Improving Discriminative Sequential Learning with Rare–but–Important Associations

Hieu Phan, Minh Nguyen
Japan Advanced Institute of Science and Technology
1–1, Asahidai, Tatsunokuchi, Ishikawa
923–1292, Japan
{hieuxuan, nguyenml}@jaist.ac.jp

Susumu Horiguchi
Tohoku University
Aobayama 09, Arimaki Sendai
980–8579, Japan
susumu@ecei.tohoku.ac.jp

Bao Ho
Japan Advanced Institute of Science and Technology
1–1, Asahidai, Tatsunokuchi, Ishikawa
923–1292, Japan
bao@jaist.ac.jp

ABSTRACT

Discriminative sequential learning models like Conditional Random Fields (CRFs) have achieved significant success in several areas such as natural language processing, information extraction, and computational biology. Their key advantage is the ability to capture various non–independent and overlapping features of inputs. However, there are several unexpected pitfalls influencing negatively on model's performance that mainly come from the imbalance among classes/labels, the irregular phenomena, and the ambiguity potentially existing in the training data. This paper presents a data–driven approach that can deal with such hard–to–predict data instances by discovering and emphasizing rare–but–important associations of statistics hidden in the training data. Mined associations are then incorporated into these models in a couple of ways to boost difficult examples. The experimental results of English 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 training corpora/datasets.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

Discriminative sequential learning, feature selection, information extraction, text segmentation, association rule

1. INTRODUCTION

Conditionally-trained or discriminative models like Maximum Entropy (MaxEnt) [3], Discriminative HMMs [5], Maximum Entropy Markov Models (MEMMs) [18], and CRFs [13] have earned significant success in many (sequential) labeling and segmenting tasks, such as part–of–speech (POS) tagging [24], text segmentation or shallow parsing [20, 25], information extraction [8, 22], object detection in computer vision [26], image analysis and labeling [10, 12], and biological sequence modeling [27]. The noticeable advantage of these models is their flexibility to integrate a variety of arbitrary, overlapping, and non–independent 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, being up to millions, e.g., in [25]. This is because one usually utilizes complex templates including conjunctions of atomic context predicates, e.g., 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 should be heavily overfitting and time–consuming to train because they contains many teacher–specific and redundant features. To reduce these drawbacks, McCallum [19] proposed a likelihood–driven feature induction for CRFs that is based on a famous feature inducing strategy for exponential models [21]. This method iteratively adds conjunctions of atomic observational tests most increasing conditional log–likelihood into the model until a stopping criteria is reached. In spite of attaining a trade–off between the number of used features and the model accuracy, this strategy may ignore rare but sensitive conjunctions with smaller likelihood gains but critical to the model performance. Also, when the number of atomic context predicates is large, the number of conjunctions becomes explosive; and thus ranking all conjunctions by likelihood gain is really expensive.

In this paper, we propose a data–driven approach that can identify and emphasize rare–but–important associations or co–occurrences of statistics[1] hidden the training data to improve prediction accuracy for hard–to–classify instances. The main motivation and the underlying idea of this ap-

[1]In this paper, 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

proach are based on the fact that (sequential) data, such as natural language or biological information, potentially contain the following phenomena that should 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 to Japan* is a singular verb or a plural noun.

- **Irregular instances** are recognized as exceptions that do not obey the common statistics or decisions. For example, a POS tagger may mark *walk* as a noun in the sentence *The disabled walk very hard* because of a regular sequential dependency that a noun should go after an adjective. However, the correct interpretation is that *The disabled* (i.e., *The disabled people*) 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 (NP) is much larger than the other phrase types, e.g., adjective phrase (ADJP). This may lead to low prediction accuracy for minor classes due to the dominance of major ones.

- **Frequently–observed vs. less–observed data instances**: For instance, a named entity recognizer may identify *New York University* as a location while it is 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 big size of sliding window to capture such useful clues because of generating too many irrelevant features.

Data instances falling into the above situations should be hard examples. Thus, the prediction of their labels does not usually obey the frequently observed statistics. In other words, the simple aggregation of singleton context predicates may lead to misleading predictions because the common statistics always overwhelm uncommon ones. To overcome this pitfall, a model should rely on rare–but–important associations or conjunctions of singleton context predicates to win the dominance of common decisions. In the first example, most contextual supports surrounding *plans* (e.g., *trip* is a singular noun, *plan**s*** ends with ***s***, and the next word is *to*) tend to say that *plans* is a singular verb rather than a part of a noun phrase. It is, however, quite easy for the model to recognize *plans* as a plural noun if relying on an important association like "if the word after *plans to* is initially capitalized, then *plans* should be a plural noun". This association rule emphasizes a rare but important co–occurrence of three factors: *phans*, *to*, and the next word is initially capitalized (i.e. like a location such as a city or a country rather than an infinitive). Although such kind of associations may occur just several times in a whole dataset, their appearance is an important source of evidence to deal with hard instances.

In spite of their benefit, searching for all rare–but–important associations of singleton statistics in big datasets is challenging because the number of candidates is prohibitively large. Fortunately, we find that association rule mining techniques, such as FP-growth [11], are really useful for discovering such kind of patterns. In our method, the set of rare–but–important associations is a special subset of rare but highly confident association rules discovered in the training data. Selected associations are then integrated into the learning process according three ways to improve the prediction accuracy for hard instances: (a) associations as normal features, (b) associations as normal features with weighted feature values, and (c) associations as constraints for the inference process.

