Kairos: Proactive Harvesting of Research Paper Metadata from Scientific Conference Web Sites

Markus Hänse[1], Min-Yen Kan[2], and Achim P. Karduck[1]

[1] Hochschule Furtwangen University
mhaense@gmail.com, karduck@hs-furtwangen.de
[2] Department of Computer Science, National University of Singapore
kanmy@comp.nus.edu.sg

Abstract. We investigate the automatic harvesting of research paper metadata from recent scholarly events. Our system, Kairos, combines a focused crawler and an information extraction engine, to convert a list of conference websites into a index filled with fields of metadata that correspond to individual papers. Using event date metadata extracted from the conference website, Kairos proactively harvests metadata about the individual papers soon after they are made public. We use a Maximum Entropy classifier to classify uniform resource locators (URLs) as scientific conference websites and use Conditional Random Fields (CRF) to extract individual paper metadata from such websites. Experiments show an acceptable measure of classification accuracy of over 95% for each of the two components.

1 Introduction

With the growing trends of digital publishing and open access publishing, scientific progress is increasingly reliant on near-instantaneous access to published research results. It is now common to find published articles citing work within the same year and even works that have yet to be formally published. Established imprints (such as Wiley, Elsevier and Springer) have adopted Really Simple Syndication (RSS), a web feed standard, to help readers stay abreast of recent news and articles. In the biomedical field, PubMed[3] serves an example of a one-stop aggregator that gives up-to-date access to the large bulk of scientific advances. However, in some fields such as computer sciences and engineering, such aggregators do not exist – hampering the ability of researchers to stay current.

There are a myriad of reasons for this that are both cultural and practical. PubMed relies on manual effort by authors and publishers to keep the information up-to-date. Also, in computer science, many cutting-edge research results are transmitted through conferences rather than journals, and such conferences often do not have RSS feeds for metadata of individual scholarly papers.

To address this problem, the communities has built a number of digital libraries – most notably CiteSeerX[4] and Google Scholar[5] – that index web-

[3] http://www.ncbi.nlm.nih.gov/pubmed/
[4] http://citeseerx.ist.psu.edu/
[5] http://www.scholar.google.com

available scientific papers. However, these crawlers are largely *reactive* – periodically scanning the web for new contributions and indexing these as they come about. To really address the need for up-to-date indexing, we must provide digital libraries with crawlers that are *proactive*: crawlers that know when a conference has just occurred and know when and where to obtain the pertinent paper metadata from each conference web site.

This paper details Kairos[6], an implemented system that aims to address this problem. Kairos uses supervised machine learning to model which URLs are indeed conference websites and to model how such conference websites present individual paper metadata. Furthermore, by extracting event date information from conference websites, Kairos can proactively schedule crawls to the event website as the date of the conference approaches. After briefly discussing related work, we give an overview of the architecture of our system in Section 3 and describe the two major components of Kairos – the crawler and the IE engine – in Sections 4 and 5 and their evaluation, in turn. We conclude by discussing the project in its larger context.

2 Related Work

Digital libraries have turned to focused crawling to harvest materials for collection. Traditionally, this had been done by downloading web pages and assessing whether they are useful. By useful we mean that crawled web pages are belonging to the topic we would like to crawl. However, this can be wasteful, as downloaded pages that are not useful consume bandwidth. As such, a key step is to estimate the usefulness of a page before downloading. In particular, scientists lately have refined the focused crawling heuristics, exploiting genre [1] and using priority estimation [2]. A number of experiments [3–5] have shown that the careful analysis of URLs can be effective estimators.

Once useful webpages are identified, information must be extracted from the semi-structured text of the webpage. While many different models have been proposed, the conditional random field (CRF) model pairs together pointwise classification with sequence labeling. CRFs have been applied to a multitude of information extraction and sequence labeling tasks [6–8]. While general CRFs can handle arbitrary dependencies among output classes, for textual NLP tasks a linear-chain CRF model often outperforms other models while maintaining tractable complexity.

