Bibliographic Meta-Data Extraction Using Probabilistic Finite State Transducers

Martin Krämer[1], Hagen Kaprykowsky[1], Daniel Keysers[1], Thomas Breuel[2]
[1]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
[2]Technical University of Kaiserslautern, Germany
{martin.kraemer, hagen.kaprykowsky, daniel.keysers}@dfki.de, tmb@informatik.uni-kl.de

Abstract

We present the application of probabilistic finite state transducers to the task of bibliographic meta-data extraction from scientific references. By using the transducer approach, which is often applied successfully in computational linguistics, we obtain a trainable and modular framework. This results in simplicity, flexibility, and easy adaptability to changing requirements. An evaluation on the Cora dataset that serves as a common benchmark for accuracy measurements yields a word accuracy of 88.5%, a field accuracy of 82.6%, and an instance accuracy of 42.7%. Based on a comparison to other published results, we conclude that our system performs second best on the given data set using a conceptually simple approach and implementation.

Input: (plain text)
Davenport, T., D. DeLong and M. Beers, "Successful knowledge management projects," Sloan management review, 39, 2, (1998), 43–57.

Output: (BibTeX)
author = "Davenport, T. and DeLong, D. and Beers, M."
title = "Successful knowledge management projects"
journal = "Sloan management review"
volume = "39"
number = "2"
year = "1998"
pages = "43–57"

Figure 1. BibTeX output of the system for a given plain-text reference.

1 Motivation

Research paper search engines like CiteSeer [4, 5, 9] have become important because they enhance researchers' efficiency due to a faster access to current resources and the possibility for quicker distribution of new results. Furthermore such systems depend on the ability to assign the highly differing syntactical descriptions of the same paper to one semantical entity to provide a sound database. As search results directly depend on the quality of the information extraction component it is a part of the system with a high degree of significance. The task of bibliographic reference recognition describes the process of labeling the different parts of a given string according to their semantic meaning. In our case it handles the extraction of bibliographic metadata as BibTeX subfields from given plain-text research paper references as illustrated in Figure 1. Although the problem seems to be not that intricate at the first glance, a closer examination discloses a multitude of complications. Basically a bibliographic reference can be defined as an arbitrary series of subfields wherein each transition between subfields occurs upon parsing a specific separator. Across different reference styles we can observe dramatic variations amongst spacing, subfield order, partitioning symbols and content representation.

By designing a system that is based on training data we remove the need for a domain expert who manually analyzes the different BibTeX styles and derives adequate rules from them. This reduces the effort needed for defining the rules and prevents errors during the process – thus resulting in a highly efficient system. Furthermore we may adjust the language model to new reference styles by repeating the training procedure on a new dataset fitting those styles. Therefore only minimal manual intervention is needed as the whole process is highly automated. This is not possible in rule-based systems as another analysis of the new styles has to be performed and appropriate rules have to be derived. Hence our system works with a high degree of flexibility and adaptability in regard to changing bibliographic reference styles.

2 Related Work

Previous work on the topic of bibliographic meta-data extraction from research paper references can be subdivided into machine learning (ML) or rule-based [1, 2, 3, 13] approaches. Approaches based on ML try to derive the relationship between input and output strings according to a given set of samples and label future inputs using that knowledge. For the latter case a set of adequate rules has to be derived manually by a domain expert via analyzing appropriate samples. Major benefits of systems based on ML are their high degree of adaptability and robustness with the drawback of required training whereas rule-based systems usually behave more rigidly and do not adapt very well. The wide assortment of applied machine learning techniques spans conditional random fields [8, 14], hidden Markov models [10, 16], support vector machines [6] and statistical models [17].

3 System Design – Language Model

As the model of choice we selected probabilistic finite state transducers (PFSTs) which are basically trainable finite state machines with transition outputs, weights and scoring mechanisms according to the laws of probability. They seem a natural choice since finite state transducer (FST) techniques have been successfully applied to various tasks of natural language processing such as dictionary encoding, text processing and speech processing due to their ease of use and close relationship to regular languages. The prevalent benefits accompanying the usage of PFSTs are their high efficiency, leading to very fast systems, which increases their practical usability as well as the possibility for fast composition of complex models via abstract operations. Furthermore the visualization of transducers is comprehensible without difficulty due to the possibility of illustrating them as a directed graph. A thorough theoretical introduction of the model itself is given in [11, 15, 18].

The various subfields of a bibliographic reference are independent in the sense that their content stands in no relation to the content of other subfields. This allows to manually model each subfield on its own and link them together adequately using the knowledge of the occurring separator symbols and the class of the previous subfield. By using FST operations we can build the final language model in a modularized way. For our implementation we used the FST toolkit available from the Massachusetts Institute of Technology (MIT-FST[1]). A brief treatment of its design and implementation is given in [7] whereas in [12] the related FST library developed by AT&T is treated in detail. The toolkit is composed out of a large number of command-line tools where each one encapsulates a single functionality and works using the UNIX pipe. This allows for rapid development of models via command chains or shell scripts.

