the training data, and thus their learned weights should be small. This means that their contributions, in several cases, may not be sufficient to win the dominance of common statistics, that is, frequently observed singleton features. To overcome this drawback, we emphasize important features by assigning larger feature function values compared to normal features.

$$f_k^{(per-state)}(s_t, \mathrm{o}, t) = \begin{cases} v & \text{if } \delta(s_t, l) \text{ and } \{x_{i1}(\mathrm{o}, t) \wedge \ldots \wedge x_{ip}(\mathrm{o}, t)\} \\ 0 & \text{otherwise,} \end{cases}$$

where $\delta(s_t, l)$ are considered as logic expressions, and v is larger than 1 (the feature value of normal features). v should be large if the occurrence frequency of the feature (also the support of the important association) is small. Thus, for each feature generated from a important association r, v is equal to (usup $-$ supp$(r) + 2$). This ensures that v is always bigger than 1 and inversely proportional to the support of r, that is, the occurrence frequency of the feature.

4.3 Important Associations as Constraints for Inference in CRFs

Constrained CRFs are extensions of CRFs in which useful constraints are incorporated into the inference process (i.e., the Viterbi algorithm) to correct potential errors existing in the most likely output state sequence for each input observation sequence. Kristjansson et al. [2004] proposed this extension with the application to interactive form-filling in which users can examine the filling process and make necessary corrections in terms of their own constraints. A recorrection applied at a particular position will propagate though the Viterbi sequence to make automatic updates for labels at other positions, that is, the correction propagation capability.

This section presents the integration of important associations with 100% confidence into the Viterbi algorithm in terms of data-driven constraints to make corrections directly to the inference process of CRFs. Unlike those used in Kristjansson et al. [2004], our constraints are 100% confidence associations and are automatically discovered from the training data.

Normally, CRFs use a variant of the traditional Viterbi algorithm to find the most likely state sequence given an input observation sequence. To avoid an exponential-time search over all possible settings of state sequence, this algorithm employs a dynamic programming technique with a forward variable $\varphi_{t+1}(s_i)$ in Equation (5).

Let $\boldsymbol{R} = \{r_1, r_2, \ldots, r_q\}$ be a set of q important associations with 100%-confidence, and each r_u $(1 \leq u \leq q)$ has the form $\{\mathrm{x}_{u1} \wedge \mathrm{x}_{u2} \wedge \ldots \wedge \mathrm{x}_{up}\} \Rightarrow l_u$ $(l_u \in \boldsymbol{L})$. Each $r_u \in \boldsymbol{R}$ is considered to be a constraint for the inference process. At each time position in the testing data sequence, we check whether or not the set of active statistics at the current position holds the LHS of any rule $r_u \in \boldsymbol{R}$. If yes, the most likely state path must go though the current state with the label l_u (i.e., the RHS or rule r_u), and the possibility of passing though other labels equals zero. The constrained forward variable is redefined as follows.

$$\varphi_{t+1}(s_i) = \begin{cases} \max_{s_j}\{\varphi_t(s_j)\exp[\sum_k \lambda_k f_k(s_j, s_i, \mathrm{o}, t)]\} \\ \text{if } \delta(s_i, l_u) \text{ and } \{x_{u1}(\mathrm{o}, t) \wedge \ldots \wedge x_{up}(\mathrm{o}, t)\} \\ 0 \qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{9}$$

The constraint applied at the time position t will propagate though the whole sequence and make some recorrections for labels at other positions (mostly around the position t).

One problem is that when the number of constraints (i.e., the number of 100%-confidence important associations) is large, the time for examining the LHS of every rule at each position in the testing sequence also becomes large. To overcome this obstacle, we propose the following algorithm for fast-checking constraints at a particular time position t in the testing sequence.

Let $\boldsymbol{R} = \{r_1, r_2, \ldots, r_q\}$ be the set of 100%-confidence rules, also known as constraints, and let $\boldsymbol{X} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_m\}$ be the set of m active statistics observed at the current position t. The target of the following algorithm is to check whether or not $\boldsymbol{X}$ holds the LHS of any constraint $r_u \in \boldsymbol{R}$. If yes, choose the constraint with the longest LHS.

—For each $\mathrm{x}_i \in \boldsymbol{X}$, look up the set of constraints $\boldsymbol{R}_i \subset \boldsymbol{R}$ in which the LHS of every constraint in $\boldsymbol{R}_i$ contains x_i. Denote $\boldsymbol{R}' = \{\boldsymbol{R}_1 \cup \boldsymbol{R}_2 \cup \ldots \cup \boldsymbol{R}_m\}$.

—For each constraint $r_j \in \boldsymbol{R}'$, let c_j be the sum of occurrence frequency of $r_j \in \boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_m$.

—Find the pair $<r_j, c_j>$ $(1 \leq j \leq |\boldsymbol{R}'|)$ such that c_j is the largest number satisfying the condition: c_j equals to the number of all statistics in the LHS of r_j.

If this algorithm finds a constraint r_j, then apply this constraint to the current position t with Equation (9), otherwise, apply the normal Viterbi recursion as in Equation (5).