Derived from the reasonable assumption about rare–but–important associations and the robustness of association rule mining techniques, our approach offers the following distinctive characteristics: (1) rare–but–important associations are globally discovered from a huge space of candidates with any length and any combination of singleton statistics; (2) models with those associations can deal with difficult instances while preventing overfitting by avoiding long and fixed conjunction templates; (3) ones can choose a suitable way to incorporate selected associations into their models. Particularly, 100%–confidence associations can be integrated into the model in terms of constraints for inference; and (4) our method can be used to improve any discriminative sequential learning application, especially for highly ambiguous and imbalanced data. Finally, our work also highlights a potential connection between pattern mining and statistical learning from large datasets.

The remaining of the paper is organized as follows. Section 2 briefly introduces linear–chain CRFs, a typical sequential learning model. Section 3 mainly presents the proposed approach. Section 4 describes the experimental results and some discussion. Section 5 mentions the related work. Finally, the conclusions are given in Section 6.

2. DISCRIMINATIVE SEQUENTIAL LEARNING

The goal of labeling/tagging for sequential data is to learn a mapping from observation sequences to their corresponding label sequences, e.g., the sequence of POS tags for words in a sentence. Discriminative HMMs [5], MEMMs [18], and CRFs [13] were intentionally designed for such sequential learning applications. In contrast to generative models like HMMs [23], these models are discriminative, i.e. trained to predict the most likely label sequence given the observation sequence. In this paper, CRFs are referred to as the undirected linear–chain of model states, i.e. conditionally–trained finite state machines (FSMs), that obey the first–order Markov independence assumption. The strength of CRFs is that it can combine both the sequential property of HMMs and the philosophy of MaxEnt as well as the global normalization that can avoid the *label–bias problem* [13]. 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 $\mathcal{S}$ be a set of FSM states, each of which is associated with a label, $l \in \mathcal{L}$. Let $\mathbf{s} = (s_1, s_2, \ldots, s_T)$ be some state sequence, CRFs [13] define the conditional probability of a

state sequence given an observation sequence as

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

where $Z(\mathbf{o}) = \sum_{\mathbf{s}'}\exp\left[\sum_{t=1}^{T}\sum_{k}\lambda_k f_k(s'_{t-1}, s'_t, \mathbf{o}, t)\right]$ 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:

$$f_k^{(per-state)}(s_t, \mathbf{o}, t) = \delta(s_t, l)\chi_k(\mathbf{o}, t), \quad (2)$$

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

where δ denotes the Kronecker–δ. A per–state feature (2) combines the label l of current state s_t and a context predicate, i.e. the binary function $\chi_k(\mathbf{o}, t)$, 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, e.g., the previous label l' = JJ and the current label l = NN (noun).

2.2 Inference in CRFs

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

$$\begin{aligned}\mathbf{s}^* &= \text{argmax}_{\mathbf{s}}\ p_\theta(\mathbf{s}|\mathbf{o}) \\ &= \text{argmax}_{\mathbf{s}}\left\{\exp\left[\sum_{t=1}^{T}\sum_{k}\lambda_k f_k(s_{t-1}, s_t, \mathbf{o}, t)\right]\right\} \quad (4)\end{aligned}$$

In order to find $\mathbf{s}^*$, one can apply dynamic programming technique with a slightly modified version of the original Viterbi algorithm for HMMs [23]. 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 $\varphi_t(s_i)$ $(0 \le t \le T-1)$ and $\varphi_0(s_i)$ to be the probability of starting in each state s_i. The recursion is given by:

$$\varphi_{t+1}(s_i) = \max_{s_j}\left\{\varphi_t(s_j)\exp\left[\sum_{k}\lambda_k f_k(s_j, s_i, \mathbf{o}, t)\right]\right\} \quad (5)$$

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

2.3 Training CRFs

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

$$L = \sum_{j=1}^{N}\log\ p_\theta(\mathbf{l}^{(j)}|\mathbf{o}^{(j)}) - \sum_{k}\frac{\lambda_k^2}{2\sigma^2}, \quad (6)$$

where the second sum is a Gaussian prior over parameters with variance σ^2, that provides smoothing to deal with sparsity in the training data [4].

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 [19]. 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 [15], are more efficient than the others [17, 25]. 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 D, then the log–likelihood gradient component of λ_k is

$$\begin{aligned}\frac{\delta L}{\delta \lambda_k} &= \left[\sum_{j=1}^{N} C_k(\mathbf{s}^{(j)}, \mathbf{o}^{(j)})\right] - \\ &\quad \left[\sum_{j=1}^{N}\sum_{\mathbf{s}} p_\theta(\mathbf{s}|\mathbf{o}^{(j)})C_k(\mathbf{s}, \mathbf{o}^{(j)})\right] - \frac{\lambda_k}{\sigma^2}, \quad (7)\end{aligned}$$

where $C_k(\mathbf{s}, \mathbf{o})$ is the count of feature f_k given $\mathbf{s}$ and $\mathbf{o}$, equal to $\sum_{t=1}^{T} f_k(s_{t-1}, s_t, \mathbf{o}, t)$, i.e. 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 correspond to the difference between the empirical and the model expected values of feature f_k. The last term is the first–derivative of the Gaussian prior.

3. IMPROVING DISCRIMINATIVE SEQUENTIAL LEARNING

This section presents the proposed framework in details: (1) how to discover rare–but–important associations from the training data and (2) how to integrate those associations in to discriminative sequential learning models, e.g. CRFs.

3.1 Mining Rare–but–Important Associations

This section first presents the concept rare–but–important associations in discriminative sequential learning based on the traditional association rules [1], and then describes the method to discover such patterns from the training data.

