Parsing and Maintaining Bibliographic References
Semi-supervised Learning of Conditional Random Fields with Constraints

Sebastian Lindner and Winfried Höhn
Institute of Computer Science, University of Würzburg, Würzburg, Germany

Keywords: References Parsing, Bibliography, Conditional Random Fields (CRFs), Constraint-based Learning, Information Extraction, Information Retrieval, Machine Learning, Sequence Labeling, Semi-supervised Learning.

Abstract: This paper shows some key components of our workflow to cope with bibliographic information. We therefore compare several approaches for parsing bibliographic references using conditional random fields (CRFs). This paper concentrates on cases, where there are only few labeled training instances available. To get better labeling results prior knowledge about the bibliography domain is used in training CRFs using different constraint models. We show that our labeling approach is able to achieve comparable and even better results than other state of the art approaches. Afterwards we point out how for about half of our reference strings a correlation between journal title, volume and publishing year could be used to identify the correct journal even when we had ambiguous journal title abbreviations.

1 INTRODUCTION

In academic research it is good practice to compare your own work with previously published documents and acknowledge these in a reference section. In consequence of the increasing number of scientific publications, there is a growing demand for an easy searchability of these previous works. Therefore the automatic analysis of these publications and their bibliographic references is getting more and more important.

There already are a few search engines for this kind of data like Google Scholar[1] or CiteSeerX[2]. In order to search in single fields each reference has to be divided into a set of fields or labels (e.g. author or journal title). While the task of separating a reference string into different fields is simple for a human reader, it is much more difficult to automate, because of the sheer diversity of reference strings.

First approaches in this field of research tried rule based algorithms, but these were too expensive to maintain and too difficult to adjust to other reference domains. Because of that we use machine learning techniques, which are much more easily adaptable to other reference labeling domains (Zou et al., 2010). In supervised machine learning already labeled reference instances are used to train a statistical model, which then can be used to label further reference strings. Generating this labeled training data is a very time consuming job.

Our focus lies on conditional random fields that use additional prior knowledge in form of constraints about the bibliography domain in addition to a few already labeled instances for its training. In this scenario a few labeled training instances and additional unlabeled data for the training of constraints are used in a semi-supervised training to achieve better results. These constraints can easily be adapted for a new domain and the inclusion of valid constraints leads to a significant improvement in labeling accuracy.

In one of our projects with Springer Science+Business Media we develop the web platform SpringerMaterials[3] for an online presentation of published documents. These contain a large amount of bibliographic information. Because these documents are divided into different books there are many different citation styles. Due to that fact we can not use one model for labeling all data, but we have to split the reference data into smaller subsets and do a separate labeling for each of these sets with only a few training instances.

After separating the reference string into different fields, we demonstrate own approaches to find corresponding long versions for journal title abbreviations.

[1] http://scholar.google.com
[2] http://citeseerx.ist.psu.edu/index
[3] http://www.springermaterials.com

Lindner S. and Höhn W..
Parsing and Maintaining Bibliographic References - Semi-supervised Learning of Conditional Random Fields with Constraints.
DOI: 10.5220/0004138602330238
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2012), pages 233-238
ISBN: 978-989-8565-29-7

The goal of this part of this paper is to reach a uniform representation of reference string.

So we propose a workflow for dealing with bibliographic information that combines different approaches to analyze and clean bibliographic data.

2 REFERENCE PARSING

First of all we describe the problem to be solved. In the first step the reference string needs to be broken down into tokens (i.e. split on whitespace). Each token in this sequence of input tokens $\mathbf{x} = \{x_1, x_2, \ldots x_n\}$ then needs to be assigned a correct output label out of a set of labels $\mathbf{y} = \{y_1, y_2, \ldots y_n\}$. For example the label *AUTHOR* has to be assigned to the name *Meier*.

The problem of assigning a label to a token of an input sequence also is a common task in the research area of natural language processing, for example in part-of-speech tagging and semantic role labeling. So the methods shown here can similarly be used in a variety of other research areas as well (Park et al., 2012).

2.1 CRFs with Extended Generalized Expectation Criteria

Linear-chain conditional random fields (CRFs) are a probabilistic framework that use a discriminative probabilistic model over an input sequence **x** and an output label sequence **y** as shown in equation 1.

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\left(\sum_i \lambda_i F_i(\mathbf{x},\mathbf{y})\right), \qquad (1)$$

where in case of a linear chain CRF $F_i(\mathbf{x},\mathbf{y})$ are a number of feature functions and $Z(\mathbf{x})$ is a normalization factor as described in (Sutton and McCallum, 2006). CRFs outperform Hidden Markov Models (HMM) and maximum entropy Markov models (MEMM) on a number of real world tasks (Lafferty et al., 2001).

