Conditional Random Fields For Local Adaptive Reference Extraction

Martin Toepfer and **Peter Kluegl** and **Andreas Hotho** and **Frank Puppe**
University of Würzburg,
Department of Computer Science VI
Am Hubland, 97074 Würzburg, Germany
{toepfer, pkluegl, hotho, puppe}@informatik.uni-wuerzburg.de

Abstract

The accurate extraction of bibliographic information from scientific publications is an active field of research. Machine learning, especially sequence labeling approaches like Conditional Random Fields (CRF), are often applied for this reference extraction task, but still suffer from the ambiguity of reference notation. Reference sections apply a predefined style guide and contain only homogeneous references. Therefore, other references of the same paper or journal often can provide evidence how the fields of a reference are correctly labeled. We propose a novel approach that exploits the similarities within a document. Our process model uses information of unlabeled documents directly during the extraction task in order to automatically adapt to the perceived style guide. This is implemented by changing the manifestation of the features for the applied CRF. The experimental results show considerable improvements compared to the common approach. We achieve an average F_1 score of 96.7% and an instance accuracy of 85.4% on the test data set.

1 Introduction

Reference sections of research papers are a valuable source for many interesting applications. A considerable amount of research has been spent on creating and analyzing citation graphs, yielding information about research communities and topics. Social bookmarking services like Bibsonomy[1] on the other hand have become essential tools for researchers and facilitate the management of bibliographic data. Both applications, citation analysis and bookmarking services, rely on a structured representation of the reference data. Often the well-known BibTex format is used to define the different fields of an information. The acquisition of this structured data demands for an automatic processing of the vast amount of the unstructured data available in publications.

The knowledge for an automatic extraction of references can be formalized using rules or templates. However, the handcrafting of rules is tedious and prone to error due to the knowledge engineering bottleneck. Several publications have shown that machine learning and especially sequence labeling approaches are more suitable for the reference extraction task [Peng and McCallum, 2004; Councill *et al.*, 2008]. These methods learn a statistical model using training sets where the interesting information, namely the BibTeX fields, is already labeled. The model is applied on newly and unseen documents in order to identify the information in unlabeled data. Hence, the model is only adapted offline on the previously seen documents of the training phase. Although these approaches achieve remarkable results, the heterogeneous styles of the references make a suitable generalization difficult and decrease the accuracy of the extraction task. The IEEE style, for example, separates the author and the title with a comma and surrounds the title with quotes. Whereas the ACM style applies no separator for the author and the date is located between the author and the title. The MISQ style surrounds the title also with quotes, but uses no separator for the author. Nevertheless, the input data of the extraction task, i.e., the reference section of scientific publications, follows a single style guide. The references within a paper or journal are usually homogenous. In order to utilize these local consistencies the model has to be adapted during the extraction phase, because the applied style guide is identified as the document is processed. This is not possible using the common process model.

In this paper, we propose a local adaptive information extraction approach using sequence labeling methods, especially Conditional Random Fields. That is a novel extension of the common process model with an automatic adaption to the previously unknown style guide. We apply two stacked models that are trained offline. The first model is applied to gain information about the document's structure and to create a description of the reference notation based on the available features. The style guides differ in their characteristics of field separation and alignment. As a result, the description, also called local model, is based on different features dependent on the currently processed document. This information is used to create style-specific features which have a steady meaning for the information extraction task. However, the manifestation of these meta features differs between documents and depends on the applied style guide. The new features are then added to the features of the second model helping to resolve ambiguities and to increase the extraction accuracy. As a result, the presented approach achieves considerably better results than a single Conditional Random Field.

The rest of the paper is structured as follows: Section 2 introduces the novel combination of methods and gives a detailed description of all parts of the process model. Then, the evaluation setting and experimental results are presented and discussed in section 3. Section 4 gives a short overview of the related work and section 5 concludes with a summary of the presented work.

[1] http://www.bibsonomy.org/

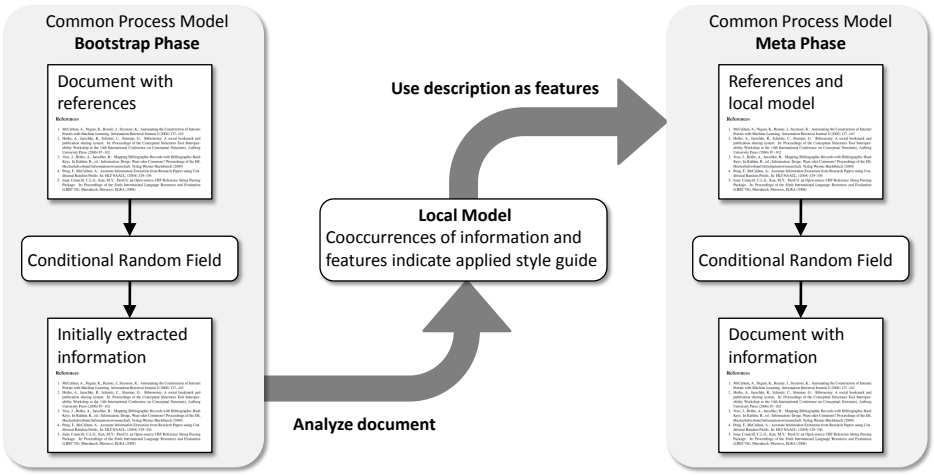

Figure 1: Overview of the applied process model with three phases: the bootstrap, local model and meta phase.