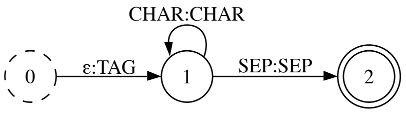

Figure 2. The proposed subfield model outputs a field tag, parses an arbitrary number of characters and stops parsing upon reading an interfield separator.

Particularly the ability to derive a probabilistically weighted transducer model via expectation maximization training from an unweighted transducer based on a set of pairs of input and output strings allows for efficient development of language models. The model's weights are iteratively adjusted in the process such that the likelihood of the training set's labeling are maximized. We have specified an FST for each possibly occurring subfield – namely author, booktitle, date, editor, institution, journal, location, note, pages, publisher, tech, title and volume. Inclusions of new subfields or exclusion of existing ones are achievable due to the modularized structure, i.e. they get included or excluded in the union operation over subfields. Each subfield model is solely represented by the intrafield unigram and separator symbol weights as they are structurally identical. These transducers are structured straightforward as they only output the corresponding subfield tag, then allow the parse of an arbitrary number of unspecified symbols via an intrafield unigram and terminates upon reading a separator symbol as shown in Figure 2. TAG represents the type of subfield (e.g. author), CHAR consists of the whole alphabet and SEP is a subset of CHAR specifying all possible interfield separator symbols like a colon or a dot. This allows parsing of intrafield separator symbols as we cannot exclude the possibility of them occurring there – e.g. a dot that indicates an abbreviation.

Now the overall language model is built as the union of the different subfields where we add epsilon transitions leading from each final state to the starting states of the subfield transducers. Thus we basically model a bigram of the subfields as it is shown in a simplified (restricted to only three subfields instead of the original 13 for enhanced readability) form in Figure 3. Consequently the parse of a subfield transducer depends on the type of the last subfield and the interfield separator symbol. The application of expecta-

[1] http://people.csail.mit.edu/ilh/fst

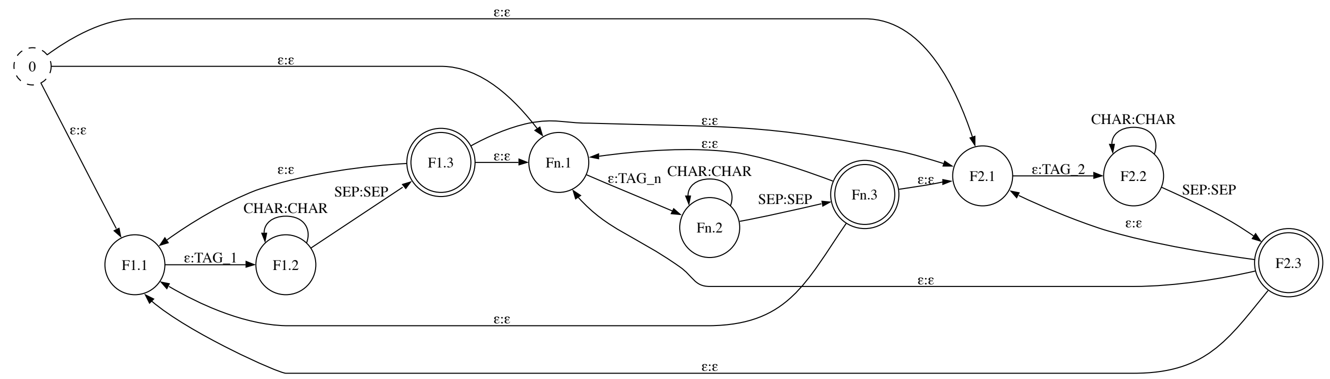

Figure 3. The final language model is generated from a union of subfield transducers with added epsilon transitions. A bigram of subfields is the result.

tion maximization training to the language model described above using the training data yields our final probabilistically weighted language model.

4 System Design – Overall Architecture

First our system receives a bibliographic reference given as plain text for input and passes it to the normalization component. Unknown symbols are removed as they would interfere with the parsing process and reserved symbols (e.g. 'ϵ' is denoted as ',' in MIT-FST) are mapped to another representation. By composing the resulting transducer with our language model we receive a tagging of subfields according to the highest probability by outputting the best parse. As the labeled reference is still in normalized format we need to revert the steps described above, i.e. restore the original meaning of symbols and rejoin them. This allows us to interpret each tagged sequence of the whole reference as a single BIBTEX subfield and their combination yields the desired BIBTEX reference as a result. BIBTEX is used commonly in the scientific community as it is highly configurable and able to handle huge bibliographies and documents without problems. Due to its widespread application to document management systems like digital libraries (CiteSeer, DBLP, CiteULike, etc.) we chose it as the output format to allow for an easy interoperability. The basic system work-flow is illustrated in Figure 4.