In the training process the parameters λ_i for each feature function are learned from the training data. One approach for semi-supervised learning are CRFs with Generalized Expectations (GE) as described in (Mann and McCallum, 2010). For our experiments we use MALLET (McCallum, 2002) which implements the CRF with Generalized Expectations. Generalized Expectation Criteria use prior knowledge in form of a user provided conditional probability distribution

$$\hat{p_\lambda} = p_\lambda(y_k | f_i(\mathbf{x},k) = 1) \qquad (2)$$

given a feature f_i. For example the probability for a label *AUTHOR* of the word *Meier* should be 0.9. So GE constraints express a preference for a specified label given a certain feature.

We extended this version to allow more complex constraints, though. In the original version only one feature function is allowed for the target probability distribution (compare equations 2 and 4). This way more complex constraints can be used which depend on multiple feature functions so that we could improve our labeling results.

Given a set of training data $T = \left\{\left(\mathbf{x}^{(1)},\mathbf{y}^{(1)}\right),\ldots,\left(\mathbf{x}^{(m)},\mathbf{y}^{(m)}\right)\right\}$ the goal of training a conditional random field is to maximize equation 3 i.e. the log-likelihood (first term) with a Gaussian prior for regularization (second term) and a term to take constraints into account.

$$\Theta(\lambda, T, U) = \sum_i log p_\lambda\left(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}\right) - \frac{\sum_i \lambda_i^2}{2\sigma^2} - \delta D(q||\hat{p_\lambda}), \qquad (3)$$

where q is a given target distribution and

$$\hat{p_\lambda} = p_\lambda(y_k | f_1(\mathbf{x},k) = 1, \ldots, f_m(\mathbf{x},k) = 1) \qquad (4)$$

with all $f_i(\mathbf{x},k)$ being feature functions that only depend on the input sequence. $D(q||\hat{p_\lambda})$ is a distance function for the two provided probability distributions. In our case the Kullback-Leibler (KL) and L_2 distance are used and compared against each other. The value δ is used to weight the divergence between the two distributions. This way the expectations encourage the model to penalize differences in the distributions (Lafferty et al., 2001). In case of the L_2-distance a range-based version is used where a valid probability range can be specified. If the compared value is within this range the distance is 0.

2.2 CRF Features for Reference Parsing

For the task of tagging reference strings, we used a set of binary feature functions similar to the ones used by ParsCit (Councill et al., 2008). These can be divided into the following four categories.

Word based Features. These features indicate the presence of some significant predefined words *'No.','et al', 'etal', 'ed.'* and *'eds.'*.

Dictionary based Features. These features indicate whether a dictionary contains a certain word in the reference string. We use dictionaries for author first- and lastnames, months, locations, stop words, conjunctions and publishers.

Regular Expression based Features. Features that indicate whether a word in the reference string matches a regular expression. We use regular expression patterns for ordinals (e.g. 1st, 2nd...), years,

paginations (e.g. 200-215), initials (e.g. J.F.) and patterns that indicate whether a word contains a hyphen, ends with punctuation, contains only digits, digits or letters, has leading/trailing quotes or brackets or if the first char is upper case.

Keyword Extraction based Features. We extract keywords for a particular label in a separate step, so that we can reuse this information when we define constraints for the CRF. The corresponding training feature then indicates whether a word is in an automatically extracted list of keywords for a certain label.

We used the *GSS* measure mentioned in the paper about automated text categorization (Sebastiani, 2002) for the purpose of automatic keyword extraction. *GSS* is thereby defined as

$$GSS(t_k, c_i) = p(t_k, c_i) \cdot p(\overline{t_k}, \overline{c_i}) - p(t_k, \overline{c_i}) \cdot p(\overline{t_k}, c_i), \quad (5)$$

where $p(\overline{t_k}, c_i)$ for example is the probability that given a label c_i, the word t_k does not occur under this label.

The top *GSS* scored words and their corresponding labels are then reviewed and used in a keyword feature function and in our constraint set. Words with less than three characters and stop words are specifically excluded from the suggested keyword list.

Besides the already mentioned features we also use a window feature with a size of 1 for training our CRF. We also use a feature that indicates the position of the word in the reference string itself. Therefore we divide the reference string into six parts and give each word a position number feature whose value indicates its position from 1 to 6.