3.1.1 Associations in Sequential Training Data

Recall to the training dataset for sequential learning: $\mathcal{D} = \{(\mathbf{o}^{(k)}, \mathbf{l}^{(k)})\}_{k=1}^{N}$ where $\mathbf{o}^{(k)}$ and $\mathbf{l}^{(k)}$ are the k^{th} data observation and label sequences, respectively. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_M\}$ be the set of M context predicate templates in which each A_i $(1 \le i \le M)$ captures a particular type of contextual information about data observations. In a sense, $\mathcal{A}$ is similar to the set of attributes in a relational table. Applying all predicate templates in $\mathcal{A}$ to each position in every training sequence $(\mathbf{o}^{(k)}, \mathbf{l}^{(k)})$ in the training data $\mathcal{D}$, we obtain a transactional database $\mathcal{TD}$ in which each transaction consists of a label and a list of active context predicates.

For example, the first part of Table 1 shows the training data $\mathcal{D}$ for POS tagging in which each training sequence ($\mathbf{o}$,$\mathbf{l}$) is an English sentence together with POS tags of words. The second part is a set of 4 context predicate templates: the identities of the previous word (w_{t-1}), the current word (w_t), the next word (w_{t+1}), and the 2-character suffix of the previous word (suf2(w_{i-1})). The third part is the transactional database $\mathcal{TD}$ after applying templates in $\mathcal{A}$ for $\mathcal{D}$.

Let $\mathcal{I} = \{\chi_1, \chi_2, \ldots, \chi_n\}$ be the set of all possible context predicates in the transactional database $\mathcal{TD}$, let $\mathcal{L}$ be the

Table 1: Transactional database of POS tagging data

Sequential Training Data $\mathcal{D}$
$(\mathbf{o}^{(k-1)}, \mathbf{l}^{(k-1)})$: ...
$(\mathbf{o}^{(k)}, \mathbf{l}^{(k)})$: ... highly_RB ambiguous_JJ data_NNS ...
$(\mathbf{o}^{(k+1)}, \mathbf{l}^{(k+1)})$: ...
C.P. templates $\mathcal{A} = \{w_{t-1}, w_t, w_{t+1}, \text{suf2}(w_{t-1})\}$
Transactional Database $\mathcal{TD}$
...
RB, ... w_t:highly, w_{t+1}:ambiguous, ...
JJ, w_{t-1}:highly, w_t:ambiguous, w_{t+1}:data, suf2(w_{t-1}):ly
NNS, w_{t-1}:ambiguous, w_t:data, ..., suf2(w_{t-1}):us
...

set of all labels, and $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ be the set of all transactions in $\mathcal{TD}$. Our target is to examines predictive association rules [1] having the form below,

$$\text{Predictive association rule } r\text{:} \quad X \Rightarrow l \tag{8}$$

where the left hand side (LHS) of r, $X = \langle \chi_{i1} \wedge \chi_{i2} \wedge \ldots \wedge \chi_{ip} \rangle \subset \mathcal{I}$, is a conjunction of p context predicates in $\mathcal{I}$, and the right hand side (RHS) of rule r, i.e. $l \in \mathcal{L}$, is a particular label. The support of the rule r, denoted as $sup(r)$, is the number of transactions in $\mathcal{T}$ contains $\{l\} \cup X$, and the confidence of r, denoted as $conf(r)$, is the conditional probability that a transaction in $\mathcal{T}$ has the label l given that it contains X, i.e. $conf(r) = sup(X \cup \{l\})/sup(X)$. In a sense, this kind of rules is similar to the associative classification rules in [14, 16] except that our work mainly focuses on rare–but–important associations discussed in the next section.

3.1.2 Rare–but–Important Associations

Derived from the predictive association rules defined in (8) and the concepts of support and confidence factors, we have a descriptive definition of rare–but–confident associations below,

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

$$lsup \leq supp(r) \leq usup \text{ and } conf(r) \geq lconf$$

All predictive association rules satisfying definition (1) are rare–but–confident. However, NOT all of them are important. This is based on an important observation that: "if most context predicates in the LHS of a rare–but–confident rule r strongly support for the label l, then the rule r is trivial". In other words, if most context predicates in the LHS of r largely support for label l in a separated manner, there is no need to examine the co-occurrence of all items in the LHS, and the model can still work properly without this rule. For example, in named entity recognition, the rule $\langle w_{t-1}$:New $\wedge$ w_t:York $\wedge$ w_{t+1}:weather $\Rightarrow$ $label_t$=LOCATION$\rangle$ is not important because both "w_{t-1}:New" and "w_t:York" strongly support for the label "LOCATION", and thus their conjunction should be unnecessary. In other words, the named entity recognizer can predict the label "LOCATION" for the word "York" without the above rule because both "New" and "York" are frequently observed in the training data as a location name, i.e. "New York". Based on this observation, we define the concept of "rare–but–important" associations as follows,

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

$$\exists l' \in \mathcal{L} : \sum_{\chi \in X} sup(\chi \Rightarrow l') > \sum_{\chi \in X} sup(\chi \Rightarrow l)$$

Why association rules satisfying definition (2) are important? Intuitively, if a such rule, r, exists in the training data but not being discovered and emphasized, the model may predict the label l' for any data instance/transaction holding all context predicates in the LHS of r whereas the correct label is l. This is because most singleton context predicates in LHS of r tend to support for the label l' rather than l. This is why the appearance of rules satisfying definition (2) is important.

