Stacked Conditional Random Fields Exploiting Structural Consistencies

Peter Kluegl[1 2], Martin Toepfer[1], Florian Lemmerich[1], Andreas Hotho[1] and Frank Puppe[1]
[1]*Department of Computer Science VI, University of Würzburg, Am Hubland, Würzburg, Germany*
[2]*Comprehensive Heart Failure Center, University of Würzburg, Straubmühlweg 2a, Würzburg, Germany*
{*pkluegl, toepfer, lemmerich, hotho, puppe*}*@informatik.uni-wuerzburg.de*

Keywords: crf, stacked graphical models, structural consistencies, collective information extraction, rule learning

Abstract: Conditional Random Fields (CRF) are popular methods for labeling unstructured or textual data. Like many machine learning approaches these undirected graphical models assume the instances to be independently distributed. However, in real world applications data is grouped in a natural way, e.g., by its creation context. The instances in each group often share additional structural consistencies. This paper proposes a domain-independent method for exploiting these consistencies by combining two CRFs in a stacked learning framework. The approach incorporates three successive steps of inference: First, an initial CRF processes single instances as usual. Next, we apply rule learning collectively on all labeled outputs of one context to acquire descriptions of its specific properties. Finally, we utilize these descriptions as dynamic and high quality features in an additional (stacked) CRF. The presented approach is evaluated with a real-world dataset for the segmentation of references and achieves a significant reduction of the labeling error.

1 Introduction

The vast availability of unstructured and textual data increased the interest in automatic sequence labeling and content extraction methods over the last years. One of the most popular techniques are Conditional Random Fields (CRF) and their chain structured variant linear chain CRF. CRFs model conditional probabilities with undirected graphs and are trained in a supervised fashion to discriminate label sequences. Although CRFs and related methods achieve remarkable results, there remain many possibilities to increase their accuracy.

One aspect of improvement has been the relaxation of the assumption that the instances are independent and identically distributed. Relational and non-local dependencies of instances or interesting entities have been in the focus of collective information extraction. Due to the fact that these dependencies need to be represented in the model structure, approximate inference techniques like Gibbs Sampling (Finkel et al., 2005) or Belief Propagation (Sutton and McCallum, 2004) are applied. They achieved significant improvements, but at the cost of a computationally expensive inference. It has been shown by several approaches that combined models based only on local features and exact inference can match the results of complex models while still being efficient. Kou and Cohen (Kou and Cohen, 2007) used stacked graphical learning to aggregate the output of a base learner and to add additional features based on related instances to a stacked model. Another example is Krishnan and Manning (Krishnan and Manning, 2006) who exploit label consistencies with a two-stage CRF. However, all these approaches take only similar tokens or related instances into account while the consistencies of the structure are ignored.

Semi-structured text like any other data is always created in a certain context. This may often lead to structural consistencies between the instances in this creation context. While these instances are locally homogeneously distributed in one context, the dataset is globally still heterogeneous and the structure of information is possibly contradictory. The bibliographic section of a scientific publication, for example, applies a single style guide and its instances, that are the references, share a very similar structure, while their structure might differ for different style guides. Previously published approaches, c.f., (Gulhane et al., 2010) represent structural properties directly in a higher-order model and thus suffer from an computationally expensive inference and furthermore apply a domain-dependent model.

In this paper, we propose a novel and domain-independent method for exploiting structural consistencies in textual data by combining two linear chain

CRFs in a stacked learning framework. After the instances are initially labeled, a rule learning method is applied on label transitions within one creation context in order to identify their shared properties. The stacked CRF is then supplemented with high quality features that help to resolve possible ambiguities in the data. We evaluate our approach with a real world dataset for the segmentation of references, a domain that is widely used to assess the performance of information extraction techniques. The results show a significant reduction of the labeling error and confirm the benefit of additional features induced online during processing the data.

The rest of the paper is structured as follows: First, Section 2 gives a short introduction in the background of the applied techniques. Next, Section 3 describes how structural consistencies can be exploited with stacked CRFs. The experimental results are presented and discussed in Section 4. Section 5 gives a short overview of the related work and Section 6 concludes with a summary of the presented work.

2 Background

The presented method combines ideas of linear chain Conditional Random Fields (CRF), stacked graphical models and rule learning approaches. Thus, these techniques are shortly introduced in this section.

2.1 Linear Chain Conditional Random Fields

Linear chain CRF (Lafferty et al., 2001) are a chain structured case of discriminative probabilistic graphical models. The chain structure fits well with sequence labeling tasks, naturally reflecting the inherent structure of the data while providing efficient inference. By modeling conditional distributions, CRFs are capable of handling large numbers of possibly interdependent features.

