ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

E. Fersini,, E. Messina, G. Felici

SOFT CONSTRAINED INFERENCE OF CONDITIONAL RANDOM FIELDS

R. 26, Dicembre 2012

Elisabetta Fersini – DISCo, University of Milano-Bicocca,
Viale Sarca, 336 - 20126 Milano, Italy Email: `fersini@disco.unimib.it`.

Enza Messina – DISCo, University of Milano-Bicocca,
Viale Sarca, 336 - 20126 Milano, Italy Email: `messina@disco.unimib.it`.

Giovanni Felici – CNR, Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti"
Viale Manzoni, 30 - 00185 Roma, Italy Email: `giovanni.felici@iasi.cnr.it`.

This work was partially done under IASI support

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

The discovering of semantic information embedded within natural language documents can be viewed as a decision making process aimed at assigning a sequence of labels to a set of interdependent variables. This problem can be addressed through a recent and promising model known as Conditional Random Fields (CRFs). Although the Viterbi algorithm can be adopted for solving the CRF inference problem, the introduction of some background knowledge in terms of non-local and non-sequential constraints makes this algorithm no longer applicable. This paper is focused on two main issues: (1) extracting background knowledge from training data and (2) relaxing the inference problem to preserve complex relationships during the output prediction. Experimental results on both real and benchmark data show the potentiality of learning constraints from data as well the proposed inference relaxation.

Key words: Named Entity Recognition; Conditional Random Fields; Inference Relaxation; Rule Extraction; Integer Programming

1. Introduction

Information Extraction (IE) is a process focused on automatic extraction of structured information from unstructured text sources. One open research field of IE relates to Named Entity Recognition (NER), aimed at identifying and associating atomic elements in a given text to a predefined category such as names of persons, organizations, locations, dates, quantities and so on.

Early NER systems have been defined as rule-based approaches with a set of fixed and manually coded rules provided by domain experts [33, 27, 35, 2]. Considering the costs, in terms of human effort, to reveal and formulate hand-crafted rules, several research communities ranging from Statistical Analysis to Natural Language Processing and Machine Learning have provided valuable contributions for automatically derive models able to detect and categorize pre-defined entities. The first tentatives aimed at deriving these rules, under the form of boolean conditions, are based on inductive rule learner where rules can be learnt automatically from labelled examples. The inductive rule learning approach has been instantiated according to different learning paradigm: bottom-up [8, 7, 12], top-down [39, 30, 26, 21] and interactive rule learning [22, 6, 4].

An alternative approach to inductive rule learners is represented by statistical methods, where the NER task is viewed as a decision making process aimed at assigning a sequence of labels to a set of either joint or interdependent variables, where complex relationships can hold among them. This decision making paradigm can be addressed in two different ways: (1) at segment-level, where the NER task is managed as a segmentation problem in which each segment corresponds to an entity label; (2) at token level, where an entity label is assigned to each token of a sentence. Concerning segment-level models [16, 37, 1] the output of the decision process is a sequence of segments, where each segment defines an entity. More formally, a segmentation s of an input sequence of length N is a sequence of segments $s_1...s_p$ such that the last segment ends at N, the first segment starts at 1, and segment s_{j+1} begins right after segment s_j ends. Each segment s_j consists of a start position l_j, an end position u_j, and a label y_j belonging to a set of entity labels Y. The second decision making paradigm is represented by token-level models [41, 5, 38, 32, 23, 34, 25], where the unstructured text is tackled as a sequence of tokens and the output of the decision process is a sequence of labels that in consecutive tokens are aggregated with the same entity label. Considering a sequence of token $x = x_1, .., x_N$, the goal is to classify each x_i as one of the entity labels $y_j \in Y$ for originating a tag sequence $y^* = y_1, .., y_N$.

Nowadays, the state-of-the-art model for tackling the NER task is represented by Linear Chain Conditional Random Fields [25], which is a discriminative undirected graphical model - initially defined as token level approach for then be extended as segment level - able to encode known relationships among tokens (observations) and labels (hidden states). The efficiency of Linear Chain Conditional Random Fields (CRF) is strictly related to the underlying Markov assumption: given the observation of a token, the corresponding hidden state depends only on the labels of its adjacent tokens. In order to efficiently enhance the description power of CRF, during the last ten years several approaches have been proposed to enlarge the information set exploited during training and inference. In particular, two main research directions have been investigated: (1) relaxing the Markov assumption [37, 14, 1] to model long distance relationships and (2) introducing additional domain knowledge in terms of logical constraints during the inference phase [24, 36, 10, 9, 17]. Considering that the relaxation of the Markov assumption implies an increasing computational complexity related to the training and inference phase, in this paper we focused our attention on constraining the inference phase by addressing two main issues: learning declarative rules from data and including them through soft constraints into

an integer linear programming (ILP) formulation to support the inferences task. In this way, conflicts between the labels of different target variables can be solved by correcting mistakes of local predictions through constraints able to produce best global assignment. In particular, standard CRFs are enhanced in a two stages approach, where the "extra knowledge" related to long distance relationships is represented through declarative rules learned from training data. The label assignment problem is therefore tackled through a constrained optimization problem where the logical rules that should be preserved are easily introduced as linear inequalities. This approach, as shown by the experimental results, makes it possible to significantly improve the performances of CRF in NER tasks.