For instance, rule $\langle w_{t-1}$:New $\wedge$ w_t:York $\wedge$ w_{t+1}:University $\Rightarrow$ $label_t$=ORGANIZATION$\rangle$ is important for recognizing the named entity type of the current word ("York") since there is another label, "LOCATION", that should satisfy the condition addressed in definition (2), i.e., $\sum_{\chi \in \mathbf{X}} supp(\chi \Rightarrow$ LOCATION$) > \sum_{\chi \in \mathbf{X}} supp(\chi \Rightarrow$ ORGANIZATION$)$. This is because both *New* and *York* strongly support for the label "LOCATION" rather than for "ORGANIZATION". Thus, the appearance of the above rule can help the model to recognize "New York University" as an ORGANIZATION instead of a LOCATION.

3.1.3 Discovering Rare–but–Important Associations

Mining rare–but–important associations from the transactional database $\mathcal{TD}$ faces following problems: (1) the number of data items, i.e. the number of atomic context predicates and labels $|\mathcal{I} \cup \mathcal{L}|$, is relatively large; (2) the support thresholds, i.e. $lsup$ and $usup$, are very small compared to the number of transactions $|\mathcal{T}|$. This means that there are a huge number of combinations of items that must be examined during the mining process.

Fortunately, FP–growth [11], a frequent pattern mining algorithm without candidate generation, can discover such associations in an acceptable computational time. This 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 with a partitioning, recursive fashion without candidate generation. See [11] for a complete description of this algorithm.

Taking the sequential training data $\mathcal{D} = \{(\mathbf{o}^{(k)}, \mathbf{l}^{(k)})\}_{k=1}^{N}$, the set of context predicate templates $\mathcal{A} = \{A_1, A_2, \ldots, A_M\}$, the lower and upper support thresholds $lsup$, $usup$ ($lsup \leq usup \ll |\mathcal{T}|$), and the lower confidence threshold $lconf$ ($0 \leq lconf \leq 1$ and $lconf \simeq 1$) as inputs, The rare–but–important association mining includes the following steps:

1. Transforming the sequential training data $\mathcal{D}$ to a transactional database $\mathcal{TD}$ by applying all predicate templates in $\mathcal{A}$. $\mathcal{TD}$ includes the set of items $\mathcal{I} \cup \mathcal{L}$ (all

possible generated context predicates and all labels), the set of all transactions $\mathcal{T}$.

2. Mining all itemsets with supports larger or equal to $lsup$ using FP–growth algorithm.

3. Generating all rare–but–confident association rules in the form of $X \Rightarrow l$ (8) with supports belonging to [$lsup$, $usup$], and the minimum confidence threshold $lconf$.

4. Selecting all possible rare–but–important association rules from rare–but–confident ones by applying the condition stated in the definition (2).

In the fourth step, to determine whether or not a rare–but–confident rule, r: $X \Rightarrow l$, is rare–but–important, we have to scan over the database to compute the sums of supports of context predicates in the LHS of r for all other labels. This is an expensive operation. Fortunately, we can perform this on the FP–tree by traversing node–links of each label, which starts at the header table, and keeping looking upward, downward to count the supports from context predicates appearing in the LHS of r. See [11, 14] for the detailed description of FP–tree.

3.2 Incorporating Rare–but–Important Associations into Conditional Random Fields

This section presents three ways to incorporate rare–but–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.

3.2.1 Rare–but–Important Associations as Normal Features of Conditional Random Fields

All rare–but–important associations are in the form $X \Rightarrow l$, in which $X = \langle \chi_{i1} \wedge \chi_{i2} \wedge \ldots \wedge \chi_{ip} \rangle$ ($\subset \mathcal{I}$) is a conjunction of p context predicates and $l \in \mathcal{L}$ is a particular label. These associations can be integrated into CRFs in terms of normal per–state features as follows.

$$f_k^{(per-state)}(s_t, \mathbf{o}, t) = \delta(s_t, l)\{\chi_{i1}(\mathbf{o}, t) \wedge \ldots \wedge \chi_{ip}(\mathbf{o}, t)\}$$

These per–state features are similar to those in (2) except that they capture a co–occurrence of p atomic context predicates rather than a single one. There features are treated as normal features and are trained together.

3.2.2 Rare–but–Important Associations as Normal Features with Weighted Feature Values

It is noticeable that rare–but–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, i.e. frequently observed singleton features. To overcome this drawback, we emphasize rare–but–important features by assigning larger feature function values comparing to the normal features.

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

where $\delta(s_t, l)$ and $\{\chi_{i1}(\mathbf{o}, t) \wedge \ldots \wedge \chi_{ip}(\mathbf{o}, t)\}$ 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 rare–but–important association) is small. Thus, for each feature generated from a rare–but–important association r, v equals to $(usup - sup(r) + 2)$. This ensures that v is always bigger than 1 and inversely proportional to the support of r, i.e. the occurrence frequency of the feature.

3.2.3 Rare–but–Important Associations as Constraints for the Inference Process

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. [8] proposed this extension with the application to interactive form filling in which users can examine the filling process and make necessary re–corrections in terms of user constraints. A re–correction applied at a particular position will propagate though the Viterbi sequence to make automatic updates for labels at other positions, i.e. the correction propagation capability.

This section presents the integration of rare–but–important associations with 100%-confidence into the Viterbi algorithm in terms of data–driven constraints to make the corrections directly to the inference process of CRFs. Unlike those used in [8], our constraints are 100%–confidence associations and are automatically discovered from the training data.

Normally, CRFs use a variant of 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 the dynamic programming technique with a forward variable $\varphi_{t+1}(s_i)$ in definition (5).

Let $R = \{r_1, r_2, \ldots, r_q\}$ be a set of q rare–but–important associations with 100%–confidence, and each r_u ($1 \leq u \leq q$) has the form $\langle \chi_{u1} \wedge \chi_{u2} \wedge \ldots \wedge \chi_{up} \rangle \Rightarrow l_u$ ($l_u \in \mathcal{L}$). Each $r_u \in R$ 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 context predicates at the current position holds the LHS of any rule $r_u \in R$. 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 to zero. The constrained forward variable is re-defined as follows.