Let $\mathbf{x}$ be a sequence of tokens $\mathbf{x} = (x_1, \ldots, x_T)$ referring to observations, e.g. the input text split into lexical units, and $\mathbf{y} = (y_1, \ldots, y_T)$ a sequence of labels assigned to the tokens. Taking $\mathbf{x}$ and $\mathbf{y}$ as arguments, let $f_1, \ldots, f_K$ be real valued functions, called *feature functions*. To keep the model small, we restrict the linear chain CRF to be of Markov order one, i.e. the feature functions have the form $f_i(\mathbf{x}, \mathbf{y}) = \sum_t f_i(\mathbf{x}, y_{t-1}, y_t, t)$. A linear chain CRF of Markov order one has K model parameters $\lambda_1, \ldots, \lambda_K \in \mathbb{R}$, one for each feature function, by which it assigns the conditional probability

$$P_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp\left(\sum_{t=1}^{T} \sum_{i=1}^{K} \lambda_i \cdot f_i(y_{t-1}, y_t, \mathbf{x}, t) \right),$$

to $\mathbf{y}$ with a given observation $\mathbf{x}$. The feature functions typically indicate certain properties of the input, e.g. capitalization or the presence of numbers, while the model parameters weight the impact of the feature functions on the inference. The partition function $Z_\mathbf{x}$ normalizes $P_\lambda(\mathbf{y}|\mathbf{x})$ by summing over all possible label sequences for $\mathbf{x}$. The properties of a token x_t indicated by feature functions usually consider a small fixed sized window around x_t for a given state transition. In the following, we will use the terms feature and feature function interchangeably.

2.2 Stacked Graphical Models

Stacked Graphical Models is a general meta learning algorithm, c.f., (Kou and Cohen, 2007; Krishnan and Manning, 2006). First, the data is processed by a base learner with conventional features representing local characteristics. Subsequently, every single data instance is expanded by information about the inferred labels of related instances. In a second stage, a stacked learner is provided this extra information. The process of aggregating and projecting the predicted information on instance features to support another stacked learner can be repeated several times.

Stacked Graphical Models have two central advantages: The approach enables to model long range dependencies among related instances and the inference for each learner remains effective. If the base learner and the stacked learner are both linear chain CRFs, then the inference time is only twice the time of a single CRF plus the effort to determine the aggregate information.

Similar to Wolpert's Stacked Generalization (Wolpert, 1992), Stacked Graphical Models use a cross-fold technique in order to avoid overfitting. That is, each instance $\mathbf{x}$ of the training data for the stacked learner F^k at level k, is classified by a model for F^{k-1} that was trained on data that did not contain $\mathbf{x}$. As a result, the stacked learners get to know realistic errors of their input components that would also occur during runtime. However, Stacked Generalization and Stacked Graphical Models are essentially different approaches. In short, Stacked Generalization learns a stacked learner to combine the output of several different base learners on a per instance basis. In contrast, Stacked Graphical Models utilize a stacked learner to aggregate and combine the output of one base learner on several instances, thus supporting collective inference.

2.3 Rule Learning

In this paper, we propose to utilize rule-based descriptions as an intermediate step of our general approach, cf. Section 3. For this task we will transfer the data into a tabular form of attribute value pairs and learn rules on this data representation. While over the last decades a large amount of rule learning approaches have been proposed, we will concentrate on two main approaches in this paper:

Ripper (Cohen, 1995) is probably the currently most popular learning algorithm for learning a *set* of rules. Ripper learns rules one at a time by growing and pruning each rule and then adds them to a result set until a stop criterion is met. After adding a rule to the result, examples covered by this rule are then removed from the training data. Ripper is known to be on par regarding classification performance with other learning algorithms for rule sets, e.g., C 4.5, but is computationally more efficient.

As an alternative, we utilize *Subgroup Discovery* (Klösgen, 1996) (also called Supervised Descriptive Rule Discovery or Pattern Mining) to describe structural consistencies. In this approach, an exhaustive search for the best k conjunctive patterns in the dataset with respect to a pre-specified target concept and a quality function, e.g., the F_1 measure, is performed. Additionally different constraints on the resulting patterns can be applied, e.g., on the maximum number of describing attribute value pairs or the minimum support for a rule. While the resulting rules are not intended to be used directly as a classifier, a related approach using patterns based on improvement of the target share and additional constraints has recently been successfully applied as an intermediate feature construction step for classification tasks (Batal and Hauskrecht, 2010).

3 Method

For introducing the proposed method, we first motivate the problem. Then, the stacked inference, the induction of the structural properties and the parameter estimation are presented.