2 Method

Machine learning and sequence labeling approaches are often applied for reference extraction and Conditional Random Fields (CRF) are one of the most popular techniques for this task. Normally, a simple process model is used: The feature extraction identifies valuable properties in the unstructured data. They are used by a given model in order to extract interesting information, that is, labeling the fields of a reference. The extraction model is trained on labeled examples in a previous phase. Therefore, the model is only adapted offline in the learning process on the global consistencies of the domain. This prevents a good generalization for the global model and induces errors in the extracted information. In order to overcome this problem, the local patterns and the consistency of one document, the applied style guide in this domain respectively, need to be addressed directly for a resolution of the ambiguity. The style guide can however be identified as soon as the document is processed. Hence, the common process model is incompatible to an online adaption during the extraction process on the local consistency.

The presented approach tries to utilize the common process model with CRFs in a novel combination. The unlabeled documents are used to identify the applied style guide directly during the extraction process. Then, this information about the homogenous notation within the current document is exploited to increase the extraction accuracy in an additional phase. The model is learnt offline, but the features it is based on are adapted online during the extraction process of each single document. Since the process can adjust to the local consistencies, it is called *local adaptive*. Figure 1 provides an overview of the applied process that consists of three stages: the bootstrap, the construction of the local model and the meta phase. The purpose of the *bootstrap phase* is to provide the fundamental information that is needed to perceive style information for a document. We employ a common process model as it can be found in previous CRF approaches. Features are extracted from the current document and a previously learnt (base) model is applied in order to gain information. However, this is just an intermediate step required for an examination of the local information patterns. The second phase, the construction of the *local model*, tries to create a description of the applied style guide. This is achieved by investigating the cooccurrence of information and features and the selection of features that describe the different characteristics of the style guide very well. Finally, the acquired style information is used to create special features, called meta features. These possess a different manifestation for each document. The *meta phase* is the final step of the process model. It is built on the common process model of conditional random fields, but uses an enhanced set of features. Additionally to the base features of the bootstrap, it also considers the new meta features that provide hints on how the information is structured in the applied style guide.

In summary, the bootstrap phase takes an initial look at the reference section and handles apparent style information over to the meta phase which finally processes the reference section as if the applied style guide was known.

For a detailed description of the process, first the applied terminology is presented. Then, conditional random fields and the usage of features are addressed. The elements of the local model that contain the knowledge of the document's structure build an important part of the approach and are described in detail with an example.

2.1 Terminology

In the presented approach different frameworks and toolkits are combined. In order to clarify the terminology we explain some central terms. We use a nomenclature oriented at the Apache UIMA framework [Ferrucci and Lally, 2004].

Definition 1 (Typesystem, Information Type)**.** *A* typesystem *is a set* $\mathbb{T}$*, whose elements are called* (annotation or information) types*.*

As an example, we define the label type system $\mathbb{T}_{\text{label}} = \{\text{AUTHOR, BOOKTITLE, DATE, EDITOR, INSTITUTION, JOURNAL, LOCATION, NOTE, PAGES, PUBLISHER, TECH, TITLE, VOLUME}\}$ which contains all field labels. Furthermore, we introduce the overall typesystem $\mathbb{T}_{\text{all}}$ which contains all types.

Definition 2 (Annotation)**.** *Given a text document* D *and a typesystem* $\mathbb{T}$*, we define an* annotation *as a triplet* $(s, i, j) \in \mathbb{T} \times \mathbb{N} \times \mathbb{N}$*, consisting of an information type* $s \in \mathbb{T}$ *and two naturals* $i \leq j$*, indicating the begin and the end of the annotation in* D*.*

For instance, we can assign an annotation $(\text{NUM}, 28, 32)$ to a document to state that the

text covered by the offsets 28 and 32 is a number. Therefore, we define an appropriate typesystem $\mathbb{T}_{\text{feat}} = \{\text{COMMA, CW, SW, NUM, FirstName}\ldots\}$ with $\mathbb{T}_{\text{feat}} \cap \mathbb{T}_{\text{label}} = \emptyset$. The typesystem $\mathbb{T}_{\text{feat}}$ contains several useful low level information types called features, e.g., COMMA indicating commas, NUM indicating numbers, CW for capitalized words, SW for lower case words and FirstName indicating first names. These annotations are automatically assigned by the feature extraction, e.g. a word list with first names is provided and for each occurrences of an entry an annotation of the type FirstName is created.

Moreover, we partition documents into pieces of atomic lexical units, called *tokens*, to make use of the ClearTK framework [Ogren *et al.*, 2008] and the machine learning toolkit Mallet[2] for the implementation of the CRF.

Definition 3 (Token)**.** *We postulate $\tau \in \mathbb{T}_{token}$ (the Token-Type) to be a type which satisfies the following conditions.*

- *Annotations of the type τ do not cover white space characters and*
- *all other characters are covered of exactly one annotation of the type τ.*

Annotations of the type τ are called tokens*.*

Punctuations and special characters are put in single tokens. Alphabetic and numerical character sequences are split into separate token sets.

Definition 4 (Feature)**.** *Iff a token x_t is within*[3] *an annotation of a type $\varphi \in \mathbb{T}_{feat}$, we say that x_t has the* feature *φ.*

If a token $x_a = (\tau, 6, 7)$ has the feature COMMA $\in \mathbb{T}_{\text{feat}}$, then the text covered by the token is a comma. The terms feature and type are used synonymously in the following sections.