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

The constraint applied at the time position t will propagate though the whole sequence and make some re–corrections 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 rare–but–important association rules) 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.

Let $R = \{r_1, r_2, \ldots, r_q\}$ be the set of 100%–confidence rules or also known as constraints, let $X = \{\chi_1, \chi_2, \ldots, \chi_m\}$ be the set of m active context predicates observed at the

current position t. The target of the following algorithm is to check whether or not X holds the LHS of any constraint $r_u \in R$. If yes, choose the constraint with the longest LHS.

1. For each $\chi_i \in X$, lookup the set of constraints $R_i \subset R$ in which the LHS of every constraint in R_i contains χ_i. Denote $R' = \{R_1 \cup R_2 \cup \ldots \cup R_m\}$.

2. For each constraint $r_j \in R'$, let c_j be the sum of occurrence frequency of r_j in R_1, R_2, ..., R_m.

3. Find the pair $\langle r_j, c_j \rangle$ $(1 \leq j \leq m)$ such that c_j is the largest number satisfying the condition: c_j equals to the number of all context predicates in the LHS of r_j.

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

4. EXPERIMENTAL RESULTS

4.1 Experimental Settings

All the experiments were performed with our C/C++ implementation of CRFs – FlexCRFs[2] – on 2.5GHz, 1Gb RAM, Pentium IV processor with RedHat Linux. All CRF models were trained using the limited–memory quasi–Newton method for unconstrained optimization, L–BFGS [15]. Unlike those used in [25], our CRF models are more simple and easier to implement by obeying the first–order Markov property, i.e., the label of the current state only depends on the label of the previous state.

Training and testing data for English phrase chunking and named entity recognition can be found at the shared tasks of CoNLL2000[3] and CoNLL2003[4], respectively.

4.2 Phrase Segmentation

Phrase chunking, an intermediate step towards full parsing of natural language, is to identify 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]."

4.2.1 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 (8936 sentences, 211727 tokens) and section 20 as testing data (2012 sentences, 47377 tokens). Each line in the annotated data is for a token and consists of three columns: the token (a word or a punctuation), 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–⟨PharseType⟩), or continues a phrase (I–⟨PhraseType⟩). For example, the label sequence of the above 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".

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

This dataset contains 11 phrase types as shown in the first column of Table 3. Two consecutive data sequences (sentences) are separated by a blank line.

4.2.2 Feature Selection

On the phrase chunking dataset, we use feature templates as shown in Table 2. 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 a context predicate ($\chi(\mathbf{o}, t)$) observed surrounding the current position t influences on the label (l) of the current state. A context predicate captures a particular property of the observation sequence. For instance, the per–state feature "l = I–NP" and "word$_{t-1}$ 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 3 describes both transition and per–state feature templates. Context predicates for per–state features are identities of words, POS tags of words surrounding the current position t, such as words and POS tags at positions $t-2$, $t-1$, t, $t+1$, $t+2$ (i.e., window size is 5).

Table 2: Feature templates for phrase chunking

Transition feature templates	
Current state: s_t	Previous state: s_{t-1}
l	l'

Per–state feature templates	
Current state: s_t	Context predicate: $\chi(\mathbf{o}, t)$
l	w_{t-2}; w_{t-1}; w_t; w_{t+1}; w_{t+2}
	$w_{t-1} \wedge w_t$; $w_t \wedge w_{t+1}$
	p_{t-2}; p_{t-1}; p_t; p_{t+1}; p_{t+2}
	$p_{t-2} \wedge p_{t-1}$; $p_{t-1} \wedge p_t$
	$p_t \wedge p_{t+1}$; $p_{t+1} \wedge p_{t+2}$
	$p_{t-2} \wedge p_{t-1} \wedge p_t$
	$p_{t-1} \wedge p_t \wedge p_{t+1}$
	$p_t \wedge p_{t+1} \wedge p_{t+2}$

We also employ the 2–order conjunctions of the current word with the previous ($w_{t-1} \wedge w_t$) or the next word ($w_t \wedge w_{t+1}$), the 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 3 and the feature rare threshold of 1, 321526 context predicates and 152856 CRF features were generated from 8936 training data sequences.

4.2.3 Mining Rare–but–Important Associations

Let $\mathcal{I}$ be the itemset of 321548 data items, i.e. the union set of 321526 context predicates and 22 phrase labels; $\mathcal{T}$ be the set of 211727 data transactions corresponding to 211727 tokens of the training data (the maximum transaction length is 20, i.e., 19 context predicate templates plus the label). Let lower support (*lsup*) and upper support (*usup*) thresholds be 4 and 8, respectively; the lower confidence (*lconf*) threshold be 0.98 or 98%. In fact, all output rules have the confidence of 100% because $lconf = 0.98 > \frac{7}{8}$ and therefore larger than all other confidence levels. We also confine the length of LHS of all rare–but–important associations between 3 and 6 because of an important observation that rules with LHS length smaller than 3 are too general and rules with LHS length larger than 6 are too specific.

The mining process for rare–but–important associations took 2 hours using FP–growth algorithm and the filter criteria presented in the definitions 1 and 2. The output was a set of 10364 rare–but–important associations with the length of LHS between 3 and 6, the support between 4 and 8, and the confidence 100%. This set of associations were integrated into the CRF model in terms of normal features, normal features with weighted feature values, and constraints for the inference process.