Our system is available online[2] via a web-based PHP front-end for demonstration purposes. A bibliographic reference can be entered into a multi-line HTML text area to pass it to the underlying system for classification and the result is displayed as a BIBTEX entry on a web-page. In the current state all entries are classified as '@misc' as no mechanisms for distinguishing between the reference types have been incorporated into the system yet.

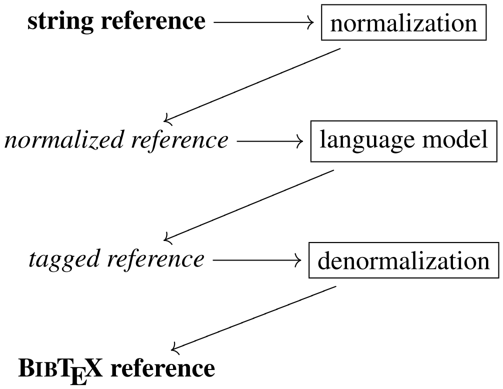

Figure 4. Visualization of basic system work-flow: first the plain-text reference is stripped from all unknown symbols in the normalization component. The transducer representing the resulting string is composed with our language model which yields a labeled reference according to the best parse. Finally the denormalization component outputs the result in plain-text again.

5 Performance Evaluation

We use the Cora[3] dataset for training and evaluation purposes due to its public availability and relatively widespread

[2] http://demo.iupr.org/refrec

[3] http://www.cs.umass.edu/∼mccallum/data/cora-ie.tar.gz

<author>Kambhampati, S., Knoblock, C., & Yang Q.</author> <date>(1995).</date> <title>Planning as refinement search: a unified framework for evaluating design tradeoffs in partial-order planning.</title> <journal>Artificial Intelligence,</journal> <volume>76,</volume> <pages>167-238.</pages>

<author>J. M. Ponte and W. B. Croft,</author> <title>Text Segmentation by Topic,</title> <booktitle>in Proceedings of the First European Conference on Research and Advanced Technology for Digitial Libraries, </booktitle> <pages>pp. 120-129,</pages> <date>1997.</date>

<author>J. C. Butcher.</author> <title>General linear method: A survey.</title> <journal>Appl. Numer. Math.,</journal> <volume>1 273, </volume> <pages>1985.</pages>

<author>J. C. Butcher.</author> <title>General linear method: A survey.</title> <journal>Appl. Numer. Math.,</journal> <volume>1</volume> <pages>273,</pages> <date>1985.</date>
(corresponding correct parse at position N=3)

Figure 5. The first two references illustrate cases in which our system generates the correct output. In the third case an example of an incorrect parse is shown with the corresponding correct parse and its position within the best parses.

usage. It consists of 500 research paper citations that we partitioned into a training set composed of the firsts 350 entries and a test set including the remaining ones. Unfortunately no explicit partitioning scheme has been published for comparison issues by other researchers [3, 10, 14]. This prevents an exact performance comparison as changes to the training/testing set partitioning may lead to deviations of system accuracy.

For quantifying the performance of the system we use the common measures of word, field and instance accuracy – as introduced in [3, 14] – which are defined as follows:

$$\text{word accuracy} = \frac{|\text{correctly recognized words}|}{|\text{words}|}$$

$$\text{field accuracy} = \frac{|\text{correctly recognized subfields}|}{|\text{subfields}|}$$

$$\text{instance accuracy} = \frac{|\text{correctly recognized references}|}{|\text{references}|}$$

Table 1. Accuracy comparison on the Cora data set [%].

	word	field	instance
CRF [8, 14]	95.4		77.3
PFST [this work]	88.5	82.6	42.7
HMM [10, 16]	85.1		10.0
INFOMAP [3]		73.3	

The different measures are structured in a hierarchical sense to give a better impression of the system's performance. Word accuracy favors fields with many words, field accuracy favors fields with few words and instance accuracy gives a picture of how well the system performs in an overall view as only completely correct references are counted. An overview of systems that have been evaluated on the Cora reference dataset is given in Table 1. The CRF-based system by [8, 14] performs best on the Cora dataset. It is worthwhile noting that they are not operating on the plain text but extract a set of various features from the reference, e.g. layout properties or dictionary matching. Our approach performs slightly worse on the word accuracy and significantly worse on the instance accuracy in comparison. The HMM-based system by [10, 16] performs slightly worse on the word accuracy and remarkably worse on the instance accuracy than our system. As the rule-based approach by [3] only measured the field specific performance no overall comparison is possible. Nevertheless it can be claimed that our approach performs better due to the significant accuracy difference. Thus our system seems to be the second best in regard to the Cora dataset. Since other relevant bibliographic reference recognition systems [1, 2, 6, 13, 17] have not been evaluated on Cora we can not make statements regarding performance comparisons.