2.2 Conditional Random Fields

Conditional Random Fields (CRF) [Lafferty *et al.*, 2001] model conditional probabilities with undirected graphs. As usual in information extraction and sequence labeling tasks, we use linear chain CRFs. That is, we take a sequence of tokens $\mathbf{x} = (x_1, \ldots, x_T)$ as input. Given binary feature functions $f_1, \ldots, f_K$ and parameters $\lambda_1, \ldots, \lambda_K \in \mathbb{R}$, we compute the conditional probability of the label sequence $\mathbf{y} = (y_1, \ldots, y_T)$ under $\mathbf{x}$ by

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

with a normalization factor

$$Z_\mathbf{x} = \sum_{\mathbf{y}' \in Y} \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{K} \lambda_i \cdot f_i(y'_{t-1}, y'_t, \mathbf{x}, t)\right).$$

Y is the set of all possible label sequences $\mathbf{y}'$ for $\mathbf{x}$.

In short, a feature function $f_i(y_{t-1}, y_t, \mathbf{x}, t))$ can testify evidence for the token at the position t to be labeled as y_t, depending on the label of it's predecessor and the observed input sequence. The feature functions are weighted by parameters $\lambda_1, \ldots, \lambda_K$. Hence, if $f_i(y_{t-1}, y_t, \mathbf{x}, t)) = 1$ and λ_i has a high value, then we have strong evidence for labeling x_t as y_t. Accordingly, the parameters determine how we infer the labels from the information given by the feature functions, i.e., we assume the label sequence that is most likely given some observation sequence. As usual in supervised machine learning, we use a learning algorithm which sets the weights to make good predictions on a training set.

In principle, a feature function can make complex use of the whole input sequence, the current label and the predecessing label. However, we mainly use simpler feature functions, named *annotation-based feature functions*, which factorize into two parts. Given two labels $y_a, y_b \in \mathbb{T}_{\text{label}}$, a typesystem $\mathbb{T}$ and a type $\varphi \in \mathbb{T}_{\text{feat}}$, an annotation based feature function has the form:

$$f_{\varphi,y_a,y_b}(y_{t-1}, y_t, \mathbf{x}, t) = \mathbf{1}_{\{y_{t-1}=y_a\}} \cdot \mathbf{1}_{\{y_t=y_b\}} \cdot f_\varphi(x_t).$$

The first part is only an indication of the label transition and ensures that we can learn separate weights for each combination of labels. On the contrary, the second part is independent from the labels. $f_\varphi(x_t)$ just shows if the token at the position t has the feature φ. In different words,

$$f_\varphi(x_t) = \begin{cases} 1, & \text{if } x_t \text{ has the feature } \varphi, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, we create annotation based feature functions f_{φ,y_a,y_b} for every $y_a, y_b \in \mathbb{T}_{\text{label}}$ and every type $\varphi \in \mathbb{T}_{\text{feat}}$. By example, the CRF learns a parameter $\lambda_{\text{NUM,AUTHOR,YEAR}}$ for the feature function $f_{\text{NUM,AUTHOR,YEAR}}$, i.e., a weight for having the NUM (number) feature and transitioning from an author field to a year field.

2.3 Local Adaptivity

The local model phase is the main part of the local adaptivity. It analyses the given features and the initially extracted information of the bootstrap phase and creates a description of the characteristics of the applied style guide. This description is then projected as features in order to be useful for the CRF in the meta phase. Therefore, the local model consists of two major steps: the creation of the description and the projection of features. Overall, a local model can be seen as a representation of specific knowledge of each single document's structure. There are various means to describe these local patterns of a document. The rule-based approach for local adaptivity [Kluegl *et al.*, 2010] has shown that two characteristics are describing the applied style guide sufficiently for a considerable increase of accuracy.

field separation One important consistency inside a reference section is the way how fields are separated. For instance, one writer always ends the author lists with a period, another writer may use a colon instead. If such a field separator is once determined with the help of other references, then it can help solving ambiguos cases, for example, in the case when one of the first tokens of the title also contains a colon. For every label type $\varphi_{\text{label}} \in \mathbb{T}_{\text{label}}$ we try to detect features which indicate the begin or the end of φ_{label} fields in a document. These additional features $\text{BEGIN}_{\varphi_{\text{label}}}$ and $\text{END}_{\varphi_{\text{label}}} \in \mathbb{T}_{\text{meta-feat}}$ then indicate the document specific seperators for the meta phase. For instance, if a token x_t has the feature LPAREN and we have recognized that date fields begin with a left parenthesis in this document, then we assign an annotation of the type $\text{BEGIN}_{\text{DATE}} \in \mathbb{T}_{\text{meta-feat}}$ to x_t to state that

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

[3] A token (τ, a, b) is within an annotation (φ, x, y), iff a $\geq$ x and b $\leq$ y.

we have evidence for the begin of a date field. In addition to these two meta features, we also introduce two specialized meta features $\text{BEGIN}^t_{\varphi_{\text{label}}}$ and $\text{END}^t_{\varphi_{\text{label}}} \in \mathbb{T}_{\text{meta-feat}}$ that restrict the projection of the feature dependent on the initally extracted information.

field sequence Style guides define not only the way of field separation. The sequence and alignment of the fields normally does not change within a reference section. Although some fields are optional and may be skipped by the author, information about the occurring sequences can resolve ambiguities and be of assistance in classification. As a simple example, we refer to the date field of the reference. Normally, the date is located either directly after the author or near the end of the reference. If no features indicate a date in the current reference, then information about the field before and the field after the dates of the remaining references hepls to find the date. For every label type $\varphi_{\text{label}} \in \mathbb{T}_{\text{label}}$ we try to detect fields that are normally located before and after the fields with the label φ_{label}. These additional features $\text{BEFORE}_{\varphi_{\text{label}}}$ and $\text{AFTER}_{\varphi_{\text{label}}} \in \mathbb{T}_{\text{meta-feat}}$ then indicate the inherent sequences of the reference section for the meta phase. For instance, if the analysis of the extracted information is confident that the field date is always followed by the pages field, then we assign an annotation of the type $\text{AFTER}_{\text{DATE}} \in \mathbb{T}_{\text{meta-feat}}$ to each token that was labeled with type φ_{PAGES}.