Focused crawling and information extraction are often used serially in many applications that distill data from the web, however, to our knowledge, there have been few works that discuss their integration in a single application.

3 Overview

We introduce Kairos pictorially and trace the subsequent workflow, from the beginning input to the resulting extracted paper metadata per conference. Figure 1 shows an architectural overview of our system.

[6] *καιρόσ*, a classical Greek word for "opportune moment".

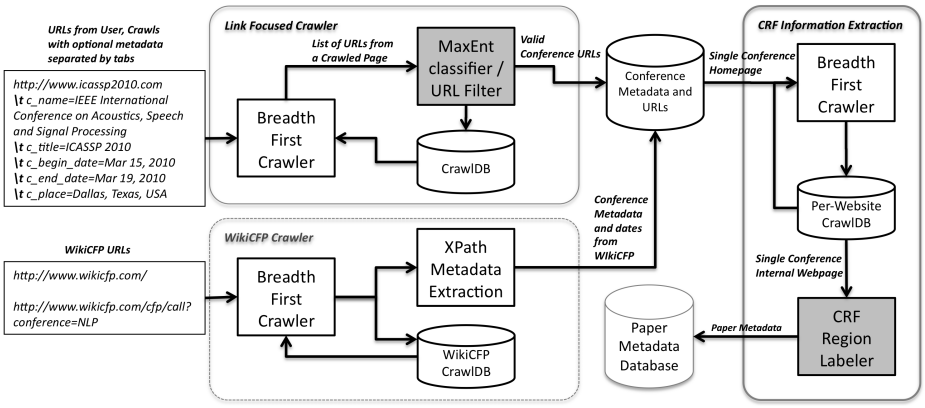

Fig. 1. Architectural overview of Kairos. New modules of interest highlighted in gray.

Kairos consists of two components: 1) a crawler that uses a maximum entropy classifier to determine whether input URLs are scientific conference websites and 2) an information extraction engine that extracts individual paper metadata from a web page using a linear chain conditional random field classifier.

The first component, the crawler, shown on the top left of Figure 1, takes a list of candidate URLs as input, and decides whether they are indeed scientific conference websites. If they are, key metadata about the event date is extracted for later use in the second component. For example, given a conference that issues its first call-for-papers, Kairos' crawling component will attempt to locate the correct site for the conference and extract the key dates for its deadlines (paper notification and actual conference date).

The second component, the IE engine, shown on the right of Figure 1, is run periodically around the dates for each conference. It is encapsulated in its own per-conference crawler that does a periodic crawl of a particular conference's website around its key dates, looking for webpages where individual paper metadata have been posted. The IE engine is run during these crawls, extracting individual paper metadata from the conference website when and where they are found, and converting them to rows of paper metadata and PDF link locations. This final product can be then ingested into a digital library.

As discussed in the introduction, certain web sites also act as scientific event portals, listing many different conferences and other scholarly events and meetings. In the computer science community, both DBLP[7] and WikiCFP[8] are such websites. WikiCFP, in particular, allows external parties to advertise information about workshops and conferences. It has become a major aggregator of information related to scholarly paper submission deadlines in this community, listing over 5,200 conference venues, as of January 2010. WikiCFP also tracks

[7] http://www.informatik.uni-trier.de/~ley/db
[8] http://www.wikicfp.com/

specific conference metadata, including the URL of the conference website, the date range for the conference itself, submission and camera ready deadlines.

WikiCFP sub-crawler. For these reasons, we also implemented a component customized for WikiCFP. This is shown in the bottom left of Figure 1. This sub-crawler collects and extracts conference metadata using XPath queries from publicly visible pages in WikiCFP, including the author notification date and the dates of the meeting itself. We extract these two dates from both the sub-crawler for WikiCFP, as well as any other URLs that are encountered or entered. These two dates are selected as the list of papers to appear at the conference is occasionally posted after the author notification date and the full paper metadata and links to source papers often appears after the event starts. Kairos schedules its IE engine runs around these key dates per conference.