3.1 Problem Description

Recap the inference formula of CRFs (c.f. Section 2.1). From the model designers' perspective, the classification process is mainly influenced by the choice of the feature functions f_i. The feature functions need to provide valuable information to discriminate labels for all possible kinds of instances. This works well when the feature functions encode properties that have the same meaning for inference across arbitrary instances. For example in the domain of reference segmentation, some special words have a strong indicative meaning for a certain task: The word identity feature "WORD=proceedings" always suggests labeling the token as "Booktitle". Thus, the learning algorithm will fix the corresponding weights to high values, leading the inference procedure into the right direction. Some features, however, violate the assumption of a consistent meaning. Their validity depends on a special context or is restricted through long range dependencies. In our example of reference extraction, the feature that indicates colons might suggest an author label if the document finishes author fields with colons. However, other style guides define a different structure of the author labels. Consequently, the learning algorithm assigns the weights to average the overall meaning. On the one hand, this yields good generalization given enough training data. On the other hand, averaging the weights of such features restricts them to stay behind their discriminative potentials. If we knew that a certain feature has a special meaning inside the given context, we could do better by increasing (or decreasing) the weights, dynamically adapting to the given context. This procedure on some particular features cannot be performed independently of the remaining weights. Hence, we apply a different approach in this paper. Instead of changing the model parameters, we learn the weights of additional feature functions describing these structural and context dependent consistencies.

Structural consistencies can be found in many domains, especially when several instances are created within one creation context. Besides the already mentioned segmentation of references, where the knowledge about the applied style guide can greatly increase the accuracy, there are many other examples. The segmentation of physicians' letters severely depends on the identification of headlines. Each author applies different and often contradictory layouts for the headlines using word processing software. However, the headlines within a letter rely always on an identical structure. By providing information about this consistency additionally to common keywords, the headlines can be accurately identified. As another example, the labeling of interesting entities in curriculum vitae, e.g., the employer, relies on large dictionaries. Their incompleteness can be neglected when exploiting the uniform composition of one curriculum vitae. Besides these examples there are many other domains like content extraction from websites or recognition of handwriting that allow an increase of accuracy by exploiting structural consistencies.

3.2 Stacked Inference

Sequence labeling methods like CRFs assign a sequence of labels $\mathbf{y} = (y_1, \ldots, y_T)$ to a given sequence of observed tokens $\mathbf{x} = (x_1, \ldots, x_T)$. Figure 1 contains two examples to illustrate the assignment. Let $crf(\mathbf{x}, \Lambda, F) = \mathbf{y}$ be the function that applies the CRF model with the weights $\Lambda = \{\lambda_1, \ldots, \lambda_K\}$ and the set of feature functions $F = \{f_1, \ldots, f_K\}$ on the input sequence $\mathbf{x}$ and returns the labeling result $\mathbf{y}$. The set of model weights must of course correspond to the set of feature functions. Since the CRF processes this sequence of tokens in one labeling task, we call $\mathbf{x}$ an instance. All instances together form the dataset D which is split in a disjoint training and testing subset. An information or entity consists often of several tokens and are encoded by a sequence of equal labels. We assume here that the given labels already specify an unambiguous encoding. An instance itself may contain multiple entities specified by arbitrary amount of labels, one label for each token of the input sequences. Furthermore, we assume that the dataset $D = \{C_1, \ldots, C_n\}$ can be completely and disjointly partitioned into subsets of instances $\mathbf{x}$ that originate from the same creation context C_i. Similar to the relational template in (Kou and Cohen, 2007), we imply that a trivial context template exists for the assignment of the context set. Staying with the previous examples, the reference section of this paper defines a context C with 22 instances.

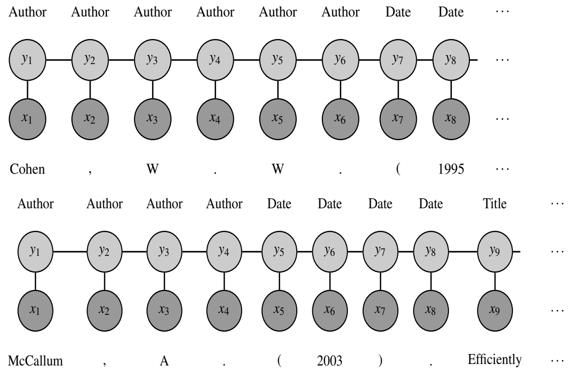
Author Author Author Author Author Author Date Date ···
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8 ···
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 ···
Cohen , W . W . (1995 ···
Author Author Author Author Date Date Date Date Title ···
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8 y_9 ···
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 ···
McCallum , A . (2003) . Efficiently ···

Figure 1: The start of the fourth and the fourteenth reference of the reference section of this paper displayed as a linear chain with correct labels. The indices for $\mathbf{x}$ and $\mathbf{y}$ start at 1 for each instance.