Summarizing, annotations of the types $t \in \mathbb{T}_{\text{meta-feat}}$ are used to enrich the feature functions of the meta phase. In the following, we describe how these types are determined with the use of the given annotation-based features and the initially extracted information.

First, an observed meta feature is created for the selection of types that are suitable for a meta feature.

Definition 5 (Observed Meta Feature)**.** *$\varphi^*_{meta} \in \mathbb{T}_{meta\text{-}observed}$ is defined as the manifestation of the corresponding meta feature $\varphi_{meta} \in \mathbb{T}_{meta\text{-}feat}$ in the case that the information was extracted perfectly.*

In other words, the observed meta feature $\varphi^*_{\text{meta}} \in \mathbb{T}_{\text{meta-observed}}$ is automatically assigned to those tokens of each reference that are located exact at the positions indicated by the meta feature $\varphi_{\text{meta}} \in \mathbb{T}_{\text{meta-feat}}$. The observed meta features $\text{BEGIN}^*_{\text{AUTHOR}}$, for example, is assigned to the first token of the initially extracted author field and the observed meta features $\text{END}^*_{\text{AUTHOR}}$ is assigned to the last token of the author field.

The observed meta features will only be utilized to determine suitable types for the meta features. All available types of features $\varphi \in \mathbb{T}_{\text{feat}}$ are compared to the observed meta features φ^*_{meta} using a similarity measure. It is useful to consider the shape of the tokens and their properties to be the outcome of a stochastic event. From this point of view, f_φ (cf. section 2.2) is a random variable and $p(f_\varphi{=}1)$ represents the probability that a token has the feature φ. Additionally, $p(f_{\varphi_1}{=}1, f_{\varphi_2}{=}1)$ is a joint probability, indicating how likely a token has both the feature φ_1 and the feature φ_2. By example, $p(f_{\text{NUM}}{=}1, f_{\text{CW}}{=}1) = 0$ since tokens cover either numbers or capitalized words.

The mutual information has shown to be a sound similarity measure. Between two random variables X and Y

$$\text{MI}(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \cdot \log \left(\frac{p(x,y)}{p_X(x) \cdot p_Y(y)} \right)$$

measures how much information X and Y share. It covers all possible outcomes of X and Y. However, since we are only interested in the coincidence of a feature φ and the observed meta features φ^*_{meta} and not, for example, in the absence of a feature, the sum of all different values or occurrences of the features is removed. Hence, the mutual information can be reduced to a weighted pointwise mutual information. The probability distributions are then estimated by the observed frequencies $\hat{p}$ in the reference section:

$$\alpha(\varphi, \varphi^*_{\text{meta}}) = \hat{p}(f_\varphi{=}1, f_{\varphi^*_{\text{meta}}}{=}1) \cdot \log \frac{\hat{p}(f_\varphi{=}1, f_{\varphi^*_{\text{meta}}}{=}1)}{\hat{p}(f_\varphi{=}1) \cdot \hat{p}(f_{\varphi^*_{\text{meta}}}{=}1)}$$

High values of $\alpha(\varphi, \varphi^*_{\text{meta}})$ indicate that the type φ is suitable to describe the meta feature φ_{meta}. The weighted pointwise mutual information is motivated with the fact the rare occurrences of an information and a feature aren't representative for the complete reference section.

After applying the formula on all available features, we gain a sorted list of rated candidates for each meta feature. For the presented work no conjunctions of features for the manifestation of a meta feature are utilized. However, a meta feature cannot always be described by a single feature, but requires sometimes a disjunction of features. Therefore, instead of only using the highest rated feature for the description of the meta feature, each feature is consulted that fulfills two conditions: its α rating exceeds a given threshold β and the annotations of the feature are disjoint to the other selected features whereas higher rated features are preferred. On the one hand, some rare applied style guides are able to create different separators for a field. But also with a strict style guide applied, the absence of some information can require a description of a meta feature with several features. If the date contains an information about the month in fifty percent of its occurrences, then the start separator of the date is either a number or a word indicating a month name. Hence, the description of the begin of the date would be described best with two features. For that reason, several feature are allowed for the manifestation, but only if they are not redundant, i.e. are disjoint to the already selected features of higher rating. For the computation of disjoint features the joint probability with the observed frequencies is reused. Two features φ_1 and φ_2 are considered disjoint iff $\hat{p}(f_{\varphi_1} = 1, f_{\varphi_2} = 1) \approx 0$. A minimal margin was applied since we assume a fallible feature extraction that erroneously assigns a feature on rare occasions.

The threshold is applied because of the weighted pointwise mutual information. Features that only occur once or twice in a document are not confident enough for the description of the local model even if they are disjoint with the already selected features. This selection of disjoint features does not need to be applied for the sequences of the fields due to the characteristics of the reference parsing domain where the labels build a disjoint partition of the complete reference. Only the threshold is used to filter rare sequences of fields.