All of the crawler instances in Kairos are built on top of Nutch[9], an open-source crawler. Nutch itself builds on Lucene, a high-performance, text search engine library.

4 Link based Focused Crawling

The focused crawler requires a module that judges the fitness of a link, to decide whether or not a potential page may lead to useful metadata to extract. In one guise, such a module may take a URL as input and output a binary classification. We place our classification module in between the Fetcher and Injector modules of Nutch, so that it can make judgments on URLs found in newly downloaded web pages, to save bandwidth.

Before computing individual features from the candidate URLs, we preprocess them into smaller units that are more amendable to analysis, similar to [4, 5]. We glean features from the three components in the URL: namely, the hostname, path and document components; hostnames are further broken up into top level domain (TLD), domain, and subdomain components. These and other components are further broken up into *tokens* by punctuation boundaries.

Given these ordered set of tokens, we compute a set of binary feature classes for maximum entropy (ME) based classification. We use ME (a pointwise classification) instead of CRFs for this, as the URLs are short and the modeling sequential information between instances is unnecessary. We now list feature classes that we extract.

- **Tokens and Length**: This feature class captures the length (in characters) and identity of the tokens with respect to their ordinal position in the component.
- **URL Component Length in Tokens**: This feature captures the number of tokens in the each component. Short domain names and/or path names can indicate good candidates for conference URLs.
- **Precedence Token Bigrams**: From the tokens created from the preprocessing step, we form bigrams as features. These include normal adjacent

[9] http://lucene.apache.org/nutch

bigrams as well as ones that contain a gap. The latter gapped bigrams are used to combat sparse data in this feature class. For example, given the hostname "www.isd.mel.nist.gov", we create (4+3+2+1 =) 10 individual binary features such as "www≺isd" and "www≺nist".
- **Ordinals**: This aims to detect ordinal numbers used in many full forms of conference and workshop meetings. We use a regular expression to detect such cases as "1st", "2nd" or "Twentieth".
- **Possible Year**: This class of features aims to find both double digit and full forms of years, which is also a frequent unit of information in conference URLs. Again, a regular expression is used to capture years. We normalize possible years in the YY format to their full form (e.g., "07" → "2007"). We also separate any component found as a prefix to the year and add special features if the candidate year detected is the current year. This last part helps to favor spidering conferences that are in this calendar year.

Table 4 illustrates an example URL that has gone through preprocessing and subsequent feature extraction.

Feature Class	**Example**
Original URL	http://www.aisb.org.uk/convention/aisb08/
Tokens & Length	H:0:www |H|0:3 H:1:aisb |H|1:4 H:2:org |H|2:3 H:3:uk
URL Component Length	|H|ALL:4 |P|ALL:2
Precedence Token Bigram	H:www|aisb H:www|org H:www|uk H:aisb|org H:aisb|uk H:org|uk P:convention|aisb08
Ordinals	-
Possible Year	P:1:aisb P:1:08 P:1:2008 P:1:aisb2008 P:1:aisb<year> P:1:<year> P:1:4<year> P:1:aisb08

Table 1. Maximum Entropy features extracted from one sample URL.

4.1 Collecting Training Data

To train our classifier, we use both WikiCFP and DBLP as positive seed instances and collect negative instances through a search engine API. WikiCFP provides a semi-structured, fixed format that provides the conference name, date and website URL. We extract these fields from conference information pages in WikiCFP, using the same crawler component discussed earlier (bottom left of Figure 1. DBLP also provides an XML dump that also lists the conference title, year and URL information, which are similarly extracted.

To construct negative instances, we need to find URLs that are not pages which contain metadata about papers. However, for the negative instances to be useful in discriminating positive from negative, they should also share some attributes in common with the positive examples. For this reason, we use tokens extracted from the positive URL instances to construct queries to a search engine to retrieve potential negative examples. For example, given a known positive URL of "*http://www.icadl2010.org*", we construct a query "*inurl:(icadl 2010 org) filetype:html -site:www.icadl2010.org*" and send to a search engine to retrieve other webpage URLs that are negative examples. Often, such URLs are call-for-papers for the target conference that have listed elsewhere or blog posts about

the conference, as well as other miscellaneous information. Our inspection of this collection process suggests that it is largely accurate, but that some URLs are false negatives (i.e., are actual conference websites).