2.3 CRF Constraints for Reference Parsing

Since we use Conditional Random Fields with extended Generalized Expectations for the reference labeling task, as shown in equation 3, we can choose different distance metrics. We compare the KL-divergence, where a full feature-label target probability distribution q has to be specified, and the L_2-Range based distance.

We use the following constraints that depend on a single feature function:

1. Words contained in the previously mentioned dictionaries in section 2.2 should be tagged with the corresponding label for that dictionary (exceptions: conjunctions, stop words, first and last names)
2. Extracted keywords for a label should be tagged with that same label
3. Words that match a year pattern should be labeled with *DATE*
4. Words that match a predefined word should be labeled with the corresponding label (see word based features in section 2.2)

We encode these constraints into the model by providing distributions q in the following kind:

For constraints 1.-3. we set the desired label probability to 0.8 and equally distribute the 0.2 among the rest of the labels in the case of the KL-divergence as the distance metric. When we use a L_2-Range specification we define a target probability range from 0.8 to 1.0. We do this because '1992' for example could be a page number. For constraint 4. the specified target label has the probability 1.0.

We also use some more complex constraints that take usage of the possibility to specify constraints over multiple feature functions. These are:

1. Words that appear at the beginning of the reference string and are contained in the dictionary of first or last names should be labeled *AUTHOR* (for the middle and ending of the reference string *EDITOR*)
2. Conjunctions that appear at the beginning of the reference string and are between words contained in the dictionary of first or last names should be labeled *AUTHOR* (for the middle and ending of the reference string *EDITOR*)
3. A number right to the word 'No.' should be labeled *VOLUME* as the word 'No.' itself
4. A year number right or left to a name of a month should be labeled *DATE* as the month itself

We use the same target probability distribution as in case of the constraints 1.-3. that use only one feature function. These more complex constraints show that with our extension it is easy to define constraints for CRFs with GE that take use of multiple features.

2.4 Experiments

As our test domain we used the Cora reference extraction task (McCallum et al., 2000). This set of citations contains 500 labeled reference strings of computer science research papers. These citations contain the 13 labels: *AUTHOR*, *BOOKTITLE*, *DATE*, *EDITOR*, *INSTITUTION*, *JOURNAL*, *LOCATION*, *NOTE*, *PAGES*, *PUBLISHER*, *TECH*, *TITLE*, *VOLUME* . We use this dataset to be able to compare own labeling results with previous approaches.

2.4.1 Preparations

In order to get good labeling results for only a few

training instances our first step was to clean up the dictionaries we had available. Therefore we first removed entries in the first name, last name and location dictionary that only contained two letters or less. Then we made our dictionaries pairwise disjoint by removing entries that are contained in more than one dictionary.

Then we used the previously described keyword extraction mechanism on several labeled citation sources to gather important words. A brief excerpt of these extracted keywords is shown in the following list of labels with their extracted keywords:

BOOKTITLE: *'Proceedings'*, *'Conference'*, *'Symposium'*, *'International'*, *'Programming'* - PAGES: *'pages'*, *'pp'* - PUBLISHER: *'Press'* - JOURNAL: *'ACM'*

The list shows that *GSS* is able to extract several useful keywords for different labels. So these not only reduce the manual effort to get good labeling results, but also ensure a better adaptability to other labeling domains.

2.4.2 Labeling Results

We used the same test approach as described in (Chang et al., 2007). Therefore we randomly split the 500 reference instances into three sets of 300/100/100 reference strings. We use the 300 as training, 100 as development and 100 as testing set. From the set of 300 instances we randomly choose our reference strings for training with varying training set sizes from 5 to 300. The 100 test instances are used in the evaluation process. In the semi-supervised settings we also use 1000 instances of unlabeled data which we mostly took from FLUX-CiM and CiteSeerX databases which are available on the ParsCit[4] website.

The results are shown in Table 1. The column **Sup** contains the results for a CRF with no constraints that uses the same features as our approaches with constraints. The results reported below are the averages over 5 runs with random training sets with corresponding size. The column **GE-KL** contains the data for our Generalized Expectation with Multiple Feature approach using the KL-divergence and last column uses the L_2-Range distance metric **GE-L_2-Range**. In this table we report token based accuracy i.e. the percentage of correct labels.

We compare our method against other state of the art semi-supervised approaches like the constraint-driven learning framework in column **CODL** (Chang et al., 2007), which iteratively uses the top-k-inference results in a next learning step. We also

[4]http://aye.comp.nus.edu.sg/parsCit/

Table 1: Comparison of token based accuracy for different supervised/ semi-supervised training models for a varying number of labeled training examples N. Results are an average over 5 runs in percent.