In order to use the meta features in the common process model, the types of annotations are projected by providing an annotation-based feature function. Additionally to

the already described feature function, a specialized feature function $f_{\varphi^t_{\text{meta}}}$ is added for the separation of the fields.

$$f_{\varphi^t_{\text{meta}}}(x_t) = \begin{cases} 1, & \text{if } x_t \text{ has the feature } \varphi_{\text{meta}} \\ & \text{and } |t-o| \text{ is minimal with} \\ & f_{\varphi^*_{\text{meta}}}(x_o) = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Here, a token possesses this meta feature only if it is located nearest to the observed meta feature φ^*_{meta}. The projection is of course limited to the currently considered reference. The combination of both strategies for the projection of separators enforces the reuse and simultaneously the correction of the initially extracted information.

The complete process of the creation of the local model and its projection as features is summarized in algorithm 1:

Algorithm 1 Local Model Phase

```
for all φ_meta ∈ T_meta-feat do
    l ← new list
    for all φ ∈ T_feat do
        if (∀φ' ∈ l : p̂(f_φ = 1, f_φ' = 1) ≈ 0) ∧
        α(φ, φ*_meta) > β then
            add φ to l
    project l as manifestation of φ_meta
```

2.4 Example

The selection of features and their projection as meta features are illustrated with a simplified example focusing on the meta feature $\text{END}_{\text{AUTHOR}}$. The input of the presented approach is a reference section with 20 references and overall 799 tokens that have been labeled in the bootstrap phase. Figure 2 contains two references of the reference section. The first row shows the begin of the reference, whereas the labels assigned by the bootstrap phase are depicted in the second row. Obviously, the CRF falsely labeled the tokens "Exokernel:" as author in the first reference. In the next four rows, a selection of features are added. *PM* stands for all punctuation marks, *PERIOD* for periods, *COLON* for colons and *PeriodSep* for periods that are not part of abbreviations or name initials. Finally, the last two rows contain the computed meta features $\text{END}_{\text{AUTHOR}}$ and $\text{END}^t_{\text{AUTHOR}}$. Applying the similarity measure on the given features results in a rating how good the feature describes the end of the author. The values are given for two suitable features:

$$\alpha(\varphi_{\text{PERIOD}}, \text{END}^*_{\text{AUTHOR}}) = \frac{18}{799} \cdot \log \frac{\frac{18}{799}}{\frac{143}{799} \cdot \frac{19}{799}} = 0.0375$$

$$\alpha(\varphi_{\text{PeriodSep}}, \text{END}^*_{\text{AUTHOR}}) = \frac{18}{799} \cdot \log \frac{\frac{18}{799}}{\frac{63}{799} \cdot \frac{19}{799}} = 0.0560$$

Since a token with the feature $\varphi_{\text{PeriodSep}}$ always has the feature φ_{PERIOD}, both features are not disjoint. The local model now states that the end of the author is at best described by a single feature. Hence, $\varphi_{\text{PeriodSep}}$ is assigned to $\text{END}_{\text{AUTHOR}}$ in this document and the feature function $f_{\text{END}_{\text{AUTHOR}}}(x_t) = 1$, iff x_t has the feature $\varphi_{\text{PeriodSep}}$. In a different reference section, for example, with another style guide applied $f_{\text{END}_{\text{AUTHOR}}}(x_t) = 1$, iff x_t has the feature φ_{COMMA} or φ_{COLON}. The meta feature $\text{END}^t_{\text{AUTHOR}}$ is consequently only assigned to the token that is nearest to the observed meta feature $\text{END}^*_{\text{AUTHOR}}$. That is the 20th token in the first reference and the 8th token in the second reference. The CRF of the meta phase now has access to additional features of high quality resulting in an increased accuracy.

3 Experimental Study

We have evaluated the presented process model, the idea of the local adaptivity and its novel combination of state of the art methods in an experimental study. First, the applied data sets, features and settings of the study are described. Then, the results of the evaluation are presented and discussed.

3.1 Data sets

The labeled data sets CORA (500 references), CITESEERX (200 references) and FLUX-CIM (300 references, CS domain)[4] build the source of the evaluation data set. All three data sets consist of a listing of single references without the context of the original reference section. Therefore, these data sets are not directly applicable for the presented approach. A simple script was developed in order to reconstruct reference sections as they would occur in real publications using only references originated in the available data sets. Due to the simplicity of the assignment script and the distribution of the reference styles in the dataset a considerable amount of references could not be assigned to a paper. The resulting data set D_{Paper} contains 28 documents and overall 452 references and resembles reference sections of real papers. Therefore, our data set can be considered more natural. Some erroneous labels and defects due to obvious differences in the annotation guide lines of the three original data sets were corrected. D_{Paper} is randomly splitted into three folds for the evaluation. D^{Train}_{Paper} (315 references, two folds) is used for the training and D^{Test}_{Paper} (137 references) for testing. Additionally, D_{Rest} contains 350 randomly selected references of the remaining references of the original data sets.

3.2 Features

Similar to previous studies with CRFs we use features indicating the capitalization, the length of tokens, numbers, whitespaces on the left and on the right of the observed token, the relative position inside the reference string, n-gram prefixes, n-gram suffixes, as well as the covered text of the token and the covered text of tokens on the left and on the right of the observed token. These features are integrated as normal feature functions. Additionally, annotation-based feature functions, previously denoted by $\mathbb{T}_{\text{feat}}$, for token classes and combinations of tokens are applied. To these belong different usages of punctuations, regular expressions for URLs and simple combinations of features, for example a first name and a capitalized word. Dictionaries for first names, stop words, locations, keywords, journals and publishers were created and added to the annotation-based feature functions. Overall, the applied features are comparable to previously published approaches.