The main outline of the paper is the following. In section 2 a brief review of CRF is presented along with a background overview of the training phase, joint with the most relevant inference approaches able to include domain constraints. In section 3 the proposed soft-constrained inference approach based is detailed by focusing on learning constraints from data and by presenting the mathematical programming formulation. In section 4 the experimental investigation on real and benchmark datasets is described, while in section 5 conclusions and ongoing research are summarized.

2. Conditional Random Fields

A Conditional Random Field is an indirected graphical model that define a single joint distribution $P(y|x)$ of the predicted labels (hidden states) $y = y_1, ..., y_N$ given the corresponding tokens (observations) $x = x_1, ..., x_N$. Formally, the definition of CRF [25] is given subsequently:

Definition 1 (Conditional Random Fields) *Let G = (V,E) be a graph such that* $Y = (Y_v)_{v \in V}$ *, so that Y is indexed by the vertices of G. Then (X,Y) is a Conditional Random Field, when conditioned on X, the random variables* Y_v *obey the Markov property with respect to the graph:* $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$*, where* $w \sim v$ *means that w and v are neighbors in G.*

According to the Hammersley-Clifford theorem [20, 13], given $\mathcal{C}$ as the set of all cliques in the graph, a CRF model defines a conditional probability distribution over G:

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{C \in \mathcal{C}} \Phi_C(\vec{x}_C, \vec{y}_C) \tag{1}$$

where Φ_C represents the set of maximal cliques $C \in \mathcal{C}$, the vertices $\vec{x_C}$ and $\vec{y_C}$ of the clique C correspond to hidden states y and observations x, and $Z(\vec{x}) \in [0, 1]$ is the partition function for global normalization. Formally $Z(\vec{x})$ is defined as:

$$Z(\vec{x}) = \sum_{\vec{y_C}} \prod_{C \in \mathcal{C}} \Phi_C(\vec{x}_C, \vec{y_C}) \tag{2}$$

Considering that each clique $\Phi_C(\cdot) \in \mathcal{C}$ can encode one or more potential functions (in this case we consider the log-linear ones), the probability of a sequence of label y given the sequence of observations x can be written as:

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \exp\Big(\sum_{t=1}^{N} \sum_{k=1}^{K} \omega_k f_k(y_t, y_{t-1} x, t) \Big) \tag{3}$$

where $f_k(y_t, y_{t-1} x, t)$ is an arbitrary real-valued feature function over its arguments and ω_k is a learned weight that tunes the importance of each feature function. In particular when for a

token x_t a given feature function f_k is active, the corresponding weight ω_k indicates how to take into account f_k: (1) if $\omega_k > 0$ it increases the probability of the tag sequence y; (2) if $\omega_k < 0$ it decreases the probability of the tag sequence y; (3) if $\omega_k = 0$ has no effect whatsoever.

The partition function $Z(x)$ is an instance-specific normalization function formally defined as:

$$Z(\vec{x}) = \sum_{\vec{y}} \Big\{ \exp \sum_{k=1}^{K} \omega_k f_k(y_t, y_{t-1}, x, t) \Big\} \tag{4}$$

The conditional probability distribution can be estimated by exploiting two different kinds of feature functions such that $p(y|x)$ can be rewritten as follows:

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \exp \Big(\sum_{t=1}^{N} \sum_{i}^{|I|} \lambda_i s_i(y_t, x, t) + \sum_{j}^{|J|} \mu_j t_j(y_{t-1}, y_t, x, t) \Big) \tag{5}$$

where I and J represent the given and fixed set of *state feature function* $s_i(y_t, x, t)$ and *transition feature function* $t_j(y_{t-1}, y_t, x, t)$, while λ_i and μ_j are the corresponding weights to be estimated from training data. State feature function and transition feature function model respectively the sequence of observations x with respect to the current state y_t and the transition from the previous state y_{t-1} to the current state y_t. The parameters λ_i and μ_j are used to weight the corresponding state and transition feature function.

The choice of the feature functions strongly depends from the application context; for example, in the NER cases considered in this paper, a typical state feature function is XXXX, while a transition feature function could be YYYYY.