In stacked graphical learning, several models can be stacked in a sequence. Experimental results, e.g., of Kou (Kou and Cohen, 2007), have shown that this approach already converges with a depth of two learners and no significant improvements are achieved with more iterations of stacking. Therefore, we only apply stacked graphical learning with CRF in a two-stage approach like Krishnan and Manning (Krishnan and Manning, 2006). In order to extract entities collectively, we define the stacked inference task on the complete set of instances $\mathbf{x}$ in one context C. The two CRFs, however, label the single instances within that context separately as usual. The following algorithm summarizes the stacked inference combined with online rule learning. Section 3.3 describes the rule learning techniques for the identification of structural consistencies and how the "meta-features" f^m are induced. Details about the estimation of the weights (e.g., Λ^m) are discussed in Section 3.4.

1. **Apply base CRF**
   Apply $crf(\mathbf{x}, \Lambda, F) = \hat{\mathbf{y}}$ on all instances $\mathbf{x} \in C$ in order to create the initial label sequences $\hat{\mathbf{y}}$.

2. **Learn structural consistencies**
   Create a tabular database T by combining all instances $\mathbf{x} \in C$, their corresponding labeled sequences $\hat{\mathbf{y}}$ and a feature set $F' \subseteq F$. Learn classification rules for the target attributes and construct a feature function $f^m \in F^m$ for each discovered rule.

3. **Apply stacked CRF**
   Apply $crf(\mathbf{x}, \Lambda \cup \Lambda^m, F \cup F^m) = \mathbf{y}$ again on all instances $\mathbf{x} \in C$ in order to create the final label sequences $\mathbf{y}$.

3.3 Learning Structural Consistencies during Inference

First, the overall idea how structural consistencies are learned during the inference is addressed. The technical details are then described after a short example.

We apply rule learning techniques on all (probably erroneously) label assignments $\hat{\mathbf{y}} \in C$ of the base CRF. The rules are learned in order to classify certain label transitions and, thus, describe the shared properties of the transition within the context C. The labeling error in the input data is usually eliminated by the generalization of the rule learning algorithm. The label transition is optimally described by a single pattern that covers the majority of transitions despite of erroneously outliers. The learned rules are then used as binary feature functions in the same context C: They return 1 if the the rule applies on the observed token x_t, and 0 otherwise. We gain therefore additional features that indicate label transitions if the instances are consistently structured. Even if the learned rules are misleading due to erroneously input data or missing consistency of the instances, their discriminative impact on the inference is yet weighted by the learning algorithm of the stacked CRF.

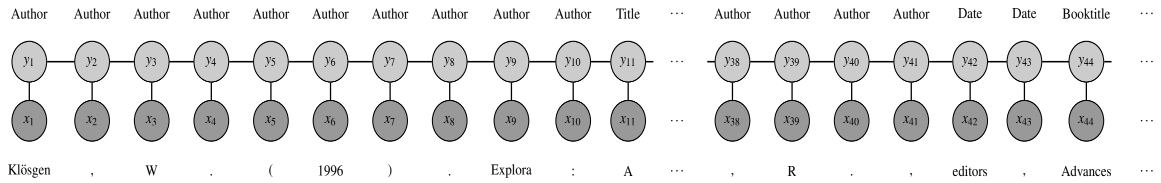

Figure 2: Two excerpts of the ninth reference with erroneous labeling: The date (token x_5 to x_8) and the begin of the title (token x_9 and x_{10}) was falsely labeled as author, e.g., due to the high weight of the colon for the end of an author. The editor was additionally labeled as an author (up to token x_{41}) and date (token x_{42} and x_{43}).

This process is illustrated by a simple example concerning the author label, but can also be applied to any other label. Let the reference section of this paper be processed by the base CRF that classified all instances but one correctly as in Figure 1. For some reasons the base CRF missed the date and editor and misclassified the tokens x_5 to x_{10} and the tokens x_{18} to x_{43} in the ninth reference (c.f. Figure 2). The input of the rule learning now consists of 22 transitions from author to date whereas one transition is incorrect. In this case, a reasonable result of the rule learning is the rule "if the token x_t is a period and followed by a parenthesis, then there is a transition from author to date at t". Converted to a feature function, this rule returns 1 for token x_4 and 0 for all other tokens of the reference in Figure 2. The weight of this new feature function is then estimated by the stacked CRF. Therefore, the stacked CRFs' likelihood of an transition from author to date is increased at the token x_4 and decreased at the token x_{41} due to the presence or absence of the meta-features.

In general, any classification method can be applied to learn indicators for the structural consistencies. In this work, we restrict ourselves to techniques for supervised descriptive rule discovery because their learning and inference algorithm are efficient and the resulting rules can be interpreted. This allows studies about the properties of good descriptions of structural consistencies. We disregarded the usage of the Support Vector Machines (Cortes and Vapnik, 1995) because several models need to be trained and executed during the stacked inference.

