Annotated Corpus of Scientific Conference's Homepages for Information Extraction

Piotr Andruszkiewicz, Rafał Hazan
Instititute of Computer Science, Warsaw University of Technology
Warsaw, Poland
P.Andruszkiewicz@ii.pw.edu.pl, R.Hazan@stud.elka.pw.edu.pl

Abstract

In this paper, we present a new corpus that contains 943 homepages of scientific conferences, 14794 including subpages, with annotations of interesting information: name of a conference, its abbreviation, place, and several important dates; that is, submission, notification, and camera ready dates. The topics of conferences included in the corpus are equally distributed over five areas: artificial intelligence, natural language processing, computer science, telecommunication, and image processing. The corpus is publicly available. Beside the characteristics of the corpus, we present the results of information extraction from the corpus using SVM and CRF models as we would like this corpus to be considered a reference data set for this type of task.

Keywords: annotated corpus, scientific conference's homepages, information extraction

1. Introduction

Up-to-date information about conferences is important for scientists who track conferences of their interest and check, e.g., dates of a conference, deadlines, that could change especially during submission period. Thus, a system that gathers such information could ease scientists' lives. The system should collect data about conferences and keep it up-to-date. Moreover, it should provide data in a structured way to facilitate searching conferences and obtaining information about any changes. The crucial part of this system are methods for collecting data about conferences automatically, e.g., homepages of a conference for the current and previous years, when and where a conference will be held, submission, notification, camera ready dates, etc.

In this paper, we present a new corpus that contains homepages of conferences with annotations of interesting information, e.g., name of a conference, its abbreviation, several important dates for the conference. The motivation behind this task was that according to our knowledge there is no any publicly available corpus in such a domain. The corpus can be used to train a tool for information extraction from unstructured sources containing data describing conferences. We chose conference home pages as a source as they contain up-to-date information. Structured services, such as WikiCFP, do not always update information (e.g., when deadline changes) and cannot be used in a real system for gathering up-to-date information about conferences.

Beside the characteristics of the corpus, we present results of information extraction as a baseline and a proof that this corpus can be used as a reference data set for information extraction from homepages of conferences. The corpus is publicly available and can be downloaded from the following website http://ii.pw.edu.pl/~pandrusz/data/conferences/.

The remainder of this paper is organised as follows: In Section 2. we describe the corpus we created. In Sections 3. the preprocessing and features are described. The experimental results are discussed in Section 4.. Section 5. presents related work. Finally, Section 6. summarizes the conclusions of the study and outlines avenues to explore in the future.

2. The Corpus

On the internet one can find corpora for information extraction, e.g., the corpus for information extraction from researcher's homepages (Tang et al., 2008), seminar announcements (Califf and Mooney, 1999; Freitag and McCallum, 1999). However, we could not find any publicly available corpus for scientific conferences. Therefore, we created an annotated corpus for the task of information extraction from homepages of scientific conferences. This corpus is publicly available and can be found on the website http://ii.pw.edu.pl/~pandrusz/data/conferences/.

Our decision to collect homepages of conferences, not Call For Papers (CFPs), is based on the following findings. We verified 100 passed conferences from our corpus for which we were able to find a running homepage and determine the important dates for a conference. Then we compared the data from a homepage and from WikiCFP service. It appeared that in WikiCFP about 70% of conferences have not up-to-date information about important dates, mostly submission date, as this date changes most often as the deadline approaches/passes. The dates are stable until the submission date comes, then dates are changed on a homepage, however, they are not updated in WikiCFP. Furthermore, data provided in CFPs is limited, e.g., it usually lacks information about sponsors. In (Xin et al., 2008) the authors stated that only less than 10% of CFPs analysed by them presented information about sponsors. Moreover, a service might not have information about conference we are looking for because it is field specific or covers only small part of all conferences in the field. According to authors of (Xin et al., 2008), it was possible to find only 40% of the textual CFPs of the top 293 computer science conferences listed at *Citeseer* (http://citeseer.ist.psu.edu/impact.html), while searching such conference services.

In our work CFPs proved useful in gathering not detailed information on conferences; namely, the list of conference webpage addresses. During the process of gathering

the corpus we wanted to make it as automatic as possible. To that end, as a first step we gathered a list of conferences from a conference hub. We chose the WikiCFP (http://wikicfp.com/cfp/) for that purpose, as it is a well known service and contains CFPs for areas we are interested in. Then we downloaded the homepage link and other data about each conference from WikiCFP. After that we downloaded the homepages and subpages (the depth level was restricted to one) within the same domain as the main page.