2.1. Training: estimating the parameters of CRF

The learning problem in CRF relates to the identification of the best feature functions $s_i(y_t, x, t)$ and $t_j(y_{t-1}, y_t, x, t)$ by unrevealing the corresponding weights λ_i and μ_j. These parameters can be quantified either by exploiting some background domain knowledge or by learning from training data. When no background knowledge is available, several learning approaches can be adopted for estimating λ_i and μ_j. Among them a widely used approach consists of maximizing the conditional log-likelihood of the training data. Given a training set $\mathcal{T}$ composed of training samples (x, y), with y_t ranging in the set $\mathcal{Y}$ of entity labels, the conditional (penalized) log-likelihood is defined as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{T}) &= \sum_{(\vec{x},\vec{y}) \in \mathcal{T}} \log p(\vec{y}|\vec{x}, \lambda, \mu) - \sum_{i=1}^{|I|} \frac{\lambda_i^2}{2\sigma_\lambda^2} - \sum_{j=1}^{|J|} \frac{\mu_j^2}{2\sigma_\mu^2} \\ &= \sum_{(\vec{x},\vec{y}) \in \mathcal{T}} \sum_{t=1}^{N} \sum_{i}^{|I|} \lambda_i s_i(y_t, x, t) + \sum_{j}^{|J|} \mu_j t_j(y_{t-1}, y_t, x, t) - \log Z(x) \\ &- \sum_{i=1}^{|I|} \frac{\lambda_i^2}{2\sigma_\lambda^2} - \sum_{j=1}^{|J|} \frac{\mu_j^2}{2\sigma_\mu^2} \end{aligned} \tag{6}$$

where the terms $\sum_{i=1}^{|I|} \frac{\lambda_i^2}{2\sigma_\lambda^2}$ and $\sum_{j=1}^{|J|} \frac{\mu_j^2}{2\sigma_\mu^2}$ map a Gaussian prior on λ and μ used for avoiding over-fitting.

The objective function $\mathcal{L}(\mathcal{T})$ is concave, and therefore both parameters λ and μ have a unique set of global optimal values. A standard approach to parameter learning computes the gradient of the objective function to be used in an optimization algorithm [25, 40, 29, 28]. Among them we choose the quasi-Newton approach known as Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [28] due the main advantage of providing a dramatic speedups in case of huge number of feature functions.

2.2. Inference: finding the most probable state sequence

The inference problem in CRF corresponds to find the most likely sequence of hidden state y^*, given the set of observation $x = x_1, ..., x_N$. This problem can be solved approximately or exactly by determining y^* such that:

$$y^* = \arg\max_{y} p(\vec{y}|\vec{x}) \tag{7}$$

The most common approach to tackle the inference problem is represented by the Viterbi algorithm [3, 31]. Now, consider $\delta_t(y_j|\vec{x})$ as the probability of the most likely path of generating the sequence $y^* = y_1, y_2, ..., y_j$. This path can be derived by one of the most probable paths that could have generated the subsequence $y_1, y_2, ..., y_{j-1}$. Formally, given $\delta_t(y_j|\vec{x})$ as

$$\delta_t(y_j|\vec{x}) = \max_{y_1, y_2, ..., y_{j-1}} p(y_1, y_2, ..., y_j|\vec{x}) \tag{8}$$

we can derive the induction step as:

$$\begin{aligned} \delta_{t+1}(y_j|\vec{x}) &= \max_{y' \in S} \Big[\delta_t(y')\Phi_{t+1}(\vec{x}, y, y')\Big] = \\ &= \max_{y' \in S} \Big[\delta_t(y') \exp\Big(\sum_{k=1}^{K} \omega_k f_k(y_j, y', \vec{x}, t)\Big)\Big] \end{aligned} \tag{9}$$

The recursion terminates in $y^* = \arg\max_{y_j}[\delta_T(y_j)]$, allowing the backtrack to recover y^*.

3. Introducing Background Knowledge in CRF

CRFs, in their native form, are able to capture some local properties through the definition of transition and state feature functions. However, some relationships among output variables (labels to be predicted) could exist when addressing NER problems. For instance, when annotating scientific citations the label *Author* should appear before the label *Title*.

Two different strategies are suitable to include relationships in the probabilistic model: the feature way and the constraint way. The first one is concerned with the training phase by the definition of feature functions able to capture different kinds of relationships [37]. The second one relates to the introduction of constraints during the inference phase for preserving the necessary relationships over the output prediction [24, 36, 10, 11]. While the first strategy might lead to intractable training and inference due to the necessity of learning additional parameters or defining higher order models, the second paradigm allows us to keep the model simple by enclosing expressive constraints directly during the inference phase. In this context we can distinguish between constraining the Viterbi algorithm [24, 15] and formulating the inference process as a constrained optimization problem.

The main idea underlying the Constrained Viterbi algorithm is concerned with the clamping of some hidden variables to particular values. The resulting algorithm alters the induction step

outlined in equation (9) such that y^* is constrained to pass through a sub-path $C = \langle y_t, y_{t+1}... \rangle$. The constraint C is encoded in the induction step as follows:

$$\delta_{t+1}(y_j|\vec{x}) = \begin{cases} \max\limits_{y' \in S} \Big[\delta_t(y') \exp\Big(\sum\limits_{k=1}^{K} \omega_k f_k(y_j, y', \vec{x}, t) \Big) \Big] & \text{if } y_j = y_{t+1} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

for all $y_{t+1} \in C$. For time steps not involved in C the equation (8) is used instead, restricting the algorithm to only consider paths that respect the defined constraint.