4.2 Evaluation

We used the above methodology to retrieve training URLs for our classifier, that were automatically labeled as positive and negative instances. In total, we retrieved 9,530 URLs, of which 4,765 were positive instances. To assess the effectiveness of our classifier, we used stratified 10-fold cross validation: the dataset was randomly divided into 10 equal parts, each with the original proportion of positive and negative instances. We use a publicly available maximum entropy classifier implementation[10], and train the classifier on nine parts and test on the remaining part, and repeat this process ten times. The resulting binary URL classifier achieves an accuracy, precision, recall and F_1 measure of 96.0%, 94.8%, 97.2% and .960 respectively, when conference URLs are considered positive.

Table 2 gives the resulting confusion matrix for our ten-fold cross validation test. These results show that there is a slight imbalance of the system to err towards false negatives.

		Gold Standard		**total**
		+ve	**-ve**	
System	**+ve**	4519	246	4765
Judgment	**-ve**	127	4638	4765
	total	4646	4884	9530

Table 2. Confusion matrix for the URL conference classifier.

5 CRF-based Information Extraction

Given a URL that passes the maximum entropy classifier's test for being a conference web page, Kairos downloads the web page represented by the URL and attempts to extract pertinent paper metadata from the page, if present. This second stage is run only during the key date periods as extracted from the first stage. This task represents a standard web information extraction task, where web pages may present paper metadata in different formats. In conference and workshop websites, paper metadata is commonly found in some (semi-)structured format, such as tables, paragraphs or lists.

As the conference website (and metadata gleaned from WikiCFP and DBLP) may already describe the venue and publisher information for paper that are presented at the venue, our system's goal is to identify the three remaining salient pieces of paper metadata: namely the *title*, *author* and *a link to the PDF*[11] of the paper itself.

[10] Version 2.5.2, available at http://maxent.sourceforge.net/

[11] Kairos currently only handles source papers in PDF, PostScript or MS Word formats, and ignores HTML versions of papers.

To accomplish this second task, we again make use of a supervised classifier. The task here is to scan a web page for the individual pieces of metadata related to papers. Different from the pointwise classification that characterizes the previous URL classification task, for this information extraction task, we now must deal with labeling and extracting multiple and related fields from a stream of (richly formatted) text. For this reason, we turn to methodologies developed for sequence labeling, and adopt conditional random fields (CRF) as our model representation.

We divide each input webpage into a set of regions, where regions are small blocks of text delimited from other parts of the page by certain HTML table, lists and logical tags: <P>, <LI>, <TR>, <TH>,
, <HR>. The resulting regions are meant to be minimal spans that are assigned a classification label.

As PDF links to papers are easily detected in anchor tags <A> which lead to documents with a ".pdf" extension, we concentrate on the harder task of identifying titles and authors from the remaining HTML text. From observation, we noted that for the purpose of our task, regions on conference pages can largely be classified into one of five classes: *title*, *author*, *author+affiliation*, *affiliation* and *n/a*. The *author+affiliation* class may seem redundant, but is needed as a separate class, as some formats multiplex authors and their affiliations together. Figure 2 shows such an example of such (albeit rarer) cases. Note that Kairos merely identifies the span of regions that contain this information; ingestion modules outside of the scope of Kairos are responsible for the data cleaning issue – transforming names into an appropriate canonical form (e.g., "D Lee" and "Lee, Dongwon" may be deemed as equivalent).

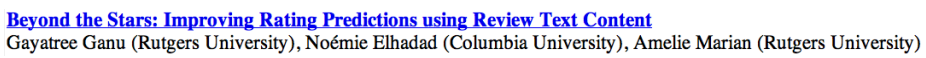

Fig. 2. An example block that shows interleaved authors and their affiliations.