N	Sup	PR	CODL	GE-KL	GE-L_2-Range
5	69.0	75.6	76.0	74.6	75.4
10	73.8	-	83.4	81.2	83.3
20	80.1	85.4	86.1	85.1	86.1
25	84.2	-	87.4	87.2	88.4
50	87.5	-	-	89.0	90.5
300	93.3	94.8	93.6	93.9	94.1

compare our results with the results from CRF with Posterior Regularization(Bellare et al., 2009). In Posterior Regularization (column **PR**) the E-Step in the expectation maximization algorithm is modified in such a way that it also takes a KL-divergence into account(Ganchev et al., 2010). Dashes indicate that no comparison data was available in the referenced papers.

As one can see our method has about the same performance as other leading semi-supervised training methods. With a very limited amount of training data the results are slightly worse than the other approaches but with more and more training instances it outperforms most of the other techniques (e.g. with $N = 25$). The provided constraints improve the labeling results in comparison to a CRF without constraints (column **Sup**). For $N = 20$ the improvement for **GE-L_2-Range** is 6 percentage points in token accuracy. Our experiments show that we get the best performances using GE constraints with multiple feature functions and $L_2 - Range$ as distance metric.

The results also show that with an increasing number of training instances N, the positive influence of constraints decreases. The traditional CRF is then better able to extract the significance of features for a label by itself. We supposedly did not get the best labeling result for $N = 300$ in comparison to **PR** because we used more complex constraints. These did not have such a big influence on the labeling results with a high number of training reference strings.

Table 2 shows precision, recall and the F_1 measure for the label accuracy with 15 training instances using GE with multiple feature functions and L_2-Range as distance metric.

As we can see there is a big difference in the F_1 value for different labels. Because of the defined constraints *AUTHOR* and *DATE* labels have a high precision. *NOTE* labels are hard to identify and do not occur very often in the training material. This results in a rather poor labeling performance for *NOTE* labels. Because some labels like authors are much more common in reference strings, it is important to get a good performance for such labels.

Table 2: Precision, recall and F1 measure for label accuracy with $N = 15$ for a CRF with GE-L_2-Range.

Label	Precision	Recall	F_1
AUTHOR	98.5	98.6	98.6
DATE	95.0	82.5	88.3
EDITOR	92.3	52.8	67.2
TITLE	85.5	98.2	91.4
BOOKTITLE	84.3	84.1	84.2
PAGES	82.6	90.0	86.1
VOLUME	75.8	73.5	74.6
PUBLISHER	73.7	35.0	47.5
JOURNAL	71.9	66.6	69.1
TECH	67.5	25.7	37.2
INSTITUTION	62.4	43.4	51.2
LOCATION	51.4	60.0	55.4
NOTE	15.6	10.0	12.2

We also have to mention the fact that especially in training runs with only few reference strings the results might have a high standard deviation, because of the diversity of the training material.

3 POSTPROCESSING

After our reference strings have been tagged, the reference data is split into fields like journal, author, title and publishing year. But there could be different abbreviations and long versions for one journal name: *'Japan. J. Appl. Phys.'*, *'Jpn. J. Appl. Phys.'* and *'Trans Japan Inst Metals'*, *'Trans Jpn Inst Mct'*. Journal title entries can also contain OCR errors like in the second example.

When you search for a journal title or an author name you want to get all occurrences of this particular entity, regardless of the exact term or abbreviation used. Therefore it is important to collect all different notations of this entity in order to trigger a search for all these alternatives.

3.1 Journal Identification

Our first step is to use a string-based clustering for the journal titles. This can however introduce conflicts, e.g. when different journals have the same names or abbreviations, but actually refer to different journals. In this section we provide a method to resolve these ambiguities.

For example the journal abbreviation *'Phys. Rev.'* for *'Physical Review'* is sometimes also used for *'Physical Review Letters'* or *'Physical Review B'*. In figure 1 you can see a plot for the volume-year combinations of reference strings, which have *'Phys. Rev.'* as journal name. Although the reference data for the two lines in this plot have the same journal abbreviation, they actually refer to two different journals with

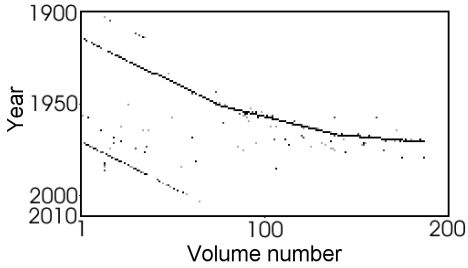

Figure 1: Volume-year combinations for *'Phys. Rev.'*.

different release schedules.