For an efficient implementation, we maintain a map between statistics and the rules containing them. Each map component is a pair of statistics and a list of rule indexes that contain that statistic in their left-hand side. We can use a hash table or any indexing data structures for maintaining this map in the main memory. The initialization of this map is performed only once when loading the rules into the memory. The complexity of the previous algorithm is determined as follows. Let $\overline{m}$ be the average number of active statistics in $\boldsymbol{X}$; let $\overline{l}$ be the average length of the list of rule indexes in each map pair; let $\overline{t}$ be the average time for looking up a statistic in the map. The time complexity of the first step in the algorithm is $\mathbf{O}(\overline{mt})$. This is because we have to look up $\overline{m}$ statistics of $\boldsymbol{X}$ in the map. The second step scans over the all the lists of rule indexes returned in the first step to count the occurrence frequencies of those

rules. Thus the complexity of this step is $\mathbf{O}(\overline{ml})$. The final step finds the pair $< r_j, c_j >$ such that c_j is the largest number satisfying the condition c_j equals to the number of statistics in LHS of r_j. The complexity of this step in the worst case is $\mathbf{O}(\overline{ml})$. Therefore, the overall complexity is $\mathbf{O}(\overline{mt}+\overline{ml}+\overline{ml}) = \mathbf{O}(\overline{mt}+\overline{ml})$.

As we will see later in the experiments, the average number of active statistics ($\overline{m}$) is around 20 or 30. The number of rules is around ten thousand and thus the looking up time in the map is very fast. We observed that if the number of rules is not too large, the constrained Viterbi inference using this algorithm can be realtime.

5. EMPIRICAL EVALUATION

5.1 Experimental Settings

All the experiments were performed with our C/C++ implementation of CRFs - FlexCRFs[2]—on a 3Gb RAM, Intel Core 2 desktop with Fedora Core 5. All CRF models were trained using the limited memory quasi-Newton method for unconstrained optimization, L-BFGS [Liu and Nocedal 1989]. The variance σ in the Gaussian prior for smoothing is 10. Unlike those used in Sha and Pereira [2003], our CRF models are simpler and easier to implement by obeying the first-order Markov property, that is, the label of the current state depends only on the label of the previous state. Training and testing data for phrase-chunking and named entity recognition can be found at the shared tasks of CoNLL2000[3] and CoNLL2003[4], respectively.

5.2 Phrase Segmentation

Phrase-chunking, an intermediate step toward full parsing of natural language, identifies phrase types (e.g., noun phrase - NP, verb phrase - VP, PP - prepositional phrase, etc.) in text sentences. Here is an example of a sentence with phrase marking: [NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September].

Training and Testing Data. The training and testing data for this task is available at the shared task for CoNLL-2000. The data consist of the same partitions of the Wall Street Journal corpus (WSJ): sections 15-18 as training data (8,936 sentences, 211,727 tokens) and section 20 as testing data (2,012 sentences, 47,377 tokens). Each line in the annotated data is for a token and consists of three columns: the token (a word or a punctuation mark), the part-of-speech tag of the token, and the phrase type label (label for short) of the token. The label of each token indicates whether the token is outside a phrase (O), starts a phrase (B-<PhraseType>), or continues a phrase (I-<PhraseType>). For example, the label sequence of the previous sentence is B-NP B-VP B-NP I-NP I-NP I-NP B-VP I-VP B-PP B-NP I-NP I-NP I-NP B-PP B-NP O. This

[2]The source code and documents of FlexCRFs are available at www.jaist.ac.jp/~hieuxuan/flexcrfs/flexcrfs.html or sourceforge.net/projects/flexcrfs.
[3]http://cnts.uia.ac.be/conll2000/chunking/.
[4]http://cnts.uia.ac.be/conll2003/ner/.

Table VI. Feature Templates for Phrase Chunking

Transition feature templates	
Current state s_i	Previous state s_{i-1}
l	l'
Per-state feature templates	
Current state s_i	Statistics $\mathrm{x}(\mathbf{o}, i)$
l	w_{-2}; w_{-1}; w_0; w_1; w_2; $\mathrm{w}_{-1} \wedge \mathrm{w}_0$; $\mathrm{w}_0 \wedge \mathrm{w}_1$; t_{-2}; t_{-1}; t_0; t_1; t_2; $\mathrm{t}_{-2} \wedge \mathrm{t}_{-1}$; $\mathrm{t}_{-1} \wedge \mathrm{t}_0$; $\mathrm{t}_0 \wedge \mathrm{t}_1$; $\mathrm{t}_1 \wedge \mathrm{t}_2$; $\mathrm{t}_{-2} \wedge \mathrm{t}_{-1} \wedge \mathrm{t}_0$; $\mathrm{t}_{-1} \wedge \mathrm{t}_0 \wedge \mathrm{t}_1$; $\mathrm{t}_0 \wedge \mathrm{t}_1 \wedge \mathrm{t}_2$

dataset contains 11 phrase types as shown in the first column of Table VII. Two consecutive data sequences (sentences) are separated by a blank line.

On the phrase-chunking dataset, we use feature templates as shown in Table VI. All transition features obey the first-order Markov dependency that the label (l) of the current state depends on the label (l') of the previous state (e.g., "l = I-NP" and "l' = B-NP"). Each per-state feature expresses how much influence a statistic ($\mathrm{x}(\mathbf{o}, i)$ observed surrounding the current position i has on the label (l) of the current state. A statistic captures a particular property of the observation sequence. For instance, the per-state feature "l = I-NP" and "word_{-1} is *the*" indicates that the label of the current state should be I-NP (i.e., continue a noun phrase) if the previous word is *the*. Table VI describes both transition and per-state feature templates. Statistics for per-state features are identities of words, POS tags surrounding the current position such as words and POS tags at -2, -1, 1, 2.