4.2.4 Results

Table 3: The performance of English phrase chunking without rare–but–important associations

Phrase	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
NP	12422	12399	11613	93.66	93.49	93.57
PP	4811	4832	4684	96.94	97.36	97.15
VP	4658	4690	4375	93.28	93.92	93.60
SBAR	535	538	459	85.32	85.79	85.55
ADJP	438	398	303	76.13	69.18	72.49
ADVP	866	864	686	79.40	79.21	79.31
PRT	106	95	75	78.95	70.75	74.63
LST	5	0	0	0.00	0.00	0.00
INTJ	2	1	1	100.0	50.00	66.67
CONJP	9	16	6	37.50	66.67	48.00
UCP	0	0	0	0.00	0.00	0.00
Avg1.				74.12	70.64	72.34
Avg2.	23852	23833	22202	93.16	93.08	93.12

Table 4: The performance of English phrase chunking with rare–but–important associations as normal features of CRFs

Phrase	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
NP	12422	12402	11645	93.90	93.74	93.82
PP	4811	4849	4686	96.64	97.40	97.02
VP	4658	4687	4381	93.47	94.05	93.76
SBAR	535	522	473	90.61	88.41	89.50
ADJP	438	416	337	81.01	76.94	78.92
ADVP	866	850	726	85.41	83.83	84.62
PRT	106	104	76	73.08	71.70	72.38
LST	5	3	1	33.33	20.00	25.00
INTJ	2	1	1	100.0	50.00	66.67
CONJP	9	11	7	63.64	77.78	70.00
UCP	0	0	0	0.00	0.00	0.00
Avg1.				81.11	75.39	78.14
Agv2.	23852	23845	22333	93.66	93.63	93.65

Table 3 shows the highest performance (achieved at the 48^{th} L–BFGS iteration) of the phrase chunking task trained on the original set of 152856 CRF features without rare–but–important associations. In each line, the first column is the phrase type; the second (#Hm.) is the number of human annotated phrases; the third (#Ml.) is the number of phrases automatically marked by the CRF model; the fourth (#Mt.) is the number of correct phrases marked by the model; the last three columns are precision (Pr.), recall (Rc.), and F1–measure (F1), respectively. The last two lines are the average performance calculated according two ways: precision–recall based and phrase based; The first is based on the precision and recall values of separated phrase types and the second is based on the average numbers of human–annotated, model, and correct phrases. The first average F1 (72.34%) reflects the balance and the trade–off among per–label performances while the second average F1 (93.12%) reflects the total performance.

Table 4, having the same format as Table 3, describes the performance of phrase segmentation in case that discovered rare–but–important associations were integrated into the CRF model as normal features. The highest average F1–measure achieved at the 45^{th} L–BFGS iteration is 93.65%, i.e. 0.53% higher than the original performance.

Table 5: The performance of English phrase chunking with rare–but–important associations as normal features with weighted feature values

Phrase	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
NP	12422	12398	11659	94.04	93.86	93.95
PP	4811	4851	4694	96.76	97.57	97.16
VP	4658	4683	4385	93.64	94.14	93.89
SBAR	535	524	473	90.27	88.41	89.33
ADJP	438	415	339	81.69	77.40	79.48
ADVP	866	853	726	85.11	83.83	84.47
PRT	106	103	79	76.70	74.53	75.60
LST	5	4	3	75.00	60.00	66.67
INTJ	2	1	1	100.0	50.00	66.67
CONJP	9	10	6	60.00	66.67	63.16
UCP	0	0	0	0.00	0.00	0.00
Avg1.				85.32	78.64	81.84
Agv2.	23852	23842	22365	93.81	93.77	93.79

Table 6: The performance of English phrase chunking with rare–but–important associations as constraints for inference

Phrase	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
NP	12422	12401	11662	94.04	93.88	93.96
PP	4811	4852	4697	96.81	97.63	97.22
VP	4658	4683	4385	93.64	94.14	93.89
SBAR	535	524	475	90.65	88.79	89.71
ADJP	438	419	342	81.62	78.08	79.81
ADVP	866	853	724	84.88	83.60	84.24
PRT	106	103	81	78.64	76.42	77.51
LST	5	4	3	75.00	60.00	66.67
INTJ	2	2	1	50.00	50.00	50.00
CONJP	9	9	7	77.78	77.78	77.78
UCP	0	0	0	0.00	0.00	0.00
Avg1.				82.31	80.03	81.15
Agv2.	23852	23850	22377	93.82	93.82	93.82

Table 5 shows the performance in the case that all rare–but–important associations were incorporated into the CRF model in the form of features with weighted feature values. The highest average F1–measure (at the 47^{th} iteration) is 93.79%, i.e. 0.67% higher than the original performance.

Table 6 describes the performance in the case that all 100%–confidence rare–but–important associations were used as constraints for the inference process. The highest average F1–measure is 93.82%, i.e. 0.70% higher than the original performance.

4.3 Named Entity Recognition

Named entity recognition (NER), a subtask of information extraction, is to identify names of persons (PER), organizations (ORG), locations (LOC), times (DATE/TIME),

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 ..."

4.3.1 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 14041 sentences (203621 tokens), and the testing data contains two parts: the development test set (testa: 3250 sentences, 51362 tokens) and the final test set (testb: 3453 sentences, 46435 tokens). The data files contain four columns separated by a blank space. Each token (a word or a punctuation) 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), inside a named entity (I–⟨NamedEntityType⟩). Only 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 ...".