Therefore our disambiguation approach uses the relationship between the fields journal name, volume number and publishing date to separate same journal title abbreviations. Since most journals are published on a regular basis, there is a linear dependency between the volume number and publishing date for a journal. In case of changes in the release schedule we at least have line segments, which stand for periods while the release schedules stay the same (see Figure 1 upper line).To detect these lines we use the Hough transform (Duda and Hart, 1972).

This line detection is done for each of the string-based clusters. After that we select the line which contains the most data points. For this line we select the most common journal title in the cluster as our representative title. Next we determine the corresponding start and end volume by selecting the longest line segment which has only smaller gaps than three years.

Because the extracted line segment may contain data points from other journals (e.g. a crossing line segment) we have to remove these data points from our considered line segment. Therefore we take each unique journal title from our cluster and remove all of its data points if less than 25% in the year/volume-range of the considered line segment lie on the line segment itself. All journal titles that correspond to remaining data points are extracted as synonyms for that journal.

After that all references which match the found characteristics are removed and the procedure is repeated until we are unable to find further schedule ranges which are longer than three years.

3.2 Journal Identification Examples

We were able to extract 49 release schedules from our data, which covers 48.4% of the 248407 input reference strings. Table 3 shows a few of this extracted journal release schedules. Here example 1. and 3. are correctly separated into different release schedules although they often had the same abbreviation *'Phys.*

Table 3: Results of postprocessing.

No.	Journal	start		end		months between
		month	vol.	month	vol.	consecutive volumes
1.	Phys. Rev. B	Jan. 1970	1	Jan. 2009	79	6
2.	J. Am. Chem. Soc.	Jan. 1900	22	Jan. 2008	130	12
3.	Phys. Rev. Lett.	Jul. 1958	1	Jul. 2008	101	6
4.	J. Appl. Phys.	Jan. 1941	12	Jan. 1984	55	12
		Jul. 1984	56	Jan. 2006	99	6

Rev.'.

Since reference strings most of the time only contain a publishing year but no month, this method can just detect changes in the release schedule within this accuracy. Likewise we are unable to detect changes in the release schedule that are shorter than 3 years because we used this accuracy for our line segment detection. We also have to note that in order for this procedure to work a large data set should be used where single journal titles appear several times.

4 CONCLUSIONS AND FUTURE WORK

We proposed a new method of parsing references with constraints that can easily be adapted to new domains of data. The labeling results show that the proposed method's performance is comparable to or even performs better than other state of the art semi-supervised machine learning algorithms. Afterwards we demonstrated how bibliographic references can be clustered using the novel approach of analyzing a journal title, publishing month and release time span correlation.

We want to concentrate future effort in the automatic extraction of features and using constraints in the inference process in addition to the learning phase. Although we used a method for the automatic extraction of keywords for learning, we would like to integrate data from other web knowledge bases. We would also like to investigate the possibility to automatically categorize citation data and then use the optimal corresponding CRF for its labeling. Additionally we are going to improve our string-based clustering methods since typical Levenshtein-distance based metrics do not work well with abbreviations.

REFERENCES

Bellare, K., Druck, G., and McCallum, A. (2009). Alternating projections for learning with expectation constraints. *In Proceedings of UAI*.

Chang, M.-W., Ratinov, L., and Roth, D. (2007). Guiding semi-supervision with constraint-driven learning. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 280–287.

Councill, I. G., Giles, C. L., and Kan, M.-Y. (2008). Parscit: An open-source crf reference string parsing package. In *International Language Resources and Evaluation*. European Language Resources Association.

Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15.

Ganchev, K., Graa, J., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probablistic models for segmenting and labeling sequence data. *In Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*.

Mann, G. S. and McCallum, A. (2010). Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11:955–984.

McCallum, A. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

McCallum, A., Nigam, K., Rennie, J., and Seymore, K. (2000). Automating the contruction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163.

Park, S. H., Ehrich, R. W., and Fox, E. A. (2012). A hybrid two-stage approach for discipline-independent canonical representation extraction from references. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, JCDL '12, pages 285–294, New York, NY, USA. ACM.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Sutton, C. and McCallum, A. (2006). *Introduction to Conditional Random Fields for Relational Learning*. MIT Press.

Zou, J., Le, D., and Thoma, G. R. (2010). Locating and parsing bibliographic references in html medical articles. *International Journal on Document Analysis and Recognition*, 2:107–119.