We also employ 2-order conjunctions of the current word with the previous ($\mathrm{w}_{-1} \wedge \mathrm{w}_0$) or the next word ($\mathrm{w}_0 \wedge \mathrm{w}_1$), and 2-order and 3-order conjunctions of two or three consecutive POS tags within the current window to make use of the mutual dependencies among singleton properties. With the feature templates shown in Table VI and the feature rare threshold of 1 (i.e., only features with occurrence frequency larger than 1 are included into the CRF model), 321,526 statistics and 152,856 CRF features were generated from 8,936 training sequences.

Mining Important Associations of Statistics. Let $\boldsymbol{I}$ be the itemset of 84,951 data items (only singleton statistics are used for mining), that is, the union set of 84,951 statistics and 22 phrase labels; $\boldsymbol{T}$ be the set of 211,727 data transactions corresponding to 211,727 tokens of the training data. Let the lower support (lsupp) threshold be 2 (usupp is 100%); let the lower confidence (lconf) threshold be 100%. We also confine the length of the LHS of all important associations to 2 or 3. There are several reasons why we confine the LHS length to 2 or 3. First, although simple rules (i.e., with shorter LHS length) are usually useful for generalization, we both examine both simple and complex rules provided that they are rare and confident because our main target is to deal with hard data instances which are not frequently observed in the training data. We also observed that important associations with short LHS length are useful for reducing overfitting. Second, rules with LHS length more than 3 are usually too specific and most of them are covered by rules with LHS length of 2 and 3. Also, mining and generating all long rules is time-consuming.

Table VII. English Phrase-Chunking Performance Using CRFs with Important Associations

PhraseTypes	#Phrases	(A) F_1	(B) F_1	(C) F_1	(D) F_1
NP	12422	93.57	93.81	93.76	93.51
PP	4811	97.15	97.04	97.21	96.93
VP	4658	93.60	93.60	93.90	92.94
SBAR	535	85.55	86.32	86.01	85.61
ADJP	438	72.49	72.32	74.15	72.49
ADVP	866	79.31	79.58	80.66	79.74
PRT	106	74.63	75.49	77.83	75.24
LST	5	0.00	0.00	0.00	0.00
INTJ	2	66.67	66.67	40.00	66.67
CONJP	9	48.00	45.45	50.00	40.00
UCP	0	0.00	0.00	0.00	0.00
Macro-avg		72.34	71.98	69.82	71.11
Micro-avg	23852	93.12	93.27	93.39	92.94

The mining process for important associations took 2 hours, using the FP-growth algorithm and the filter criteria presented in Definitions 1 and 2. The output was a set of 494,881 important associations with an LHS length of 2 or 3, minimum support of 2, and confidence of 100%. This set of associations were integrated into the CRF model in terms of normal features and normal features with weighted feature values. We select 13,700 rules with support larger or equal to 20 to serve as constraints for the inference process to examine whether or not inference constraints as rare but important associations can improve the performance.

Results. Table VII shows the results for English phrase-chunking. The first column is the phrase type. The second column is the number of phrases. The next four columns display the F_1 scores of four cases: (A) without important associations, (B) important associations as normal CRF features, (C) important associations as weighted CRF features, and (D) important associations as constraints for Viterbi inference. The last two lines are the macro-average and micro-average of F_1 scores calculated in two ways: precision-recall-based and phrase-based. The former is based on the precision and recall values of separated phrase types, while the latter is based on the average numbers of human-annotated, model, and correct phrases. Intuitively, the macro-average of F_1 reflects the balance and the trade-off among per-label F_1, while the micro-average of F_1 reflects the total performance.

In the first case, we obtained the highest micro-average F_1 of 93.12%. In the second case, we achieved the highest performance of 93.27% F_1 with 0.15% higher than the original performance. In the third case, we obtained the micro-verage F_1 of 93.39%, i.e., 0.27% higher than the original result (i.e., 4% error reduction). And in the fourth case, we obtained the highest micro-average F_1 score of 92.94 (i.e., 0.18% lower than the original result) when using important associations as constraints for inference.

Table VIII shows an accuracy comparison between ours and that of the other state-of-the-art chunking systems on the CoNLL-2000 dataset. Daumé III and

Table VIII. Phrase-Chunking Performance Comparison among State-of-the-Art Systems

Methods	F_1
Daumé III & Marcu 2005: (LaSO) + external lists of named entities, etc.	94.4*x*
Ando & Zhang 2005: semi-supervised + 15 million unlabeled words	94.39
Ours (second-order Markov CRFs)	93.92
Ours (first-order Markov CRFs + important associations of statistics)	93.39
Ours (first-order Markov CRFs)	93.12
Kudo & Matsumoto 2001 (SVM combination)	93.85
Carreras & Marquez 2003 (two-layer Perceptron)	93.74
Zhang et al 2002 (generalized winnow + enhanced features from full parser)	94.17

Marcu [2005] proposed a LaSO (Learning as Search Optimization) framework that can accelerate both the training and decoding phases of structured classification by relying on an approximate search strategy. They used a rich set of features as well as several external lists of names, locations, abbreviations, etc. and achieved the highest F_1 of 94.4 on chunking. Because their feature set is quite different from ours, it is not easy to compare the learning power between LaSO and CRFs. Ando and Zhang [2005] proposed a nice semi-supervised learning framework that can extract additional information from thousands of auxiliary learning problems relevant to the main task. They used an extra 15 million words from the TREC corpus as unlabeled data to improve this task and obtained the highest F_1 scores of 94.39. Kudo and Matsumoto [2001] used a SVM combination for this task. They obtained F_1 scores of 93.85. Carreras and Marquez [2003] used two-layer Perceptron and achieved F_1 of 93.74. Zhang et al. [2002] used generalized winnow and obtained F_1 score of 93.57. They exploited enhanced linguistic features from a full parser and got the highest F_1 of 94.17.