4.3.2 Feature Selection

On the named entity recognition dataset, we use the feature templates shown in Table 7. All transition features also conform to the first Markovian property. Each context predicate for a per–state feature is one of the following types: (1) the identities of words (w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2}), (2) the POS tags of words (p_{t-2}, p_{t-1}, p_t, p_{t+1}, p_{t+2}), (3) the phrase tags of words (c_{t-2}, c_{t-1}, c_t, c_{t+1}, c_{t+2}), and (4) several simple regular expressions or formats of words such as "the first character of a word is capitalized" (IsInitialCapitalized), "all chars of a word are capitalized" (IsAllCapitalized), etc. Like the phrase chunking task, all context predicates are captured within a window with size of 5. Our feature templates are more simple than those used in the previous work presented at CoNLL2003 shared task and [19] in two ways: only five simple format properties were captured (compared to 16 regular expressions in [19]), and no external dictionaries were used such as the lists of people names, organization names, countries, cities, etc.

With the feature templates described in Table 7 and the feature rare threshold of 1, 125206 context predicates and 77826 CRF features were generated from 14041 training data sequences.

4.3.3 Mining Rare–but–Important Associations

Let $\mathcal{I}$ be the itemset of 125215 data items, i.e. the union set of 125206 context predicates and 9 named entity labels; $\mathcal{T}$ be the set of 203621 data transactions corresponding to 203621 tokens of the training data (the maximum transaction length is 41, i.e., 40 context predicate templates plus the label). Let lower support (*lsup*) and upper support (*usup*) thresholds be 4 and 8, respectively; the lower confidence (*lconf*) threshold be 0.98 or 98%. The lengths of LHSs of all rare–but–important associations are also confined between 3 and 6 to eliminate too general and too specific rules.

Table 7: Feature templates for NER

Transition feature templates	
Current state: s_t	Previous state: s_{t-1}
l	l'

Per–state feature templates	
Current state: s_t	Context predicate: $\chi(\mathbf{o}, t)$
l	w_{t-2}; w_{t-1}; w_t; w_{t+1}; w_{t+2}
	p_{t-2}; p_{t-1}; p_t; p_{t+1}; p_{t+2}
	c_{t-2}; c_{t-1}; c_t; c_{t+1}; c_{t+2}
	IsInitialCapitalized(w_k)
	IsAllCapitalized(w_k)
	IsNumber(w_k)
	IsAlphaNumber(w_k)
	IsFirstWord(w_k)
	where $k \in \{t-2, t-1, t, t+1, t+2\}$

The mining process for rare–but–important associations took about 1.5 hours using FP–growth algorithm and the filter criteria described in the definitions 1 and 2. The output was a set of 9023 rare–but–important associations with the length of LHS between 3 and 6, the support between 4 and 8, and the confidence 100%. This set of associations were integrated into the CRF model in terms of normal features, normal features with weighted feature values, and constraints for the inference process.

4.3.4 Results

Table 8: The performance of English named entity recognition without rare–but–important associations

NEType	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
ORG	1325	1254	1043	83.17	78.72	80.88
PER	1829	1806	1616	89.48	88.35	88.91
LOC	1832	1829	1636	89.45	89.30	89.37
MISC	916	852	735	86.27	80.24	83.14
Avg1.				87.09	84.15	85.60
Avg2.	5902	5741	5030	87.62	85.23	86.40

Table 9: The performance of English named entity recognition with rare–but–important associations as normal features of CRFs

NEType	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
ORG	1325	1256	1104	87.90	83.32	85.55
PER	1829	1811	1628	89.90	89.01	89.45
LOC	1832	1825	1647	90.25	89.90	90.07
MISC	916	855	757	88.54	82.64	85.49
Avg1.				89.14	86.22	87.66
Agv2.	5902	5747	5136	89.37	87.02	88.18

Table 8 shows the highest performance (F1 of 86.40%, achieved at the 133^{th} L–BFGS iteration) of the NER task trained on the original set of 77826 CRF features. This table has the same format as Table 3 except that the first column of each line is the named entity type.

Table 9, having the same format as Table 8, displays the experimental results of NER in case that all rare–but–important associations were integrated into CRF model in terms of normal features. The highest F1–measure is 88.18%, i.e., 1.78% higher than the original performance. Table 10 shows the results of NER in the case rare–but–important

Table 10: The performance of English named entity recognition with rare–but–important associations as normal features with weighted feature values

NEType	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
ORG	1325	1250	1108	88.64	83.62	86.06
PER	1829	1809	1633	90.27	89.28	89.77
LOC	1832	1827	1652	90.42	90.17	90.30
MISC	916	853	757	88.75	82.64	85.59
Avg1.				89.52	86.43	87.95
Agv2.	5902	5739	5150	89.74	87.26	88.48

Table 11: The performance of English named entity recognition with rare–but–important associations as constraints for inference

NEType	#Hm.	#Ml.	#Mt.	Pr.	Rc.	F1.
ORG	1325	1255	1112	88.61	83.92	86.20
PER	1829	1807	1630	90.20	89.12	89.66
LOC	1832	1829	1655	90.49	90.34	90.41
MISC	916	855	758	88.65	82.75	85.60
Avg1.				89.49	86.53	87.99
Agv2.	5902	5746	5155	89.71	87.34	88.51

associations were encoded into the model in the form of normal features with weighted values. The highest average F1 is 88.48%. Table 11 demonstrates the performance in the case that all 100%–confidence rare–but–important associations were integrated into the inference process in terms of Viterbi constraints. The highest F1 obtained in this case is 88.51%.

4.4 Discussion

We can see that the integration of rare–but–important associations into CRF models can improve the performance of both phrase chunking and named entity recognition tasks. The F1–measure (Avg2.) of phrase chunking increases from 93.12% to 93.65%, 93.79%, and 93.82% corresponding to three methods of encoding rare–but–important associations. Similarly, the F1–measure of NER increases from 86.40% to 88.18%, 88.48%, and 88.51%. Also, the precision–recall based F1–measure (Avg1.) also increases from 72.34% to 78.14%, 81.84%, and 81.15% for phrase chunking and from 85.60% to 87.66%, 87.95%, and 87.99% for named entity recognition. This demonstrates that our approach can improve not only the total performance but also the balance among classes/labels.