Although the Constrained Viterbi algorithm is suitable to deal with short distance relationships in an efficient way, the introduction of non-local and non-sequential constraints makes this solution no longer applicable. In order to deal with more complex relationships, the inference process can be formulated as a constrained optimization problem. In particular, Roth and Yih have shown in [36], that the most likely sequence of hidden state y^* given the observation x can be formulated as a shortest path problem. Given n tokens and m labels that each token can take, we can define a graph $\Omega = (\Phi, \Psi)$ as composed of $nm + 2$ nodes and $(n-1)m^2 + 2m$ edges [1]. The label of each token is represented by a node ϕ_{iy}, where $0 \leq i \leq n-1$ and $0 \leq y \leq m-1$, while the arc connecting two adjacent nodes $\phi_{(i-1)y}$ and $\phi_{iy'}$is denoted by a directed edge $\psi_{i,yy'}$ with the associated cost $-\log(M_i(yy'|x))$ [2]. The problem consists in minimizing the cost of visiting the nodes $\phi_{iy'}$ along the entire path:

$$\arg\min_{\vec{y}} - \sum_{i=0}^{n-1} \log(M_i(y, y')|\vec{x}) = \arg\max_{\vec{y}} \prod_{i=0}^{n-1} \log(M_i(y, y')|\vec{x}) \tag{11}$$

The goal defined in equation (11), which corresponds to find the shortest path along the graph Ω, can be formulated as an Integer Linear Programming problem that allow us to introduce additional background knowledge in terms of constraints. Let $e_{i,yy'}$ denote a decision variable restricted to be 1 if the edge $\psi_{i,yy'}$ is in the shortest path and 0 otherwise. The ILP formulation for the shortest path problem[3] can be formalized as follows:

$$\max Z(e) = \sum_{\substack{0 \leq i \leq n-1 \\ 0 \leq y, y' \leq m-1}} \log M_i(y, y') \cdot e_{i,yy'} \tag{12}$$

subject to:

$$\sum_{0 \leq y_1 \leq m-1} e_{i-1,y_1 y} - \sum_{0 \leq y_2 \leq m-1} e_{i,yy_2} = 0 \tag{13}$$

$$\sum_{0 \leq y \leq m-1} e_{-1,0y} = 1 \text{ and } \sum_{0 \leq y \leq m-1} e_{n,y0} = 1 \tag{14}$$

$$e_{-1,0y},\ e_{i,y_1 y},\ e_{n,y0} \in \{0, 1\} \tag{15}$$

$$\forall i, y \text{ s.t. } 0 \leq i \leq n-1, 0 \leq y, y_1, y_2 \leq m-1 \tag{16}$$

Some relationships that should be preserved over the output prediction could be smoothly represented as Boolean function and therefore introduced as linear inequalities constraints [43].

[1] Two special nodes denoting the *start* and *end* positions of the path are additionally introduced

[2] The weight of each edge corresponds to the exponential in equation (5)

[3] The solution of the shortest path problem corresponds to the output of the Viterbi algorithm

Nevertheless, this approach is based on two main assumptions: (1) constraints must be manually defined by a domain expert; (2) constraints are introduced as"hard". The implications arisen are concerned with the time consuming activity on defining boolean functions and the satisfiability of constraints with respect to training and testing data (otherwise possible infeasible solutions could be obtained). In order to overcome the mentioned weak points, a combination of learning constraints from data and a two-stages ILP approach is proposed.

3.1. The Proposed Approach

In this section we outline in detail the approach proposed in this paper. First we describe a general approach to extract from the avaialble data additional knowledge in the for of logic rules; then we describe hos such knowledge can be embedded in a mathematical model as sofr constraints, by relaxing the inference procedure and using a set of additional constraints whose violation is minimized in the objective function.

3.1.1. Learning constraints from data

Some knowledge embedded in the training data could be automatically extracted up front and then included in the inference phase as constraints over the possible output prediction. In a sequential labeling issue, the problem of finding out valuable knowledge can be viewed as a discovery process aimed at revealing common patterns within training data and a subsequent extraction of logical relationships from the identified patterns. The identification of some common patterns and their extraction from data in the form of logic rules is formulated, as originally proposed in [18] and [19], as a sequence of minimum cost satisfiability problems. These problems are solved efficiently with an ad hoc algorithm based on the decomposition strategies presented in [42]. The method adopts a standard learning paradigm where the different tag labels are the classes and the logic rules that are found are able to separate the samples of one class from the samples of the other classes optimizing the information used to define the logic rules with the aim of controlling the performances of the system in a desired direction. The formulas so obtained are in disjunctive normal form (DNF), i.e. they are composed by the disjunction of one or more conjunctive clauses. A specific characteristic of this method is that these clauses have decreasing discrimination power and thus they can be pruned according to their power and to the importance that is given to the rules that explain the outliers that may be present in the training data. Moreover, each conjuctive clause can be easily associated with an integer linear constraints with standard techniques.