5.1 Features for the CRF classifier

As in our URL classifier, we use region level features, but further enrich our feature set to handle this more challenging problem. As we are dealing text on web pages, a significant subset of our features make use of stylistic features specific to HTML. We also use domain-specific word lists to better detect specific fields. We list these three classes of features used by our classifier.

1. **Region (R)**: These features account for the lexical and grammatical identity of the tokens within each region to be classified, as well as the identities of the tokens within the region's nearby context.
   - identity (both as-is and lowercased, for canonicalization).
   - orthography (is the token in initial caps, or in all upper case? is it a number or does it contain punctuation?)
   - part of speech as determined by a part of speech tagger.
   - length: of the word, and of the containing "line" of text.

2. **Lexica (L)**: These lexica were largely borrowed from an open-source reference string parsing project and DBLP. Personal names, common words in affiliations and paper titles can help better detect the salient metadata that we target in the CRF extraction process.
   - token is present in a lexicon of personal names (297,332 entries; listing both given and surnames).
   - token is present in an affiliation lexicon (2714 entries).
   - token is present in a lexicon of commonly-used words in research paper titles (940 entries).
3. **Stylistic (S)**: These binary features capture any salient formatting of the region's token with respect to HTML tags. Titles and authors on conferences web pages are often formatted with bold, strong, italic or anchor spans.
   - use of bold <b>, italics <i> or emphasis , tags.
   - token is within the span of an anchor <a> tag.
   - token is part of a table cell or header <td>, <th> tag.

Currently, we do not use any positional features of regions. These features are calculated for each region of interest, and passed to a publicly available linear chain CRF implementation[12] for classification. After classication, a set of heuristics are used to bundle the classified spans of text into individual papers.

The final output can be then visualized as an index, with fields for *title* and *author* gleaned from the CRF system, the *PDF link* (if one exists) gleaned by regular expressions over the webpage, *venue name* and *venue location*, gleaned from the WikiCFP source or link focused crawler.

5.2 Dataset and Annotation Collection

To train and evaluate the CRF labeler, we downloaded positive examples of conference URLs, gathered in the previous task from WikiCFP and DBLP. However, unlike the previous task, there is no automatic means of creating positive and negative instances of each class. As such, we needed to manually label a corpus of conference pages with the appropriate classes.

To ensure impartial and replicable annotations, we prepared a corpus of 265 conference pages, presegmented into regions for annotation by human subjects. We recruited 30 student volunteers to annotate these web pages according to an annotation guide, which standardized our instructions. Each volunteer was asked to label around 20 pages with the use of in-house labeling software, which took them between 20 minutes to an hour. Subjects were given a token of appreciation for their participation in the data collection task. Each page was thus annotated by two volunteers. The first author of this paper also carefully annotated all regions in the 276-page collection. The resulting dataset consists of 9015 title, 7863 author, 2044 author+affiliation, 1437 affiliation regions.

We use the volunteers' labels to check the reliability of our own annotation. After discarding annotations from volunteers whom misunderstood the guidelines, we calculated the inter-annotator reliability of the acquired annotations.

[12] CRF++ 0.53, available at http://crfpp.sourceforge.net/

We calculated a Kappa[13] [9] of .931, indicating very high agreement among annotators. We take this as confirmation that the task is feasible and that the first author's annotations can be used as a gold standard.

5.3 Evaluation

We trained and evaluated the information extraction engine using ten-fold cross validation on a region level, where a region is one or more contiguous lines delimited by certain HTML tags. We only trained the CRF with the title, author regions and treated the author+affiliation regions as an author region, as it also includes author metadata. The CRF achieves an accuracy and F_1 measure of 97.4% and .974, respectively. To assess which features were most critical, we also performed ablation tests: removing different feature classes, retraining the model and then assessing the subsequent performance. These results are reported in Table 4, which show that all three classes of features were important, but that the region and gazetteer features were the most helpful.