We can also draw some conclusions from the experimental results: (1) rare–but–important associations as normal CRF features (the first method) can significantly enhance the total performance; however, treating rare–but–important associations as normal features can not fully utilizes the advantage of them; (2) rare–but–important associations as constraints for inference (the third method) are sometimes too aggressive because they are globally true on one training dataset but may not true on another; and (3) treating rare–but–important associations as normal features with emphasized values should be the favorable choice because they are neither too loosely nor too tightly integrated with the models. The experimental results show that this method achieves both high total performance and the balance among classes/labels.

The experimental results reported in this paper are not the best performances on phrase chunking and named entity recognition due to some reasons: (1) our feature templates are relatively simple to keep the set of features compact; this is convenient for mining associations, training again and again during conducting the experiments; (2) unlike the CRF model in [25], all our CRF models obey the first–order Markov property to reduce the number features and the training time.

5. RELATED WORK

Discriminative (sequential) learning models have been applied successfully in different natural language processing and information extraction tasks, such as POS tagging [24], text chunking [20, 25], information extraction [8, 22], computer vision and image analysis [10, 12, 26], and biological modeling [27]. 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) to 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 has to unintentionally include 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.

McCallum [19] proposed an automated feature induction for CRFs that can reduce the number of used features dramatically. This likelihood–driven approach repeatedly add features with high likelihood gains into the model. The the set of induced features contains both atomic observational tests and conjunctions of them. The main difference between this work and ours is that this work focuses on features with high likelihood–gains to reduce the number of used features as many as possible while the main target of our method is to discover rare–but–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 context predicates because of the exhaustive working manner of association rule mining techniques.

An error–driven method that combines boosting technique into the training process of CRFs [2] to minimize an upper bound on the ranking loss adapted to label sequences. This method also focuses on hard observation sequences, however without integrating new useful conjunctions of basic features. Another boosting–like training for CRFs is based on gradient tree [6] to learning many conjunctions of features. One problem is that this method requires adding many trees for the training process.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a data–driven approach that can discover and highlight rare–but–important associations or co–occurrences of singleton context predicates from the sequential training data to deal with hard examples. Dis-

covered 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 rare–but–important associations can improve the model performance by fighting against the dominance of singleton but common statistics in the training data.

Though rare–but–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 rare–but–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. The future work will focus on this potential relation to estimate the extent to which useful patterns (e.g., rare–but–important associations) discovered from the training data can improve the performance of discriminative (sequential) learning models.

7. 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 FP–growth algorithm. We would like to say thank 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.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. VLDB, pages 487–499, 1994.

[2] Y. Altun, T. Hofmann, and M. Johnson. Discriminative learning for label sequences via boosting. In Proc. NIPS, 2002.

[3] A. Berger, A.D. Pietra, and J.D. Pietra. A maximum entropy approach to natural langauge processing. *Computational Linguistics*, 22(1):39–71, 1996.

[4] S.F. Chen and R. Rosenfeld. A gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, CMU, 1999.

[5] M. Collins. Discriminative training methods for hidden markov models: theory and experiment with perceptron algorithms. In Proc. EMNLP, 2002.

[6] T.G. Dietterich. Training conditional random fields via gradient tree boosting. In Proc. ICML, 2004.

[7] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[8] T. Kristjansson, A. Culotta, P. Viola, and A. McCallum. Interactive information extraction with constrained conditional random fields. In Proc. AAAI, pages 412–418, 2004.

[9] T. Kudo and Y. Matsumoto. Chunking with support vector machines. In Proc. ACL/NAACL, 2001.

[10] S. Kumar and M. Hebert. Discriminative random fields: a discriminative framework for contextual interaction in classification. In Proc. IEEE CVPR, pages 1150–1157, 2003.

[11] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Proc. ACM SIGMOD, pages 1–12, 2000.

[12] X. He, R.S. Zemel, and M.A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In Proc. IEEE CVPR, pages 695–702, 2004.

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In Proc. ICML, pages 282–289, 2001.

[14] W. Li, J. Han, and J. Pei. Accurate and efficient classifications based on multiple class-association rules. In Proc. IEEE ICDM, pages 369-376, 2001.

[15] D. Liu and J. Nocedal. On the limited memory BFGS method for large-scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[16] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In Proc. ACM SIGKDD, pages 80–86, 1998.

[17] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In Proc. CoNLL, 2002.

[18] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In Proc. ICML, pages 591–598, 2000.

[19] A. McCallum. Efficently inducing features of conditional random fields. In Proc. UAI, 2003.

[20] F. Peng, F. Feng, and A. McCallum. Chinese segmentation and new word detection using conditional random fields. In Proc. COLING, 2004.

[21] S.D. Pietra, V.D. Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[22] D. Pinto, A. McCallum, X. Wei, and W.B. Croft. Table extraction using conditional random fields. In Proc. ACM SIGIR, 2003.

[23] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In Proc. the IEEE, 77(2):257–286, 1989.

[24] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In Proc. EMNLP, 1996.

[25] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In Proc. HLT/NAACL, 2003.

[26] A. Torralba, K.P. Murphy, and W.T. Freeman. Contextual models for object detection using boosted random fields. In Proc. NIPS, 2004.

[27] G. Yeo and C.B. Burge. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. In Proc. Conf. on Computational Molecular Biology, pages 322–331, 2003.