Interesting logic relationship that are extracted from data may fall into one of the following categories:

- **Adjacency**: if label A is associated to token x_i, then label B must be associated to token x_{i+1}

$$C(\sigma_1): \sum_{0 \leq y_1 \leq m-1} e_{i-1,y_1 A} - \sum_{0 \leq y_2 \leq m-1} e_{i,By_2} - \sigma \leq 0 \tag{17}$$

- **Precedence**: if label A is associated to token x_i, then label B should be associated to token x_{i+t}

$$C(\sigma_2): \sum_{0 \leq y_1 \leq m-1} e_{i-1,y_1 A} - \sum_{\substack{0 \leq y_2 \leq m-1 \\ 0 \leq t \leq n-i}} e_{i+t,By_2} - \sigma_1 \leq 0 \tag{18}$$

- **State Change**: if x_i is a given delimiter punctuation mark (d), then label A and label B should be associated to token x_{i-1} and x_{i+1} respectively

$$C(\sigma_3) : \sum_{0 \leq y_1 \leq m-1} 2(e_{i-1,y_1 d}) - \sum_{0 \leq y_2 \leq m-1} e_{i-2,y_2 A} - \sum_{0 \leq y_3 \leq m-1} e_{i,By_3} - \sigma_2 \leq 0 \tag{19}$$

- **Begin-End Position**: label A should be associated to token x_0

$$C(\sigma_4) : \sum_{0 \leq y_1 \leq m-1} e_{1,Ay_1} - \sum_{0 \leq y_2 \leq m-1} e_{n-1,y_2 B} - \sigma_3 \leq 0 \tag{20}$$

- **Presence and Precedence**: if label A appears, then label B can not appear before label A

$$C(\sigma_5) : m(i-2)e_{i,Ay_1} - \sum_{\substack{1 \leq t \leq i-2 \\ 0 \leq y_2 \leq m-1}} (1 - e_{t,y_2 B}) - \sigma_5 \leq 0 \tag{21}$$
$$with\ 2 \leq i \leq n\ and\ 0 \leq y_1 \leq m-1$$

The linear inequality constraints enclose a variable $\sigma_h \in \{0, 1\}$, which will indicate in the following two-stages ILP approach when a given logical relationships could not be preserved over the output prediction. The constraints associated with the rules can thus be represented in a compact way with the notation:

$$L \cdot e + \sigma \geq 0$$

where L is a matrix composed of H rows containing the coefficients of the components of vector e for each of the H constraints, and σ is the binary vector of size H representing the violation of the constraints.

3.1.2. Relaxing Inference using ILP

In order to introduce the automatically learned logical rules for constraining the inference phase, a two-fold approach is proposed. The first step is aimed at determining the optimal solution of the shortest path by solving the problem formulation presented in equations (12) - (14), i.e. to determine the optimal solution $e^*_{i,yy'}$. Concerning the second step, the main idea is to identify a labeling solution consistent with the logical rules previously extracted and to ensure simultaneously a good approximation of the original shortest path solution.

The second step, targeted therefore at minimizing the number of violated constraints, is formulated as follows:

$$\min Z(\sigma) = \sum_h c_h \sigma_h \tag{22}$$

subject to:

$$\sum_{\substack{0 \leq i \leq n-1 \\ 0 \leq y,y' \leq m-1}} \log M_i(y, y'|x) \cdot e_{i,yy'} \geq \tau Z(e^*_{i,yy'}) \tag{23}$$

$$\sum_{0 \leq y_1 \leq m-1} e_{i-1,y_1 y} - \sum_{0 \leq y_2 \leq m-1} e_{i,yy_2} = 0 \tag{24}$$

$$\sum_{0\leq y\leq m-1} e_{-1,0y} = 1 \text{ and } \sum_{0\leq y\leq m-1} e_{n,y0} = 1 \tag{25}$$

$$L \cdot e + \sigma \geq 0 \tag{26}$$

$$e_{-1,0y},\ e_{i,y_1y},\ e_{n,y0},\ \sigma_k \in \{0,1\} \tag{27}$$

$$\forall i, y \text{ s.t. } 0 \leq i \leq n-1, 0 \leq y, y_1, y_2 \leq m-1 \tag{28}$$

$$\sigma_h \in \{0,1\}, \qquad h = 1, ..., H \tag{29}$$

The objective function (22) is penalized whenever a logical relationship is violated, i.e. when $\sigma_h = 1$. Constraints (24) - (29) play the same role as in the shortest path problem, while constraint (23) states that the variables $e_{iyy'}$ should assume values to guarantee a solution close to the shortest path one. This lower bound ensures the current solution of $Z(\sigma)$, which originates a novel configuration of $e_{i,yy'}$, to be coherent to the solution determined at the first step according to the threshold τ . The parameter c_h represents the cost of violating a given constraint associated to the variable σ_h. In particular, such cost is proportional to the occurrence of a clause in the training data and represents the (log) probability that the corresponding constraint is violated. Given a clause L_h representing the logical relationship among labels (for instance label A should appear before label B), the cost c_h of violating the corresponding constraint indexed by σ_h is computed as follows:

$$c_h = \log P\left(= \frac{|D(L_h)|}{|D(L_h)| + |\overline{D(L_h)}|} \right) \tag{30}$$

where $D(L_h)$ denotes the set of true clauses and $\overline{D(L_h)}$ represents the set of clauses that are not satisfied in training data.

4. Experimental Results

The proposed method has been tested with success on different benchmark data. Particular attention has been given the methods used ot evaluate and compare the performance of the method, as detailed in the following section.