For inducing the meta-features, first one tabular database $T = (I, A)$ is created for each context C as the input of the rule learning techniques described in Section 2.3. The database is constructed using all instances $\mathbf{x} \in C$, their corresponding initially labeled sequences $\hat{\mathbf{y}}$ and a feature set $F' \subseteq F$. Each individual of I corresponds to a single token of the instances in C. The set of attributes A consists of the possible labels and a superset of F': When classification methods are applied on sequential data, the attributes are also added for a fixed window, e.g., the attribute "WORD@-1=proceedings" indicates that the token before the current individual equals the string "proceedings". Hence, this superset contains the feature functions F and additionally their manifestation in window defined by the window size w. The cells in the tabular database T are filled with binary values. They are set to true if the feature or label occurs at this token and to false otherwise.

In a next step, the target attributes for the rule learning are specified. In this work, we apply the transition of two different labels. Here, the target attribute is set on all transitions of two dedicated labels in the initially labeled result $\hat{\mathbf{y}}$ of the context C.

Finally, the set of learned rules are transformed to the set of binary feature functions F^m that return true, if the condition of the respective rule applies.

3.4 Parameter Estimation

The weights of two models need to be estimated for the presented approach: the parameters of the base model and of the stacked model. The base model needs to be applied on the training instances for the estimation of the weights of the stacked model, i.e., step 1 and step 2 of the stacked inference in Section 3.2 need to be performed on the training set. If the weights of the base model are estimated as usual using the labeled training instances, then it produces unrealistic prediction on these instances and the meta-features of the stacked model are over fitted resulting in a decrease of accuracy. Since the base model is optimized in this case on the training instances, it labels these instances perfectly. The learned rules create optimal descriptions of the structural consistencies and the stacked model assigns biased weights to the meta-features. This is of course not reproducible when processing unseen data.

The simple solution to this problem is a cross-fold training of the base model for the training of the stacked CRF as described in Section 2.2 and successfully applied by several approaches (Kou and Cohen, 2007; Krishnan and Manning, 2006). Training of the base model in a cross-fold fashion is also a very good solution for the presented approach, but we simply decrease the accuracy of the model by reducing the training iterations. Thus, only one model needs to be trained for the learning phase of the stacked model. For the testing phase or common application however, a single base model learned with the default settings is applied.

The model of the stacked CRF is trained dependent on the base model and the creation context C that are both applied to induce the new features online during the stacked inference. The weights $\Lambda = \{\lambda_1, \dots, \lambda_K\}$ and $\Lambda^m = \{\lambda_1^m, \dots, \lambda_M^m\}$ of the stacked CRF are estimated to maximize the conditional probability on the instances of the training dataset:

$$P_\lambda\Big(\mathbf{y}|\mathbf{x}, C, crf(\mathbf{x}, \Lambda', F)\Big) = \frac{1}{Z_\mathbf{x}} \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{K} \lambda_i \cdot f_i(y_{t-1}, y_t, \mathbf{x}, t) + \sum_{t=1}^{T}\sum_{j=1}^{M} \lambda_j^m \cdot f_j^m\Big(y_{t-1}, y_t, \mathbf{x}, t, C, crf(\mathbf{x}, \Lambda', F)\Big)\right)$$

The resulting model still relies on the normal features functions but is extended with dynamic and high quality features that help to resolve ambiguities and substitute for other missing features. These meta-features possess the same meaning in the complete dataset, but change their interpretation or manifestation dependent on the currently processed creation context. They provide overall a very good description of the structural consistencies and are often alone sufficient for a classification of the entities.

A short example: The induced feature function for the transition of the author to the date are set to very high weights for the corresponding state transition of the learned model. As illustrated in the example of Section 3.3, this feature function returns 1 in the reference section of this paper for a token which is a period and is followed by a parenthesis. In other reference sections with a different style guide applied, the feature function for this state transition returns 1, if the token is a colon and is followed by a capitalized word. However, both examples refer only to exactly one feature function that dynamically adapts to the currently processed context.

4 Experimental Results

The presented approach is evaluated in the domain of reference segmentation. The common approach is to separately process the instances, namely the references. Within these references, the interesting entities need to be identified. Since all tokens of a reference are part of exactly one entity, one speaks of a segmentation task. In this section, we introduce the overall settings and present the experimental results.

4.1 Datasets

All available and commonly used datasets for the segmentation of references are a listing of references without their creation context and are thus not applicable for the evaluation of the presented approach. Therefore, a new dataset was manually annotated with the label set of Peng and McCallum (Peng and McCallum, 2004) concerning the fields *Author*, *Booktitle*, *Date*, *Editor*, *Institution*, *Journal*, *Location*, *Note*, *Pages*, *Publisher*, *Tech*, *Title* and *Volume*. The resulting dataset contains 566 references in 23 documents extracted only of complete reference sections of real publications. The amount of instances is comparable to previously published evaluations in this domain, c.f., (Peng and McCallum, 2004; Councill et al., 2008).