3.3 Settings

Overall, three CRFs are trained for the experimental study: BOOTSTRAP, META and COMPARE. All of them were relying on the same features described in section 3.2. The overall process is build upon UIMA and the ClearTK framework. The machine learning toolkit Mallet is used for an implementation of the CRFs. The presented process model contains two CRFs. The CRF of the bootstrap phase (BOOTSTRAP) represents a simple model for the extraction task. It is trained on the data set D_{Rest} and 250 iteration

[4] all three data sets are available online, e.g., at `http://wing.comp.nus.edu.sg/parsCit/`

	D	.	R	.	Engler	,	M	.	F	.	Kaashoek	,	J	.	W	.	O	'	Toole	.	Exokernel	:	An	Operating	System	Architecture	for	Application	Level	...
Bootstrap result	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	T	T	T	T	T	T	T	T
$\text{END}^*_{\text{AUTHOR}}$																						x								
PM	x		x		x		x		x		x		x		x				x			x								
PERIOD	x		x				x		x				x		x															
COLON																						x								
PeriodSep																				x										
$\text{END}_{\text{AUTHOR}}$																				x										
$\text{END}^t_{\text{AUTHOR}}$																				x										

	S	.	Seneff	and	J	.	Polifroni	.	A	new	restaurant	guide	conversational	system	:	Issues	in	rapid	prototyping	for	specialized	domains	.	In	Pr...					
Bootstrap result	A	A	A	A	A	A	A	A	A	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	B	B		
$\text{END}^*_{\text{AUTHOR}}$									x																					
PM	x				x				x								x								x					
PERIOD	x				x				x																x					
COLON																	x													
PeriodSep									x																x					
$\text{END}_{\text{AUTHOR}}$									x																x					
$\text{END}^t_{\text{AUTHOR}}$									x																					

Figure 2: Two exemplary references with the initially extracted information, the some given features and the assigned meta features. The occurrence of a feature is indicated with "x" and the label of a token is denoted with first letter of the label.

were applied. The CRF of the meta phase (META) has access to the additional meta features $\mathbb{T}_{\text{meta-feat}}$. The model is trained on the data set D^{Train}_{Paper} with unlimited iterations and the threshold β for the meta features is set to 0.01. An additional CRF (COMPARE) is also trained on the data set D^{Train}_{Paper} with the settings of META in order to compare the increase of accuracy due to the meta features. A gaussian variance of 10 is used and the markov order is set to one for all CRFs. The two CRFs BOOTSTRAP and META are trained on two different data sets. The meta features need to be created on real results and not on the almost perfectly labeled data of a training process. If the meta features are only created for correct results in the meta phase, then the advantages for the contextual reuse and correction of the initially extracted information are forfeited. The training of the meta phase needs to rely on realistic and therefore not perfect results for a suitable integration of the meta features.

3.4 Performance Measure

The performance of the presented approach is measured with commonly used methods of the domain. For a field label $l \in \mathbb{T}_{\text{label}}$, let $\text{tp}(l)$ be the number of true positive classified tokens for the label l and define $\text{fn}(l)$ and $\text{fp}(l)$ respectively for false negatives and false positives. Since punctuations contain no information in this domain, only alpha-numeric tokens are considered.

Precision, *recall*, F_1 and *average* F_1 are computed by

$$\text{precision}(l) = \frac{\text{tp}(l)}{\text{tp}(l) + \text{fp}(l)},$$

$$\text{recall}(l) = \frac{\text{tp}(l)}{\text{tp}(l) + \text{fn}(l)},$$

$$F_1(l) = \frac{2 \cdot \text{precision}(l) \cdot \text{recall}(l)}{\text{precision}(l) + \text{recall}(l)},$$

$$\text{Average} = \frac{1}{|\mathbb{T}_{\text{label}}|} \sum_{l \in \mathbb{T}_{\text{label}}} F_1(l).$$

The *instance accuracy* measures how many references have been perfectly classified

$$\text{Instance} = \frac{\#\text{references without an error}}{\#\text{all references}}.$$

3.5 Results

Table 2 contains the results of the experimental study. The second column lists the true positives $\text{tp}(l)$ of each field $l \in \mathbb{T}_{\text{label}}$ and the remaining columns contain the F_1 scores of the three evaluated CRFs BOOTSTRAP, COMPARE and META tested on the data set D^{Test}_{Paper}. The average F_1 and the instance accuracy are added in the last two rows. As mentioned before, the amount of true positives is much smaller than the number of tokens since only alpha-numeric tokens are considered in the evaluation. The information of a date field, for example, is independent of surrounding parentheses or punctuation marks. There are no values added for the editor and note fields. Both fields consist only of a few tokens and achieved an F_1 score of 100.0 in most of the evaluation runs. However, a F_1 score of 0.0 was also sometimes obtained dependent on the distribution of examples in the two data sets D^{Train}_{Paper} and D^{Test}_{Paper}. Therefore, both fields are not considered in the calculation of the average F_1.

Table 1: Results of the evaluation of the three CRFs. The average F_1 is computed without the editor and note field.

	tp	BOOTSTRAP CRF	COMPARE CRF	META CRF
Author	821	99.0	99.1	**99.5**
Booktitle	670	94.8	95.1	**97.5**
Date	200	95.4	**98.0**	97.8
(Editor)	7	0/100	0/100	0/100
Institution	86	32.7	**97.1**	95.1
Journal	186	96.8	89.0	**98.1**
Location	51	86.4	91.7	**92.6**
(Note)	3	0/100	0/100	0/100
Pages	222	90.7	97.5	**97.7**
Publisher	33	87.5	**98.5**	93.7
Tech	75	37.0	87.4	**94.4**
Title	1064	97.0	96.6	**98.3**
Volume	84	98.8	85.1	**98.8**
Average*		83.3	94.1	**96.7**
Instance		75.9	78.8	**85.4**

The CRF META achieved an instance accuracy of 85.4% and an average F_1 score of 96.7%. Compared to the CRF BOOTSTRAP, the error of the instance accuracy was reduced by 39.4% and the error of the average F_1 by 80.1%. Compared to the CRF COMPARE that was trained on the same data set, the error of the instance accuracy was reduced by 31.1% and the error of the average F_1 by 44.1%.