4.1. Performance criteria

The performance in terms of effectiveness has been measured by using four well known evaluation metrics, i.e. F-Measure, Precision, Recall and Accuracy. The F-Measure metric represents a combination of Precision and Recall typical of Information Retrieval. Given a set of labels Y we compute the Precision and Recall for each label $y_j \in Y$ as:

$$Precision(y) = \frac{\text{\# of tokens successfully predicted as } y}{\text{\# of tokens predicted as} y} \tag{31}$$

$$Recall(y) = \frac{\text{\# of tokens successfully predicted as } y}{\text{\# of tokens effectively labelled as y}} \tag{32}$$

The F-Measure for each class $y \in Y$ is computed as the harmonic mean of Precision and Recall:

$$F(y) = \frac{2 \cdot Recall(y) \cdot Precision(y)}{Recall(y) + Precision(y)} \tag{33}$$

Concerning the Accuracy measure, it can be summarized as follows:

$$Accuracy = \sum_{y_j \in Y} \frac{\#\text{ of tokens correctly labelled as } y}{\text{total number of tokens}} \tag{34}$$

Considering that the datasets are composed of unbalanced samples of different labels, for Precision, Recall and F-Measure, both micro and macro average have been computed. The main difference between micro and macro measures relates to the computation of global performance: macro-averaging gives equal weight to each label category (independently from the category size), while micro-averaging consider the contribution of each label class according to its dimension. While computing macro-average performance indicators, a global performance is determined by the average of performance along the considered labels, the micro-average computation takes into account the contribution of each label category according to its dimension.

4.2. Datasets

In order to compare the proposed model with traditional CRFs, we carried out experiments on several datasets: *Cora*, *Cora-Real* and *US50*. The *Cora* citation benchmark is composed of 500 citations of research papers annotated with 13 different labels: *Title, Author, Book Title, Date, Journal, Volume, Technology, Institution, Pages, Editor, Location, Notes*. The benchmark has been split for training and testing the models: 350 instances have been used as training set, while the remaining 150 instances as testing.

	Cora		**Cora-Real**
	Train	**Test**	**Test**
Author	1948	801	12071
Title	2585	1103	12326
Publisher	226	61	1567
Booktitle	1414	477	6670
Data	452	187	1922
Journal	402	219	1478
Volume	216	104	1181
Technology	173	57	817
Insitution	236	71	1372
Pages	500	208	3048
Editor	151	74	2767
Location	208	95	1956
Note	116	17	631
Tot.	**8627**	**3474**	**47806**

Table 1: Cora and Cora-Real label distribution

An additional (real) dataset concerning the citation of scientific papers has been collected and named *Cora-Real*, which comprises 5000 instances used as testing (the model has been induced by using the 150 training instances of the Cora benchmark). This dataset has been obtained by downloading the citations from popular sources collecting citations according to Springer-Verlag, ACM, IEEE and CiteSeer formats. The last dataset, named *US50*, is about US Postal Addresses. It is composed of 740 instances, where 50 are used as training and 690 as testing. Each postal address has been manually annotated according the following labels: *Street, City, Civic Number, State and Zip Code*. A summary of the considered datasets is reported in Table 1 and Table 2.

	US50	
	Train	**Test**
Street	228	1782
City	61	866
Civic Number	45	597
State	50	689
Zip Code	50	689
Tot.	**434**	**4623**

Table 2: US50 label distribution

4.3. Results

We evaluated the proposed inference approach by measuring its ability to predict the correct token labels by training and testing CRFs. In particular, the proposed inference relaxation has been compared with of the state-of-the-art Viterbi algorithm. Tables 3-5 report the performance on the considered datasets, both in terms of macro and micro average, of Precision (P), Recall (R), F-Measure (F) and Accuracy (A).

	Macro-Average			**Micro-Average**			
	P	**R**	**F**	**P**	**R**	**F**	**A**
Baseline (Viterbi)	**85.49**	72.02	76.36	87.27	87.10	86.65	87.13
Adiacency	78.56	78.37	78.01	88.07	87.84	87.77	87.85
Precedence	77.13	78.48	76.55	86.95	85.92	85.79	85.00
State Change	81.34	78.95	79.93	90.53	90.32	90.27	90,32
Begin-End	84.49	72.00	76.12	87.47	87.00	86.49	87.91
Presence and Precedence	79.59	78.27	78.60	87.81	87.70	87.59	87.70
All Constraints	84.40	**79.99**	**82.19**	**92.31**	**91.34**	**91.42**	**92.89**

Table 3: Performance comparison on the Cora benchmark

The performance are detailed by showing at first the results of the baseline inference approach (Viterbi) and then illustrating the contribution of the considered soft constraints on the relaxed ILP formulation. The learned constraints, in most cases, allows the relaxed inference to achieve good performance of the assessed criteria, by ensuring significant improvements with respect to the Viterbi solution. When introducing all the modeled constraints in the soft ILP formulation, the proposed approach achieves the highest results.