Two different sets of features are used in the experimental study: The basic features are applied for all evaluated models and correspond to the features of well-known evaluations in this domain, c.f., (Peng and McCallum, 2004; Councill et al., 2008). For an extensive definition of the set of basic features, we refer to the dataset that contains all applied basic features. Only a part of the basic features is used for the induction of the meta-features, omitting ngram and token window features. This restriction is justified with their minimal expressiveness for the identification of the structure in relation to the increase of the search space.

The dataset with all applied basic features can be freely downloaded[1].

4.2 Implementation Details

The machine learning toolkit Mallet[2] is used for an implementation of the CRF in the presented approach. For rule learning, we integrated two different methods. We chose a subgroup discovery implementation[3] because of the multifaceted configuration options that allow a deep study of the approach's limits. Additionally, we applied an established association rule learner Ripper (Cohen, 1995) for a comparable implemention[4].

[1] http://www.is.informatik.uni-wuerzburg.de/staff/kluegl_peter/research/

[2] http://mallet.cs.umass.edu

[3] http://sourceforge.net/projects/vikamine/

[4] http://sourceforge.net/projects/weka/

Table 1: Overview of the evaluated models

CRF	A single CRF trained on the same data and features.
STACKED CRF	A two-stage CRF approach. The predictions of the base CRF are added as features to the stacked CRF.
STACKED+DESCRIPTIVE	The default approach of stacked CRF combined with subgroup discovery for rule learning. Only transitions between the labels *Author*, *Title*, *Date* and *Pages* that commonly occur in most references are considered.
STACKED+RIPPER	A stacked CRF combined with the association rule learner Ripper. Only the four most common labels are addressed.
STACKED+MORE	A stacked approach using subgroup discovery that additionally learns the transitions of the labels *Booktitle*, *Journal* and *Volume*.
STACKED+MAX	A stacked approach using subgroup discovery that considers the transitions of all labels for the rule learning task.

We used only the default parameters for the CRF and all evaluated models were trained until convergence. Only for the training of the stacked model, the iterations of the base model was reduced to 50 iterations. For the default configuration of both rule learning tasks, we set the window size $w = 1$. Additionally for the default setting of the subgroup discovery, we used a quality function based on the F_1 measure, selected only one rule for each description of a label, restricted the length of the rules to maximal three selectors, and set an overall minimum threshold of the quality of a rule equal to 0.5.

The presented approach is overall easy to implement and only established standard methods are used. Its inference is still efficient in contrast to complex models with approximate inference techniques.

4.3 Performance Measure

The performance is measured with commonly used methods of the domain. Let tp be the number of true positive classified tokens and define fn and fp respectively for false negatives and false positives. Since punctuations contain no target information in this domain, only alpha-numeric tokens are considered. *Precision*, *recall* and F_1 are computed by:

$$precision = \frac{tp}{tp+fp}, \quad recall = \frac{tp}{tp+fn},$$
$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}.$$

4.4 Results

The presented approach is compared to two base line models in a five fold cross evaluation. Four different settings of stacked CRFs combined with a rule learning technique are investigated. A detailed description of all evaluated models is given in Table 1. The documents of the dataset are randomly distributed over the five folds.

The results of the experimental study are depicted in Table 2. Only marginal differences can be observed between the two base line models CRF and STACKED CRF. This indicates that the normal stacking approach cannot exploit the structural consistencies or gain much advantage of the predicted labels.

Table 2: F_1 scores averaged over the five folds.

	average F_1
Base Line	
CRF	91.3
STACKED CRF	91.8
Our Approach	
STACKED+DESCRIPTIVE	94.0
STACKED+RIPPER	93.6
Variants of DESCRIPTIVE	
STACKED+MORE	94.1
STACKED+MAX	93.6

All of our stacked models combined with rule learning techniques significantly outperform the base line models using a one-sided, paired t-tests on the F_1 scores of the single references ($p \ll 0.01$). Comparing the results of STACKED+DESCRIPTIVE that only considers the consistencies of four labels to the base line CRF, our approach achieves an average error reduction of over 30% on the real-world dataset.

The lower F_1 scores of STACKED+RIPPER can be explained by its learning algorithm. The Ripper implementation applies a coverage-based learning in order to create a set of rules, which together classify the target attribute. This can lead to a reproduction of errors of the predicted labels in the description of the structure. In the domain of reference segmentation a single description of the structure is preferable. However, in other domains where disjoint consistencies of one transition can be found, a covering algorithm for inducing the rules performs probably better.

The second configuration with a subgroup discovering technique STACKED+MORE considers the transition between seven labels and is able to slightly increase the measured F_1 score compared to our default model STACKED+DESCRIPTIVE. STACKED+MAX that induces rules for all labels achieves only an average error reduction of 26% compared to a single CRF. This is mainly caused by misleading meta-features for rare labels. The task of learning consistencies from a minimal amount of examples is error-prone and can decrease the accuracy, especially if the examples are labeled incorrectly.