The instance accuracies differ more between the systems than the word accuracies as a slightly worse classification of single words leads to a higher rate of incorrectly labeled references. If we make a few simplifying assumptions, we have a word accuracy p and the number of words in a given reference l. A reference is then classified correctly only if all words are labeled correctly and the probability of that happening is p^l. For two given systems the one with the better word accuracy has a significantly (in relation to the difference of the word accuracies) higher probability to classify a complete instance correctly.

The relatively straightforward structure of the current language model indicates that our system's performance can be increased significantly by replacing the intrafield unigram with a bi- or trigram and eventually the interfield bigram with a trigram. Also some models for common representations (i.e. authors' names or dates) could be additionally incorporated for specializing the system further according to a set of reference styles. We would have to find an acceptable tradeoff between the system's runtime and accuracy. Additionally we want to avoid any overfitting of the system to a specific type of reference style to maintain its robustness.

Example outputs of the system are shown in Figure 5.

6 Conclusion

As finite state transducers have proven their applicability to many tasks of computational linguistics they seem a natural choice for deriving a language model adjusted to the recognition of bibliographic references. Especially in comparison to rule-based approaches our system yields a higher degree of robustness and adaptability as no strict rulings are enforced and local classification errors should not propagate through the entire reference. Furthermore we eliminate the need for a domain expert and thus decrease the time requirements for adapting the system to changing demands. While normally rules would have to be manually derived after analysis of the various reference styles we just have to rerun the training procedure on an adequate dataset representing the syntactical differences of the styles. As our research group is currently developing a PFST-based optical character recognition system called OCRopus[4] an integration with the presented system is planned to allow operation on imaged input.

References

[1] D. Besagni and A. Belaid. Citation recognition for scientific publications in digital libraries. *Document Image Analysis for Libraries*, 00:244, 2004.

[2] D. Besagni, A. Belaid, and N. Benet. A segmentation method for bibliographic references by contextual tagging of fields. In *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, page 384, Washington, DC, USA, 2003. IEEE Computer Society.

[3] M.-Y. Day, R. T.-H. Tsai, C.-L. Sung, C.-C. Hsieh, C.-W. Lee, S.-H. Wu, K.-P. Wu, C.-S. Ong, and W.-L. Hsu. Reference metadata extraction using a hierarchical knowledge representation framework, December 2006.

[4] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: an automatic citation indexing system. In *DL '98: Proceedings of the third ACM conference on Digital libraries*, pages 89–98, New York, NY, USA, 1998. ACM Press.

[5] A. A. Goodrum, K. W. McCain, S. Lawrence, and C. Giles. Scholarly publishing in the Internet age: A citation analysis of computer science literature. *Inf. Process. Manage.*, 37(5):661–675, 2001.

[6] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 37–48, Washington, DC, USA, 2003. IEEE Computer Society.

[7] L. Hetherington. The MIT finite-state transducer toolkit for speech and language processing. In *INTERSPEECH 2004 - ICSLP: Proceedings of the Eighth International Conference on Spoken Language Processing*, pages 2609–2612, Jeju Island, Korea, October 2004.

[8] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[9] S. Lawrence, C. L. Giles, and K. D. Bollacker. Autonomous citation matching. In *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*, pages 392–393, New York, NY, USA, 1999. ACM Press.

[10] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. A machine learning approach to building domain-specific search engines. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 662–667, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[11] M. Mohri. Finite-state transducers in language and speech processing. *Comput. Linguist.*, 23(2):269–311, 1997.

[12] M. Mohri, F. Pereira, and M. Riley. The design principles of a weighted finite-state transducer library. *Theor. Comput. Sci.*, 231(1):17–32, 2000.

[13] F. Parmentier and A. Belaid. Logical structure recognition of scientific bibliographic references. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, page 1072ff., Washington, DC, USA, 1997. IEEE Computer Society.

[14] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*, pages 329–336, 2004.

[15] E. Roche and Y. Schabes. *Finite-State Language Processing*. MIT Press, Cambridge, MA, USA, 1997.

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

[17] A. Takasu. Bibliographic attribute extraction from erroneous references based on a statistical model. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 49–60, Washington, DC, USA, 2003. IEEE Computer Society.

[18] F. Thollard, C. de la Higuera, and R. C. Carrasco. Probabilistic finite-state machines-part i + ii. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1013–1039, 2005.

[4] http://www.ocropus.org