Our three CRF models following three different experimental setups (a) first-order Markov dependency, (b) first-order Markov dependency plus important associations of statistics, and (c) second-order Markov dependency without important associations of statistics) achieved F_1 scores of 93.12, 93.39, and 93.92, respectively. We observed that the F_1 of the first-order Markov CRF model with important associations, in this case, is smaller than that of the second-order Markov CRF model without important associations. However, the computational time is much different. The second-order CRF model needed more than 200 hours (estimated, i.e., more than 8 days) to train while the first-order Markov model took only 4 hours for training and an additional 2 hours for discovering important associations of statistics. Training second-order Markov CRFs is very time-consuming due to the fact that their time complexity is $\mathbf{O}(|\boldsymbol{L}|^3)$ rather than $\mathbf{O}(|\boldsymbol{L}|^2)$ like first-order Markov models, where $\boldsymbol{L}$ is the set of all class labels. In this case, $|\boldsymbol{L}|$ is 22, and the computational time is much different between the two cases. We had to use a parallel implementation of CRFs on massively parallel computers to evaluate the second-order Markov configuration.

5.3 Named Entity Recognition

Named entity recognition (NER), a subtask of information extraction, identifies names of persons (PER), organizations (ORG), locations (LOC), times (TIME),

Table IX. Feature Templates for NER

Transition feature templates	
Current state s_i	Previous state s_{i-1}
l	l'
Per-state feature templates	
Current state s_i	Statistics x(**o**, i)
l	w_{-2}; w_{-1}; w_0; w_1; w_2; t_{-2}; t_{-1}; t_0; t_1; t_2; $t_{-2} \wedge t_{-1}$; $t_{-1} \wedge t_0$; $t_0 \wedge t_1$; $t_1 \wedge t_2$; c_{-2}; c_{-1}; c_0; c_1; c_2; $c_{-2} \wedge c_{-1}$; $c_{-1} \wedge c_0$; $c_0 \wedge c_1$; $c_1 \wedge c_2$; IsInitCap(w_k); IsAllCap(w_k); IsNumber(w_k); IsAlphaNumber(w_k); IsFirstWord(w_k) where k $\in \{-2, -1, 0, 1, 2\}$; List of names generated from training set for look-up features

and quantities (NUMBER, CURRENCY, PERCENTAGE) in natural language. Here is an example of an English sentence with named entities marked: [LOC Germany] 's representative to the [ORG European Union] 's veterinary committee [PER Werner Zwingmann] said on Wednesday...

Training and Testing Data. The training and testing data for English named entity recognition are provided at the shared task for CoNLL-2003. The dataset is a collection of news wire articles from the Reuters Corpus. The training set consists of 14,041 sentences (20,3621 tokens), and the testing data contains two parts: the development test set (testa: 3,250 sentences, 51,362 tokens) and the final test set (testb: 3,453 sentences, 46,435 tokens). The data files contain four columns separated by a blank space. Each token (a word or a punctuation mark) has been put on a separate line, and there is an empty line after each sentence (sequence). The first item on each line is a token, the second is the part-of-speech tag of the token, the third is a phrase type tag (like the label in phrase chunking) of the token, and the fourth is the named entity label (label for short). The label of each token indicates whether the token is outside a named entity (O), or inside a named entity (I-<NamedEntityType>). If two named entities of the same type immediately follow each other, the first token of the second named entity will have tag B-<NamedEntityType>. For example, the named entity label sequence of the above sentence is I-LOC O O O O I-ORG I-ORG O O O I-PER I-PER O O O.

Feature Generation. On the named entity recognition dataset, we used the feature templates shown in Table IX. All transition features also conform to the first Markov property. Each statistic for a per-state feature is one of the following types: (1) the identities of words (w_{-2}, w_{-1}, w_0, w_1, w_2), (2) the POS tags of words (t_{-2}, t_{-1}, t_0, t_1, t_2), (3) the phrase tags of words (c_{-2}, c_{-1}, c_0, c_1, c_2), and (4) several simple regular expressions or formats of words such as "the first character of a word is capitalized" (IsInitCap), "all chars of a word are capitalized" (IsAllCap), etc. We also use the lengh-2 conjunctions of POS tags and chunk tags. Like the phrase-chunking task, all statistics are captured within a window with a size of 5. Our feature templates are much simpler than those used in the previous work presented at the CoNLL2003 shared task and in McCallum [2003] in two ways, that is, only five simple format properties were captured (compared to 16 regular expressions in McCallum [2003]), and no

Table X. Some Examples of Important Associations of Statistics Discovered from the CoNLL2003 Training Set