	Macro-Average			**Micro-Average**			
	P	**R**	**F**	**P**	**R**	**F**	**A**
Baseline (Viterbi)	**46.49**	21.28	18.53	63.19	14.67	20.34	37.80
Adiacency	44.31	45.12	42.90	64.85	50.10	53.22	56.43
Precedence	42.56	46.78	42.85	62.55	52.86	52.05	54.52
State Change	45.50	46.78	42.91	65.77	53.00	54.68	60.85
Begin-End	45.38	43.36	42.32	64.22	53.29	51.57	56.94
Presence and Precedence	44.62	44.23	42.03	64.88	53.91	54.16	57.35
All Constraints	45.63	**46.94**	**43.06**	**67.59**	**55.20**	**58.10**	**66.96**

Table 4: Performance comparison on the Cora-Real dataset

It's easy to note that for the Cora-Real dataset, the obtained results are remarkable: while the Viterbi algorithm shows poor performance, the proposed approach confirms its robustness by ensuring satisfactory results on this more complex test case.

	Macro-Average			**Micro-Average**			
	P	**R**	**F**	**P**	**R**	**F**	**A**
Baseline (Viterbi)	89.48	90.06	86.57	89.79	80.76	80.77	80.80
Adiacency	89.61	90.31	86.91	89.92	81.24	81.29	81.24
Precedence	90.74	90.42	87.52	90.40	81.57	81.74	82.12
State Change	90.02	91.05	87.07	90.31	82.66	82.80	82.91
Begin-End	90.00	90.89	87.95	90.95	82.50	82.95	83.25
Presence and Precedence	90.63	91.13	87.93	90.99	82.42	82.19	82.23
All Constraints	**91.23**	**91.90**	**88.75**	**91.90**	**83.39**	**83.90**	**84.55**

Table 5: Performance comparison on the US50 dataset

It's also interesting to highlight that the gains of the relaxed ILP formulation are larger when the dataset shows complex structural relationships among token labels, as for instance in the Cora and Cora-Real datasets. While the Viterbi algorithm could be "biased" by transition feature function encoded in $\log M_i(y, y'|x)$, which captures only local relationships, the proposed approach is able to find a more appropriate labeling solution thanks to the global relationships encoded as constraints.

14.

5. Conclusions and Ongoing Research

In this paper we have presented a two-fold approach for relaxing inference related to NER problems by using Conditional Random Fields. The proposed solution is based on the automatic extraction of background knowledge from training data and in the incorporation of such knowledge in the inference problem. The whole process is based on the use of integer linear programming to represent the Viterbi algorithm for CRF inference and on its relaxation to incorporate the additional knowledge as soft constraints in the ILP model. The adoption of a relaxation of the inference problem and the use of properly weighted violation variables for the additional constraints is designed to preserve complex relationships during the output prediction.

Experimental tests show that our system significantly outperforms the current state of the art approach, obtaining remarkable performance gains across several domains and types of data (benchmark and real datasets).

Acknowledgment

This work has been partially funded by the grant "Dote Ricercatori" - FSE, Regione Lombardia.

References

[1] G. Andrew, "A hybrid markov/semi-markov conditional random field for sequence segmentation," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 465–472, 2006.

[2] D. E. Appelt, J. R. Hobbs, J. Bear, D. J. Israel, and M. Tyson, "Fastus: A finite-state processor for information extraction from real-world text.," in *IJCAI'93*, pp. 1172–1178, 1993.

[3] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. pp. 1554–1563, 1966.

[4] M. Beckerle, L. A. Martucci, and S. Ries, "Interactive access rule learning: Generating adapted access rule sets," in *In Proc. of the Second International Conference on Adaptive and Self-adaptive Systems and Applications*, pp. 104–110, 2010.

[5] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," in *Proceedings of the fifth conference on Applied natural language processing*, (Stroudsburg, PA, USA), pp. 194–201, Association for Computational Linguistics, 1997.

[6] P. Bohannon, S. Merugu, C. Yu, V. Agarwal, P. DeRose, A. Iyer, A. Jain, V. Kakade, M. Muralidharan, R. Ramakrishnan, and W. Shen, "Purple sox extraction management system," *SIGMOD Rec.*, vol. 37, pp. 21–27, 2009.

[7] M. E. Califf and R. J. Mooney, "Relational learning of pattern-match rules for information extraction," in *Proc. of the 16th national conference on Artificial intelligence and the 11th Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pp. 328–334, American Association for Artificial Intelligence, 1999.

[8] M. E. Califf and R. J. Mooney, "Bottom-up relational learning of pattern matching rules for information extraction," *J. Mach. Learn. Res.*, vol. 4, pp. 177–210, 2003.

[9] M. Chang, L. Ratinov, and D. Roth, "Constraints as prior knowledge," in *ICML Workshop on Prior Knowledge for Text and Language Processing*, pp. 32–39, 2008.

[10] M.-W. Chang, L. Ratinov, and D. Roth, "Guiding semi-supervision with constraint-driven learning," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, (Prague, Czech Republic), pp. 280–287, Association for Computational Linguistics, 2007.

[11] M.-W. Chang, L. Ratinov, and D. Roth, "Structured learning with constrained conditional models," *Machine Learning*, vol. 88, no. 3, pp. 399–431, 2012.