A closer look at the single fields of META and COMPARE reveals that the meta phase was able to improve eight fields and worsened the results of only three fields. Two of these fields, the location and the publisher, contain less true positives than the other fields and strongly depend on dictionaries. The difference in the date fields is caused by only one single misclassified token. Overall, META created 3435 true positives, whereas COMPARE classified 3376 true positives.

3.6 Discussion

The combination of two CRFs and analysis of the local consistencies achieves better results than a single CRF, the state of the art method in the domain of reference extraction. Although the result of the bootstrap phase is mediocre, the local model and the projection of its knowledge are robust enough to create valuable meta features. Hence, the meta phase is able to outperform the commonly applied process. A closer look at the extraction results reveals that the presented approach still trails behind its own potential. The combination of features and meta features often create a situation where a correct classification is obvious. Many false positives and false negatives should not occur with the available features at hand. The boundaries of the author field, for example, are perfectly defined by the created separator features, but the CRF still labels some tokens of the author erroneously. Therefore, the presented approach still provides enormous potential for improvements. A different information extraction technique might integrate the knowledge about the local consistencies better in the meta phase than CRFs. Furthermore, a direct combination of both CRFs in the learning process or an improved projection of the meta features can improve our process model.

The effect of the presented approach on unknown style guides should be investigated in detail. The test data set D_{Paper}^{Test} already contains references with style guides that aren't present in the training data set D_{Paper}^{Train}. However, a test data set only containing unknown styles can illustrate the advantages of our approach compared to the common process model furthermore. A comparison to the results of related publications is problematic. Although the instances of the applied data set were used in previous evaluations, the results can hardly be compared as three different data sets were mixed and some references are left out.

4 Related Work

The extraction of references is an active field of research. Techniques based on Hidden Markov Models, Maximum Entropy Models and Support Vector Machines and several approaches using CRF were published. Peng and McCallum [Peng and McCallum, 2004] established CRF as the state of the art approach for the reference extraction task. They used 350 references of the CORA data sets for training and 150 references for the evaluation. Councill et al. [Councill *et al.*, 2008] applied CRF in their ParseCit system on the CORA data set and evaluated their approach with a 10 fold cross evaluation. In addition, they evaluated also the data sets CITESEERX and FLUX-CIM. Both approaches achieved an average F_1 score of ≈92% and a modified average F_1 score of ≈93%. Table 2 contains details of their evaluation results.

Ng [Ng, 2004] has built the first version of ParsCit. It was based on the Maximum Entropy paradigm and the accuracy was worse compared to the performance of the current system. However, Ng identified different categories of flaws in his extraction process and applied an additional phase for their correction. One step of these repairs processed repeating fields of a single reference, e.g., the occurrence of multiple titles. For the correction of this error, he created a list of all sequences of fields within the extracted references. Then, the multiple fields were resolved using the sequence that occurred most. With all repairs applied, the instance accuracy was increased from 45.6% to 60.8% on the CORA dataset. Compared to our approach, Ng applied only one specialized repair on the sequences in a post processing step in order to correct an error that can be prevented by applying a CRF instead. Furthermore, the evaluation with the CORA data sets itself prevents any statements about improvements by the usage of local consistencies.

Table 2: Results of related publications. In addition to the average F_1 score, the *Average** is computed without the editor and note fields.

[Peng and McCallum, 2004]	[Councill *et al.*, 2008]			
	CORA	CORA	CITESEERX	FLUX-CIM
Author	99.4	99	96	99
Booktitle	93.7	93	81	97
Date	98.9	99	94	97
Editor	87.7	86	67	-
Institution	94.0	89	74	-
Journal	91.3	91	83	89
Location	87.2	93	85	89
Note	80.8	65	29	-
Pages	98.6	98	91	97
Publisher	76.1	92	81	85
Tech	86.7	86	73	-
Title	98.3	97	93	96
Volume	97.8	96	87	92
Average	91.5	91.1	79.5	93.4
Average*	92.9	93.9	85.3	93.4
Instance	77.3	-	-	-

There are also some knowledge engineering approaches. Cortez et al. [Cortez *et al.*, 2007] evaluated their unsupervised lexicon-based approach on data sets of the domains health science and computer science. An automatically generated, domain-specific knowledge base is applied after a chunking of each text segment in order to identify the fields. Day et al. [Day *et al.*, 2007] created templates for well-known reference styles and used them to extract the fields in journal articles. They achieved an average accuracy of 92.4% on complete fields.

Kluegl et al. [Kluegl *et al.*, 2010] proposed a local adaptive extraction of references with handcrafted transformation rules. A simple model extracts initial fields. The local patterns of these information are stored in a short term memory and are used to create a description of the style guide of the reference section. Then, transformation rules match on a meta level and provide an automatic adaption on the internal previously unknown consistency of the document. The approach is evaluated only on the fields author, title, editor and date. They achieved an average F_1 score of 97.6% on the complete CORA data set and an average F_1 score of 99.7% on natural reference sections based on the CORA data set. This rule-based work on local adaptivity is the basis of our approach. We continued and extended the previous work by two major points: We exchanged the la-

borious knowledge engineering in all stages and combined an automatic creation of the local model with state of the art techniques. Only the definition of suitable features in the feature extraction requires manual effort of a knowledge engineer. Additionally, all 13 possible fields of a reference are extracted and evaluated instead of the four fields.