Important Associations: $X \Rightarrow l$	Supp & Conf	Label l'
{w:0:EU ∧ w:−1:The ∧ w:1:'s} ⇒ **B-ORG**	(3, 100%)	**B-LOC**
{w:0:IsInitCap ∧ w:2:IsInitCap ∧ w:0:Everton} ⇒ **B-ORG**	(2, 100%)	**B-LOC**
{w:0:IsInitCap ∧ w:2:IsInitCap ∧ w:−1:der} ⇒ **I-PER**	(2, 100%)	**B-PER**
{w:0:Soviet ∧ w:1:Union} ⇒ **B-LOC**	(4, 100%)	**B-ORG**
{w:0:IsAllCap ∧ w:0:REUTER} ⇒ **B-PER**	(3, 100%)	**B-ORG**
{w:0:IsInitCap ∧ w:1:IsInitCap ∧ w:-1:meeting} ⇒ **B-LOC**	(2, 100%)	**B-PER**
{w:−1:Rio ∧ w:0:de} ⇒ **I-LOC**	(3, 100%)	**I-PER**
{w:−2:The ∧ w:−1:New ∧ w:0:York} ⇒ **I-ORG**	(2, 100%)	**I-LOC**
{w:−1:Banco ∧ w:0:de} ⇒ **I-ORG**	(2, 100%)	**I-PER**
{w:−1:of ∧ w:0:Pennsylvania} ⇒ **I-ORG**	(2, 100%)	**B-LOC**
{w:0:Le ∧ w:1:Gras} ⇒ **I-PER**	(2, 100%)	**B-ORG**
...		

external dictionaries were used such as the lists of people names, organization names, countries, cities, etc. We only used the list of names generated from training data for look-up features.

Mining Important Associations of Statistics. Let $\boldsymbol{I}$ be the itemset of 124,919 data items, that is, the union set of 124,910 statistics and 9 named entity labels; $\boldsymbol{T}$ be the set of 203,621 data transactions corresponding to 203,621 tokens of the training data. Let the lower support (lsupp) threshold be (usupp is 100%); let the lower confidence (lconf) threshold be 100%. We also examine rules with the LHS length of 2 or 3.

The mining process for important associations took about 1.5 hours using the FP-growth algorithm and the filter criteria described in Definitions 1 and 2. The output was a set of 391,376 rare-but-important associations with an LHS length of 2 or 3, support larger or equal to 2, and confidence of 100%. Table X shows several examples of important associations discovered from the CoNLL2003 training dataset for NER. This set of associations was integrated into the CRF model in terms of normal features and normal features with weighted feature values. We also selected 6,820 rules with support larger or equal to 20 to serve as constraints for the inference process to examine whether or not constraints as rare but important associations can improve the performance.

Results. Table XI shows the results of named entity recognition. The first column is named entity type. The second column is the number of named entities. The next four columns display the F_1 scores of four cases: (A) without important associations, (B) important associations as normal CRF features, (C) important associations as weighted CRF features, and (D) important associations as constraints for Viterbi inference. The last two lines are the macro-average and micro-average of F_1 scores.

In the first case, we obtained the highest micro-average F_1 of 85.57%. In the second case, we achieved the highest micro-average F_1 of 86.01% with 0.44% higher than the original performance. In the third case, we obtained the micro-average F_1 of 86.44%, that is, 0.87% (i.e., 6% error reduction) higher than the original result. And in the fourth case, we obtained the highest micro-average F_1 score of 85.28% (i.e., 0.29% less than the original result) when using important

Table XI. NER Performance using CRFs with Important Associations on Development Set (testa)

NE Types	#NEs	(A) F_1	(B) F_1	(C) F_1	(D) F_1
ORG	1325	79.64	78.43	79.26	79.23
PER	1829	88.16	88.56	88.88	87.88
LOC	1832	89.03	89.85	90.24	88.77
MISC	916	81.64	83.83	84.02	81.48
Macro-avg		84.65	85.18	85.61	84.37
Micro-avg	5902	85.57	86.01	**86.44**	85.28

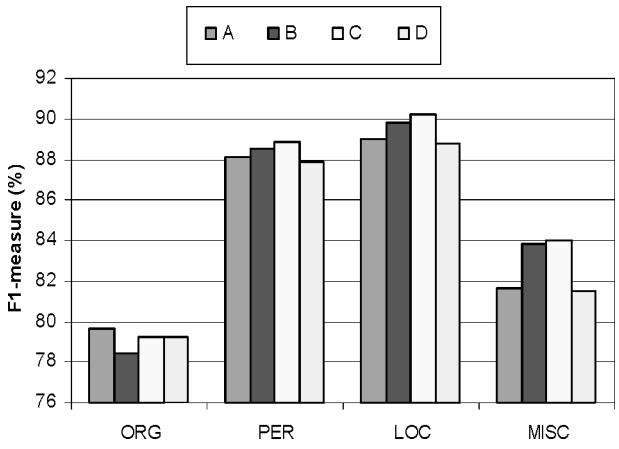

Fig. 1. NER results using CRFs with important associations on developmet set (testa).

associations as constraints for inference. Figure 1 graphically demonstrates the extent to which important associations can improve the prediction performance. We also evaluate our methods on the final test set (testb) but the improvement is lower (F_1 score from 80.86 to 81.12).