Table 3 provides closer insights in the benefit of the presented approach using the author label as an example. STACKED+DESCRIPTIVE is able is significantly improve the labeling accuracy for all folds but one. The third fold contains an unfavorable distribution of style guides between the training and testing set for the author. If the initial base CRF labels a label systematically incorrectly, then the rule learning cannot induce any valuable and correct descriptions of the structure. Nevertheless, an average error reduction of over 50% is achieved for identifying the author of the reference.

Table 3: F_1 scores of the author label.

	CRF	STACKED+ DESCRIPTIVE	error reduction
Fold 1	97.7	99.6	82.6%
Fold 2	97.0	99.2	73.3%
Fold 3	96.4	96.5	2.8%
Fold 4	97.1	98.8	58.6%
Fold 5	89.5	95.1	53.3%
average	95.5	97.8	51.6%

To our knowledge, no domain-independent approach was published that can be utilized for a comparable model. As comparison, we applied the skip-chain approach of (Sutton and McCallum, 2004) with factors for capitalized words and additionally for identical punctuation marks, but no improvement over the base line models could be measured. Furthermore, the feature induction for CRFs (McCallum, 2003) was integrated, but resulted counter-intuitively in a decrease of the accuracy.

The performance time of the presented approach for one fold averaged over the five folds is several times faster than a higher-order model with skip edges, about nine times faster using the subgroup discovery and about fourteen times faster using Ripper. The difference in speed is less compared to previously published evaluations (Kou and Cohen, 2007). This is mainly caused by the fact that the rule learning is neither optimized for this task nor for the domain, e.g., by pruning the attributes.

The presented approach significantly outperforms the common CRF without any additional domain knowledge, integrated matching methods with a bibliographic database or other jointly performed tasks like entity resolution. Nevertheless, the approach stays way behind its potential. The meta-features specify most of the time a perfect classification of the boundaries and transitions, but the stacked CRF still labels the entities erroneously. To provide the knowledge on a simple feature level may not sufficient to adapt the model to the structure of the current creation context.

5 Related Work

In the following, we give a brief overview on related work coming from different domains with context consistencies, attempts utilizing complex graphical models and stacked graphical models for collective information extraction, and approaches on feature induction.

Especially for Named Entity Recognition (NER) modelling long-distance dependencies is crucial. The labeling of an entity is quite consistent within a given document, however, conclusive discriminating features are sparsely spread across the document. As a consequence, leveraging predictions of one instance to disambiguate others is essential. Bunescu et al. (Bunescu and Mooney, 2004), Sutton et al. (Sutton and McCallum, 2004) and Finkel et al. (Finkel et al., 2005) extended the commonly applied linear chain CRF to higher order structures. The exponential increase in model complexity enforces to switch from exact to approximate inference techniques. Stacked graphical models (Kou and Cohen, 2007; Krishnan and Manning, 2006) retain exact inference as well as efficiency by using linear chain CRF.

In Kou and Cohen's Stacked Graphical Models framework (Kou and Cohen, 2007), information is propagated by Relational Templates C. Although each C may find related instances and aggregate their labels in a possibly complex manner, they utilize rather simple aggregators, e.g. COUNT and EXISTS. Likewise, the approach of Krishnan and Manning (Krishnan and Manning, 2006) uses straightforward but for NER efficient aggregate features, concentrating on the label predictions of the same entity in other instances. In contrast to Stacked Graphical Models, they also include corpus-level features, aggregating predictions across documents. In this paper, we use data mining techniques to determine rich context sensitively applied features. Rather than simply transferring labels of related instances, e.g., by majority vote aggregation, we exploit structural properties of a given context. We represent the gathered context knowledge by several meta features which are conceptually independent of the label types.

A semi-supervised approach on exploiting structural consistencies of documents has been taken by Arnold and Cohen (Arnold and Cohen, 2008) who improve domain adaption by conditional frequency information of the unlabeled data. They show that differences in the frequency distribution of tokens across different sections in biological research papers can provide useful information to extract protein names. Counting frequencies can be done efficiently and the experimental results suggest that these features are robust across documents. However, in general unlabeled data is not enough to model the context structure, e.g., frequency information can be noisy or differences in the frequency distribution may be caused non-structural. We propose to mine the distributions of predicted labels and their combinations with observed features to capture context structure.

Yang et al. use structural consistencies for information extraction from web forums (Yang et al., 2009). They employ Markov Logic Networks (Richardson and Domingos, 2006) with formulas to encode the assumed structural properties of a typical forum page, e.g., characteristic link structures or tag and attribute similarities among different posts and sites. Since context structure is represented inside of the graphical model, inference and learning have to fight model complexity. Another example for content extraction from websites that exploits related instances is Gulhane et al. (Gulhane et al., 2010). They assume two properties of web information: The values of an attribute distributed over different pages are similar for equal entities and the pages of one website share a similar structure due to the creation template. In contrast to those two approaches, the work presented in this paper relies on no structural knowledge previously known about the domain.