Our process model can be compared to stacked generalization [Wolpert, 1992] where a meta model is learnt using the output of initial, basic models. Sigletos et al. [Sigletos *et al.*, 2003] among others adapted this meta-learning approach of classification for the information extraction task. In contrast to this approach, we apply only a single base model and are not trying to compose a model directly on the initially extracted information. Instead, we are creating new features in order to exploit the patterns of the unlabeled data during the extraction task.

Recently, advances have been made in joint inference [Poon and Domingos, 2007] that combine different steps of the information extraction task. The presented work possesses some simple peculiarities of that approach, e.g., features transfer inference information in one direction. There is also work published on extending or parameterizing the features of discriminative models. Stewart et al. [Stewart *et al.*, 2008], for example, are learning flexible features for the extraction of references.

5 Conclusions

We have presented a novel combination of two CRFs applied on the local adaptive extraction of references. The initial results of the first CRF are exploited to gain information about the local consistencies. Then, the second CRF is automatically adapted to the previously unknown style guide. This is achieved by changing the manifestation of its features dependent on the currently processed reference section. The results indicate a considerable improvement towards the commonly applied process model. We achieved an average F_1 score of 96.7% and an instance accuracy of 85.4% on the test data set.

Several directions merit a further exploration of the presented work. A combined inference of both CRFs and a local model beyond a description by single features might exploit the full potential of the approach. Combinations and sequences of features are able to describe the local consistencies even if the features extraction provides only simple features. Moving further in this direction leads to a knowledge-based local model that is created, e.g., using subgroup discovery techniques. Comparing the presented approach to previous work [Kluegl *et al.*, 2010], a transformation-based correction of the initially extracted information appears to be very suitable for this task and is able to integrate the local model more straightforward. The knowledge engineering effort can be avoided by learning several binary classificators. Support Vector Machines, for example, can be trained to define transformations of the given information dependent on the local model. Our approach is not restricted to the extraction of references. Even greater improvements compared to the state of the art methods are possible in other domains like curriculum vitae and medical patient records (cf. [Kluegl *et al.*, 2009]).

References

[Cortez *et al.*, 2007] Eli Cortez, Altigran S. da Silva, Marcos André Gonçalves, Filipe Mesquita, and Edleno S. de Moura. FLUXCiM: Flexible Unsupervised Extraction of Citation Metadata. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 215–224, New York, USA, 2007.

[Councill *et al.*, 2008] Isaac Councill, C Lee Giles, and Min-Yen Kan. ParsCit: an Open-source CRF Reference String Parsing Package. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008. ELRA.

[Day *et al.*, 2007] Min-Yuh Day, Richard Tzong-Han Tsai, Cheng-Lung Sung, Chiu-Chen Hsieh, Cheng-Wei Lee, Shih-Hung Wu, Kun-Pin Wu, Chorng-Shyong Ong, and Wen-Lian Hsu. Reference Metadata Extraction using a Hierarchical Knowledge Representation Framework. *Decis. Support Syst.*, 43(1):152–167, 2007.

[Ferrucci and Lally, 2004] David Ferrucci and Adam Lally. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3/4):327–348, 2004.

[Kluegl *et al.*, 2009] Peter Kluegl, Martin Atzmueller, and Frank Puppe. Meta-level Information Extraction. In Bärbel Mertsching, Marcus Hund, and Muhammad Zaheer Aziz, editors, *KI*, volume 5803 of *Lecture Notes in Computer Science*, pages 233–240. Springer, 2009.

[Kluegl *et al.*, 2010] Peter Kluegl, Andreas Hotho, and Frank Puppe. Local Adaptive Extraction of References. In *33rd Annual German Conference on Artificial Intelligence (KI 2010)*. Springer, 2010. accepted.

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

[Ng, 2004] Yong Kiat Ng. Citation Parsing using Maximum Entropy and Repairs. Undergraduate thesis, National University of Singapore, 2004.

[Ogren *et al.*, 2008] Philip V. Ogren, Philipp G. Wetzler, and Steven Bethard. ClearTK: A UIMA toolkit for statistical natural language processing. In *UIMA for NLP Workshop at LREC*, 2008.

[Peng and McCallum, 2004] Fuchun Peng and Andrew McCallum. Accurate Information Extraction from Research Papers using Conditional Random Fields. In *HLT-NAACL*, pages 329–336, 2004.

[Poon and Domingos, 2007] Hoifung Poon and Pedro Domingos. Joint Inference in Information Extraction. In *AAAI'07: Proc. of the 22nd National Conference on Artificial intelligence*, pages 913–918. AAAI Press, 2007.

[Sigletos *et al.*, 2003] Georgios Sigletos, Georgios Paliouras, Constantine D. Spyropoulos, and Takis Stamatopoulos. Meta-Learning beyond Classification: A Framework for Information Extraction from the Web. In *Proc. of the Workshop on Adaptive Text Extraction and Mining. The 14th ECML and the 7th Euro. Conf. on PPKD*, 2003.

[Stewart *et al.*, 2008] Liam Stewart, Xuming He, and Richard S. Zemel. Learning Flexible Features for Conditional Random Fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1415–1426, 2008.

[Wolpert, 1992] David H. Wolpert. Stacked Generalization. *Neural Networks*, 5:241–259, 1992.