Table XII shows the difference in performance improvement between our method and McCallum's feature induction [McCallum 2003]. These results were obtained from the development test dataset (testa) but using different feature templates. Our feature templates are much simpler than those of McCallum, both in the number of regular expressions and external dictionaries. They used external information such as names of countries, publicly-traded companies, surnames, stop-words, universities, organizations, NGOs, and nationalities. Also, our CRFs provide much higher baseline performance (85.57 compared to 73.30). That is why we cannot give a direct comparison between the two methods.

On the test set (testb), Ando and Zhang [2005] used their semi-supervised learning framework to improve this task using an additional 27 million unlabeled words from Reuters and ECI Multilingal Text Corpus. They achieved the highest F_1 of 89.31. Florian et al. [2003] combined classifiers, used gazetteers, and achieved F_1 of 88.76 ± 0.7. Chieu and Ng [2003] used maximum entropy classifiers with carefully selected features plus gazetteers, and got F_1 of 88.31 ± 0.7.

Table XII. NER Performance Comparison among Feature Induction/Selection Methods

Methods	F_1
McCallum 2003 (CRFs + likelihood-driven feature induction)	89.00
McCallum 2003 (CRFs − without feature induction)	73.30
Ours (CRFs + important associations of statistics)	86.44
Ours (CRFs − without important associations of statistics)	85.57

5.4 Discussion

We can see that the integration of important associations into CRFs can improve the performance of both the phrase-chunking and named entity recognition tasks. The F_1-score of phrase-chunking increases from 93.12 to 93.39. Similarly, the F_1-score of NER increases from 85.57 to 86.44%. These are not a big improvement in performance.[5] However, they show that our method can automatically discover important associations from data to improve the accuracy for given a fixed set of feature templates.

Filtering association rules is probably the most important step in our method that strongly influences the prediction performance. In this work, we applied quite simple filtering criteria that might not completely reflect the nature/properties of rare but important associations discovered from training data. We observed that there are still a lot of redundant or meaningless associations in the selected rule sets. We believe that we can improve the prediction accuracy more if we can filter more fine-grained and significant rare but important associations.

The experiments also show that rare but important associations do not improve the accuracy if they are used as constraints for inference. This is because a lot of important associations discovered from the training data no longer have 100%-confidence in the test dataset. Using them as constraints for inference is too risky even we select only rules with support greater than or equal to 20. The experiments provided lower results in comparison with the baseline performance in both cases, phrase-chunking and NER[6]. Rules should be used as constraints for inference if we completely make sure that they are very reliable in any testing or future unseen data.

Using important associations as normal CRF features can significantly enhance the total performance. However, treating important associations as normal features might not fully utilizes their advantages. Treating important associations as weighted features should be the favorable choice because they are neither too loosely nor too tightly integrated with the models. The experimental results show that this way achieved highest improvement in accuracy.

We also did the experiments with rare-but-confident rules. We observed two important points that (1) the numbers of rare-but-confident rules (both noun phrase-chunking and named entity recognition) were much larger than those of important ones; and (2) the experimental results were sometimes worse because of the overfitting problem. This means that there is a large proportion

[5]The results in our KDD paper [Phan et al. 2005] are incorrect due to some evaluation mistakes. We unintentionally included the labels in training and testing datasets when we match association rules back to training data and look up constraints for testing sets.

of rare-but-confident rules that are unnecessary for capturing difficult data instances.

One of the important points in the proposed technique is how to determine the values for the parameters, such as lower bound (lsupp) and upper bound (usupp) of support factor of important associations of statistics, as well as the emphasized values for weighted features of CRFs. In both experimental tasks, we chose the lsupp of 2 is because we want to capture as many associations of statistics as possible. This helps us to capture more patterns in testing and future unseen data to reduce overfitting problem. We also tried larger lsupp values, such as 3 and 4, but they provided lower accuracy. We did not confine the upper support bound (usupp) because when the confidence bound is 100%, the number of rules with large support is very small. And after filtering using Definition 2, selected rules tend to have small support values. In addition, choosing the length of the LHS of important associations is quite important for two main reasons, computational time and the overfitting problem. Generating long associations is quite time-consuming since the number of associations exponentially increases according to their LHS length. Also, many of the long (and very confident) associations are covered by shorter ones, thus including them in the models is unnecessary. Determining values for weighted features of CRFs in our method is quite heuristic but reasonable because we tend to highlight rare associations by pretending to increase their occurrence frequency in the training data.

Computational time in our method is also one of the important aspects. Because important associations of statistics are discovered prior to training CRFs, the time complexity of discovering important associations and time complexity of training CRFs are independent. The former is proportional to the complexity of the FP-growth algorithm for mining frequent patterns from training data. The authors of FP-growth only reported the frequent-pattern mining time and the comparison with the other mining algorithms (e.g., Apriori) rather than pointing out explicitly the time complexity of the algorithm. The comparison showed that the discovery time of FP-growth is very efficient in comparison with the Apriori and TreeProjection algorithm, especially when the support threshold is small. In our algorithm, we needed only 2 and 1.5 hours for discovering and filtering important associations of statistics from phrase-chunking and named entity recognition datasets, though these linguistic datasets are large enough and the support threshold is very small in comparison with the databases' sizes. Time complexity of training CRFs is also large, but it is the common problem that our method and other methods incur. The time complexity of training first-order Markov CRFs is $\mathbf{O}(ma|\boldsymbol{L}|^2\mathrm{NT}_e)$, in which m is the number of training iterations, a is the average number of active context predicates, $\boldsymbol{L}$ is the set of class labels, N is the total of sequences in the training dataset, and T_e is the average length of each training sequence. Training a CRF model is very costly because we have to perform a very costly forward-backward computation on a series of matrices in order to evaluate the log-likelihood function of the CRF model and its gradient vector at every training iteration. The complexity of the algorithm using important associations of statistics as constraints for the inference process of CRFs was reported in Section 4.3.