McCallum contributed an improvement for CRF applications through feature induction (McCallum, 2003). Based on a given training set useful combinations of features are computed, reducing the number of model parameters. The feature induction of our approach is performed online during processing the document and applies flexible data mining techniques to specify the properties of consistent label transitions. Learning Flexible Features for Conditional Random Fields (Stewart et al., 2008) is an approach by Steward et al. that also induces features. They propose Conditional Fields Of Experts (CFOE) as CRF augmented with latent hidden states.

6 Conclusions

We have presented a novel approach for collective information extraction using a combination of two CRFs together with rule learning techniques to induce new features during inference. The initial results of the first CRF are exploited to gain information about the structural consistencies. Then, the second CRF is automatically adapted to the previously unknown composition of the entities. This is achieved by changing the manifestation of its features dependent on the currently processed set of instances. To our best knowledge, no similar and domain-independent approach was published that is able to exploit the structural consistencies in textual data. The results on a real-world dataset for the segmentation of references indicate a significant improvement towards the commonly applied models.

For future work, better ways to include the high quality descriptions need to be found for exploiting the structural consistencies. One possibility is the Generalized Expectation Criteria of Mann and McCallum (Mann and McCallum, 2010) that allow to learn with constraints that cover more expressive and structural dependencies than the underlying model. Another approach that can solve this problem is to use a complex model for joint inference despite the more expensive inference. Instead of performing two different task in the same inference for segmentation and matching (Poon and Domingos, 2007; Singh et al., 2009), a joint model can be used to infer the labeled sequence together with the description of the structural properties.

REFERENCES

Arnold, A. and Cohen, W. W. (2008). Intra-document Structural Frequency Features for Semi-supervised Domain Adaptation. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1291–1300. ACM.

Batal, I. and Hauskrecht, M. (2010). Constructing Classification Features using Minimal Predictive Patterns. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 869–878, New York, NY, USA. ACM.

Bunescu, R. and Mooney, R. J. (2004). Collective Information Extraction with Relational Markov Networks. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cohen, W. W. (1995). Fast Effective Rule Induction. In *Proceedings of the Twelfth Int. Conference on Machine Learning*, pages 115–123. Morgan Kaufmann.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Councill, I., Giles, C. L., and Kan, M.-Y. (2008). ParsCit: an Open-source CRF Reference String Parsing Package. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. ELRA.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gulhane, P., Rastogi, R., Sengamedu, S. H., and Tengli, A. (2010). Exploiting Content Redundancy for Web Information Extraction. *Proc. VLDB Endow.*, 3:578–587.

Klösgen, W. (1996). Explora: A Multipattern and Multistrategy Discovery Assistant. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI Press.

Kou, Z. and Cohen, W. W. (2007). Stacked Graphical Models for Efficient Inference in Markov Random Fields. In *Proceedings of the 2007 SIAM Int. Conf. on Data Mining*.

Krishnan, V. and Manning, C. D. (2006). An Effective two-stage Model for Exploiting non-local Dependencies in Named Entity Recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 1121–1128, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proc. 18th International Conf. on Machine Learning*, pages 282–289.

Mann, G. S. and McCallum, A. (2010). Generalized Expectation Criteria for Semi-Supervised Learning with Weakly Labeled Data. *J. Mach. Learn. Res.*, 11:955–984.

McCallum, A. (2003). Efficiently Inducing Features of Conditional Random Fields. In *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*.

Peng, F. and McCallum, A. (2004). Accurate Information Extraction from Research Papers using Conditional Random Fields. In *HLT-NAACL*, pages 329–336.

Poon, H. and Domingos, P. (2007). Joint Inference in Information Extraction. In *AAAI'07: Proceedings of the 22nd National Conference on Artificial intelligence*, pages 913–918. AAAI Press.

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

Singh, S., Schultz, K., and McCallum, A. (2009). Bi-directional Joint Inference for Entity Resolution and Segmentation Using Imperatively-Defined Factor Graphs. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 414–429. Springer-Verlag.

Stewart, L., He, X., and Zemel, R. S. (2008). Learning Flexible Features for Conditional Random Fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1415–1426.

Sutton, C. and McCallum, A. (2004). Collective Segmentation and Labeling of Distant Entities in Information Extraction. In *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*.

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*, 5:241–259.

Yang, J.-M., Cai, R., Wang, Y., Zhu, J., Zhang, L., and Ma, W.-Y. (2009). Incorporating Site-level Knowledge to Extract Structured Data from Web Forums. In *Proceedings of the 18th international conference on World wide web*, pages 181–190. ACM.