[12] F. Ciravegna, "Adaptive information extraction from text by rule induction and generalisation," in *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, pp. 1251–1256, Morgan Kaufmann Publishers Inc., 2001.

[13] P. Clifford, *Markov random fields in statistics; Disorder in Physical Systems: A Volume in Honour of John M. Hammersley.* Oxford University Press, 1990.

[14] W. W. Cohen and S. Sarawagi, "Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 89–98, ACM, 2004.

[15] A. Culotta, T. Kristjansson, A. McCallum, and P. Viola, "Corrective feedback and persistent learning for information extraction," *Artificial Intelligence*, vol. 170, no. 14â"15, pp. 1101 – 1122, 2006.

[16] H. Daumé, III and D. Marcu, "Learning as search optimization: approximate large margin methods for structured prediction," in *Proc. of the 22nd international conference on Machine learning*, pp. 169–176, ACM, 2005.

[17] Y. Dong, Q. Li, Y. Zheng, X. Xu, and Y. Zhang, "Semantic annotation of web objects using constrained conditional random fields," in *Proceedings of the 11th international conference on Web-age information management*, (Berlin, Heidelberg), pp. 28–39, Springer-Verlag, 2010.

[18] G. Felici and K. Truemper, "A minsat approach for learning in logic domains," *INFORMS Journal on computing*, vol. 14, pp. 20–36, 2002.

[19] G. Felici and K. Truemper, "The lsquare system for mining logic data," in *Encyclopedia of Data Warehousing and Mining* (J. W. ed., ed.), pp. 693–697, Idea Group Publishing, 2005.

[20] J. Hammersley and P. Clifford, *Markov fields on finite graphs and lattices.* Unpublished, 1971.

[21] T. B. Ho and D. D. Nguyen, "Chance discovery and learning minority classes," *New Generation Computing*, vol. 21, no. 2, pp. 149–161, 2003.

[22] S. Khaitan, G. Ramakrishnan, S. Joshi, and A. Chalamalla, "Rad: A scalable framework for annotator development," in *Data Engineering, International Conference on*, vol. 0, pp. 1624–1627, IEEE Computer Society, 2008.

[23] D. Klein and C. D. Manning, "Conditional structure versus conditional estimation in nlp models," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, (Stroudsburg, PA, USA), pp. 9–16, Association for Computational Linguistics, 2002.

[24] T. Kristjansson, A. Culotta, P. Viola, and A. McCallum, "Interactive information extraction with constrained conditional random fields," in *Proceedings of the 19th national conference on Artifical intelligence*, pp. 412–418, AAAI Press, 2004.

[25] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, (San Francisco, CA, USA), pp. 282–289, Morgan Kaufmann Publishers Inc., 2001.

[26] N. Landwehr, K. Kersting, and L. D. Raedt, "Integrating naive bayes and foil," *J. Mach. Learn. Res.*, vol. 8, pp. 481–507, 2007.

[27] W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman, "Umass/hughes: description of the circus system used for muc-5," in *Proceedings of the 5th conference on Message understanding*, pp. 277–291, Association for Computational Linguistics, 1993.

[28] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proc. of the 6th conference on Natural language learning - Volume 20*, pp. 1–7, Association for Computational Linguistics, 2002.

[29] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2000.

[30] J. R. Quinlan, "Learning logical definitions from relations," *Machine Learning*, vol. 5, no. 3, pp. 239–266, 1990.

[31] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, 1989.

[32] A. Ratnaparkhi, "Learning to parse natural language with maximum entropy models," *Machine Learning*, vol. 34, no. 1-3, pp. 151–175, 1999.

[33] L. Rau, "Extracting company names from text," in *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, vol. i, pp. 29 –32, 1991.

[34] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, pp. 107–136, 2006.

[35] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *Proceedings of the eleventh national conference on Artificial intelligence*, pp. 811–816, AAAI Press, 1993.

[36] D. Roth and W.-t. Yih, "Integer linear programming inference for conditional random fields," in *Proceedings of the 22nd international conference on Machine learning*, (New York, NY, USA), pp. 736–743, ACM, 2005.

[37] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *In Advances in Neural Information Processing Systems 17*, pp. 1185–1192, 2004.

[38] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction," in *AAAI'99 Workshop on Machine Learning for Information Extraction*, pp. 37–42, 1999.

[39] S. Soderland, "Learning information extraction rules for semi-structured and free text," *Machine Learning*, vol. 34, pp. 233–272, 1999.

[40] C. A. Sutton, *Efficient training methods for conditional random fields*. PhD thesis, University of Massachusetts, 2008.

[41] K. Takeuchi and N. Collier, "Use of support vector machines in extended named entity recognition," in *Proc. of the 6th conference on Natural language learning - Volume 20*, pp. 1–7, 2002.

[42] K. Truemper, *Design of Logic-based Intelligent Systems*. Wiley-Interscience, 2004.

[43] Y. A. Zuev, "Representations of boolean functions by systems of linear inequalities," *Cybernetics and Systems Analysis*, vol. 21, pp. 567–571, 1985.