The experimental results reported in this article do not represent the best possible performances on phrase-chunking and named entity recognition because (1) our feature templates are relatively simple to keep the set of features compact; this is convenient for mining associations, training again and again while conducting the experiments; (2) unlike the CRF model in Sha and Pereira [2003], all our CRF models obey the first-order Markov property to reduce the complexity.

6. RELATED WORK

Discriminative (sequential) learning models have been applied successfully in different natural language processing and information extraction tasks, such as POS tagging [Ratnaparkhi 1996], text-chunking [Peng et al. 2004; Sha and Pereira 2003], information extraction [Kristjansson et al. 2004; Pinto et al. 2003], computer vision and image analysis [Kumar and Hebert 2003; He et al. 2004; Torralba et al. 2004], and biological modeling [Yeo and Bruge 2003]. Normally, one can extract features from sequential data within a relatively large window size (i.e., the history size of contextual information) and make high-order combinations of atomic observational tests (e.g., the conjunctions of two or three consecutive words in a sentence) in the hope that they will capture as many useful predictive clues as possible. Unfortunately, such useful conjunctions are sparsely distributed in the feature space, and thus one unintentionally includes a large number of redundant conjunctions into the model. Inspired by this obstacle, our work aims at picking up useful conjunctions from a large array of conjunction candidates while keeping the set of features simple. The data-driven search with respect to support and confidence factors based on association rule mining techniques can discover desired conjunctions with an acceptable computational time.

There have been several previous studies about the discovery of interesting and/or exceptional patterns in databases [Liu et al. 1999; Padmanabhan and Tuzhilin 1998; Silberchats and Tuzhilin 1996; Suzuki and Shimura 1996]. The concept of interesting patterns/rules varies in different papers but, in general, implies the associations of weak statistics that have small support but large confidence. However, most of the work focused on finding exception rules for data analysis purposes rather than using them for improving classification accuracy. Also, these studies worked with numerical data instead of linguistic data like ours. Discovering important associations of statistics in linguistic data for improving discriminative sequential learning was actually our original proposal.

McCallum [2003] proposed an automated feature induction for CRFs that can reduce the number of used features dramatically. This likelihood-driven approach repeatedly adds features with high-likelihood gains into the model. The set of induced features contains both atomic observational tests and conjunctions of them. The main difference between this work and ours is that McCallum focuses on features with high-likelihood gains in order to reduce the number of used features as much as possible, while the main target of our method is to discover important associations or co-occurrences of weak statistics

from the training data to highlight difficult examples. Further, our method can examine any combination or conjunction of statistics because of the exhaustive working method of association rule mining techniques.

An error-driven method that combines boosting technique into the training process of CRFs [Altun et al. 2003] minimizes an upper bound on the ranking loss that was adapted to label sequences. This method also focuses on hard observation sequences, but without integrating new useful conjunctions of basic features. Another boosting-like training for CRFs is based on the use of a gradient tree [Dietterich 2004] to learn many conjunctions of features. One problem is that this method requires adding many trees for the training process.

7. CONCLUSIONS AND FUTURE WORK

In this article, we proposed a data-driven approach that can discover and highlight important associations or co-occurrences of singleton statistics from the sequential training data to deal with hard examples. Discovered associations are integrated into the exponentially-trained sequential learning models as normal features, features with weighted values, and constraints for the inference process. The experimental results show that important associations can improve the model performance by fighting against the dominance of singleton but common statistics in the training data.

Though important associations can enhance the prediction accuracy for hard examples, our approach is currently based on the occurrence frequency of statistics and the existence of important associations in the training data. We believe that there is an indirect theoretical relation between the occurrence frequencies of statistics and the learned weights of the model's features. Our future work will focus on this potential relation to estimate the extent to which useful patterns (e.g., important associations) discovered from the training data can improve the performance of discriminative (sequential) learning models.

ACKNOWLEDGMENTS

We would like to thank Dr. Bart Goethals, Department of Math and Computer Science, Antwerpen University, for sharing his lightweight and efficient implementation of the FP-growth algorithm. We would like to say thank you to Prof. Jorge Nocedal, Department of Electrical and Computer Engineering, School of Engineering and Applied Science, Northwestern University, the author of FORTRAN implementation of the L-BFGS optimization procedure. We also would like to thank Prof. Sunita Sarawagi, KR School of Information Technology, IIT Bombay, the author of the Java CRFs package, which is the precursor of our C/C++ CRFs toolkit.

REFERENCES

AGRAWAL, R. AND SRIKANT, R. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference Very Large Data Bases (VLDB)*. 487–499.

ALTUN, Y., HOFMANN, T., AND JOHNSON, M. 2002. Discriminative learning for label sequences via boosting. In *Proceedings of Neural Information Processing Systems (NIPS)*.

ANDO, R. AND ZHANG, T. 2005. A high-performance semi-supervised learning methods for text chunking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*. 1–9.