		Gold Standard		**total**
		title	**author**	
System	**title**	7506	234	7740
Judgment	**author**	174	7566	7740
	total	7680	7800	15480

Table 3. CRF region labeling confusion matrix.

6 Conclusion

We have described Kairos, an end-to-end scholarly paper metadata harvester, which autonomously finds and extracts paper metadata from seed URLs and proactive focused crawls. By knowing the key dates of a conference event, our system can locate and harvest such metadata with a shorter delay than other digital libraries and databases. Kairos is built on top of Nutch, a popular open source crawling system, adding a maximum entropy (ME) based URL classifier converting it to a focused crawler amenable to detecting webpages that may contain conference paper metadata. A conditional random field (CRF) subsequently runs on downloaded pages to identify and extract pertinent title, author data per scholarly work. Both the ME and CRF classifiers obtain good performance – over 95% accuracy.

In total, the dataset collections and annotations for both stages took over a man-month of time. As the task was very laborious, we believe that these datasets would be valuable for those also targeting similar work in the future. As such we have made these datasets available to the general public[14]. An annotation system has been set up to allow volunteers to hand-annotate scientific conference websites to further expand the existing labeled training dataset.

[13] a measure that falls between -1 (indicating complete disagreement) to 1 (total agreement)

[14] http://wing.comp.nus.edu.sg/∼mhaense

Feature Classes	Accuracy	Precision	Recall	F_1
Stylistic (S)	69.4%	97.8%	62.4%	.762
Lexica (L)	85.9%	83.3%	87.9%	.855 (+.093)
L+S	87.8%	84.5%	90.5%	.874 (+.019)
Region (R)	95.6%	94.8%	97.0%	.959 (+.085)
R+S	96.3%	95.5%	97.0%	.962 (+.003)
R+L	96.7%	96.1%	97.3%	.967 (+.005)
All (R+L+S)	97.4%	97.0%	97.7%	.974 (+.007)

Table 4. CRF feature class ablation performance. F_1 performance gain over previous row given in parentheses. All performance gains are significant at the .01 level.

In this work, we have concentrated on catering for cases in which paper metadata is presented in plain text. In future work, we plan to integrate abilities to deal with more proactive forms of publication and subscription (pub/sub): ingesting publisher RSS feeds, and exporting discovered metadata in RDFa micro formats or making output OAI-PMH compliant. In our current work, we are integrating these modules into our production scholarly digital library. When the integration is complete, we will be close to making the real-time indexing of scientific articles a reality.

References

1. de Assis, G.T., Laender, A.H., Gonçalves, M.A., da Silva, A.S.: Exploiting genre in focused crawling. In: String Processing and Information Retrieval. Volume 4726 of Lecture Notes in Computer Science. (2007) 62–73
2. Guan, Z., Wang, C., Chen, C., Bu, J., Wang, J.: Guide focused crawler efficiently and effectively using on-line topical importance estimation. In: SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2008) 757–758
3. Sun, A., Lim, E., Ng, W.: Web classification using support vector machine. In: Proceedings of the 4th international workshop on Web information and data management, ACM New York, NY, USA (2002) 96–99
4. Kan, M.Y., Thi, H.O.N.: Fast webpage classification using URL features. In: Proceedings of Conference on Information and Knowledge Management. (2005) 325–326
5. Baykan, E., Henzinger, M., Marian, L., Weber, I.: Purely URL-based topic classification. In: Proceedings of the 18th international World Wide Web Conference. (2009) 1109–1110
6. Peng, F., McCallum, A.: Information extraction from research papers using conditional random fields. Information Processing and Management **42**(4) (2006) 963–979
7. Sarawagi, S., Cohen, W.: Semi-markov conditional random fields for information extraction. Advances in Neural Info. Processing Systems **17** (2005) 1185–1192
8. Kristjansson, T., Culotta, A., Viola, P., McCallum, A.: Interactive information extraction with constrained conditional random fields. In: Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2004) 412–418
9. Carletta, J.C.: Assessing agreement on classification tasks: the kappa statistic. Computational Linguistics **22**(2) (1996) 249–254