BERGER, A., PIETRA, A. D., AND PIETRA, J. D. 1996. A maximum entropy approach to natural language processing. *Computat. Linguis. 22*, 1, 39–71.

CARRERAS, X. AND MARQUEZ, L. 2003. Phrase recognition by filtering and ranking with perceptrons. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*. 205–216.

CHEN, S. F. AND ROSENFELD, R. 1999. A gaussian prior for smoothing maximum entropy models. Tech. Rep. CMU-CS-99-108. Carnegie Mellon University.

CHIEU, H. L. AND NG, H. T. 2003. Named entity recognition with a maximum entropy approach. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*. 160–163.

COLLINS, M. 2002. Discriminative training methods for hidden markov models: theory and experiment with perceptron algorithms. In *Proceedings of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*. 1–8.

DAUMÉ, III H. AND MARCU, D. 2005. Learning as search optimization: approximate large margin methods for structured prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*.

DIETTERICH, T. G. 2004. Training conditional random fields via gradient tree boosting. In *Proceedings of the 21th International Conference on Machine Learning (ICML)*. 169–176.

FLORIAN, R., ITTYCHERIAH, A., JING, H., AND ZHANG, T. 2003. Named entity recognition through classifier combination. In *Proceedings of Conference on Natural Language Learning (CoNLL)*. 168–171.

FREUND, Y. AND SCHAPIRE, R. 1997. A decision-theoretic generalization of on-line learning and application to boosting. *J. Comput. Syst. Sci. 55*, 119–139.

KLEIN, D., SMARR, J., NGUYEN, H., AND MANNING, C. D. 2003. Named entity recognition with character-level models. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*. 180–183.

KRISTJANSSON, T., CULOTTA, A., VIOLA, P., AND MCCALLUM, A. 2004. Interactive information extraction with constrained conditional random fields. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*. 412–418.

KUDO, T. AND MATSUMOTO, Y. 2001. Chunking with support vector machines. In *Proceedings of the second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 1–8.

KUMAR, S. AND HEBERT, M. 2003. Discriminative random fields: a discriminative framework for contextual interaction in classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1150–1157.

HAN, J., PEI, J., AND YIN, Y. 2000. Mining frequent patterns without candidate generation. In *Proceedings of the ACM International Conference on Management of Data (ACM SIGMOD)*. 1–12.

HAN, J., PEI, J., YIN, Y., AND MAO, R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Disc. 8*, 53–87.

HE, X., ZEMEL, R. S., AND CARREIRA-PERPINAN, M. A. 2004. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 695–702.

LAFFERTY, J., MCCALLUM, A., AND PEREIRA, F. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*. 282–289.

LI, W., HAN, J., AND PEI, J. 2001. Accurate and efficient classifications based on multiple class-association rules. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. 369–376.

LIU, D. AND NOCEDAL, J. 1989. On the limited memory BFGS method for large-scale optimization. *Math. Program. 45*, 503–528.

LIU, B., HSU, W., AND MA, Y. 1998. Integrating classification and association rule mining. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*. 80–86.

LIU, B. 1999. Finding interesting patterns using user expectations. *IEEE Trans. Knowl. Data Eng. 11*, 817–832.

MALOUF, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of Conference on Computational Natural Language Learning (CoNLL)*. 1–7.

MCCALLUM, A., FREITAG, D., AND PEREIRA, F. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of 17th International Conference on Machine Learning (ICML)*. 591–598.

MCCALLUM, A. 2003. Efficiently inducing features of conditional random fields. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI)*. 403–410.

PADMANABHAN, B. AND TUZHILIN, A. 1998. A belief-driven method for discovering unexpected patterns. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*. 94–100.

PENG, F., PENG, F., AND MCCALLUM, A. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*.

PIETRA, S. D., PIETRA, V. D., AND LAFFERTY, J. 1997. Inducing features of random fields. *IEEE Trans. Pattern Analys. Mach. Intell. 19*, 4, 380–393.

PINTO, D., MCCALLUM, A., WEI, X., AND CROFT, W. B. 2003. Table extraction using conditional random fields. In *Proceedings of the 26th ACM International Conference on Information Retrieval (ACM SIGIR)*. 235–242.

PHAN, X. H., NGUYEN, L. M., HO, T. B., AND HORIGUCHI, S. 2005. Improving discriminative sequential learning with rare-but-important associations. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*. 304–313.

RABINER, L. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE 77*, 2, 257–286.

RATNAPARKHI, A. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

SHA, F. AND PEREIRA, F. 2003. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology / The North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*. 134–141.

SILBERSCHATS, A. AND TUZHILIN, A. 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. Knowl. Data Eng. 8*, 970–974.

SUZUKI, E. 1997. Autonomous discovery of reliable exception rules. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*. 259–262.

SUZUKI, E. AND SHIMURA, M. 1996. Exceptional knowledge discovery in databases based on information theory. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*. 295–298.

TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. 2004. Contextual models for object detection using boosted random fields. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*.

YEO, G. AND BURGE, C. B. 2003. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. In *Proceedings of Conference. on Computational Molecular Biology*. 322–331.

ZHANG, T., DAMERAU, F., AND JOHNSON, D. 2002. Text chunking based on a generalization of winnow. *J. Mach. Learn. Res. 2*, 615–637.

Received September 2005; revised June 2006; accepted October 2006