In the next step of corpus creation, for each conference (sub)page and each entity, e.g., submission date equal to 15 January 2015, we automatically found all instances of this entity and annotated them in the html source code of the web page.

The method of searching for an instance of the entity could not be a simple comparison of characters for several reasons. For instance, there are different ways of writing dates, the names of the conference provided in WikiCFP could differ slightly from the name on the page. Thus, we employed the following method for name of a conference comparison. We removed all conference name stop-words, e.g., word Workshop from Workshop on Collaborative Online Organizations. This led to Collaborative Online Organization that was searched. The conference name stop-words list has been manually created and consists of: The, International, Conference, Workshop. Moreover, we allow other single words to appear between words that were being searched. We applied case sensitive search. When conference name stop-words were neighbours of found instances, we added these conference name stop-words to annotation as their constitute a name. Though we do not annotate the year number and the consecutive number of the conference if they appear at the beginning of the name. To deal with different formats of date, we employed GATE tool and its default JAPE (Java Annotation Patterns Engine) rules (Kenter and Maynard, 2005). After the automatic process, the annotations were verified by three persons and manually corrected/added where necessary. This step is necessary as WikiCFP may not have up-to-date information as already explained. In case of disagreement majority voting was used.

The corpus we created contains 943 annotated homepages, 14794 pages including subpages, of scientific conferences. Hence, there are more than 15 pages per conference on average. The topics of conferences are equally distributed over five topics; namely, artificial intelligence, natural language processing, computer science, telecommunication, and image processing. The following entities were annotated: name and abbreviation of the conference, place, dates of the conference, submission, notification, final version due dates, the tags used in corpus are *cname*, *abbre*, *where*, *when*, *subm*, *notf*, *finv*, respectively. The annotated entity types are the most important considering a system that gathers information about conferences and is used by scientists to track conferences of their interest. However, we plan to annotate additional entity types, e.g., general and local chairs, invited speakers, sponsors.

The statistics of the corpus are presented in Table 1. Column *Avg. length* presents an average token length of an entity type. Thus, *abbreviation* contains one token. The name of a *place*, where a conference is held, sometimes consists of two tokens, which is consistent with names of cities or countries, e.g., New Zealand. The length of the name of a conference is almost 7. The lenght of each *important date* is about 3.5 and is consistent within the dates. The *date* of a conference is longer (about 4.8) because it contains a range of days. Two next columns *Inst.* and *Inst. per conference* present the number of instances of an entity type in the corpus and the average number per conference. *Important dates* are less frequent on homepages. The most frequent is *place* and surprisingly it is mentioned over 80 times per conference.

Entity type	Tag	Avg. length	Inst.	Inst. per conference
Name	cname	6.93	9954	10.6
Abbreviation	abbre	1.00	52222	55.4
Place	where	1.07	79091	83.9
Date	when	4.78	11261	11.9
Submission	subm	3.54	3196	3.4
Notification	notf	3.56	2081	2.2
Final ver. due	finv	3.54	3851	4.1

Table 1: The characteristics of the corpus.

3. Preprocessing and Features

To reduce the number of features, we use Snowball stemmer (Porter, 2001) in the preprocessing phase. We also remove words from a custom stoplist. Words that often occur in conference names, e.g., 'the', 'and', 'on', are not included in the stoplist.

We extract a main article or paragraphs from a web page using Boilerpipe (Kohlschütter et al., 2010) library. Text is not removed from the web page in order to avoid situation in which important elements are removed by mistake.

In our approach, we distinguish four group of features; namely, local, offset, layout, and dictionary features.

3.1. Local Features

Local features are created based on a current word that is being analysed. The commonly used feature is a *word*. This feature is not created for words from the stoplist and those tokens that contain nonalphabetic characters. The second feature contains part of speech (POS) tags for a current word calculated by *Penn Pos Tagger* from *factorie* package (McCallum et al., 2009). *Short word* feature is assigned a value *true* for words containing 2 to 5 characters. *Shape of a word* represents numbers with *1*, capital letters with *A* and small letters with *a*. If there are more than two the same characters in the value of this feature, the sequence is reduced to two characters. For *type of a word* feature we created eight types of words. *Short phrase* is set for words being a sequence of length of one or two words, for instance, named entities with two words, e.g., Carl Brunto. *Long phrase* indicates words of sequences with at least three words. We distinct between short and long phrases because conference names are usually long phrases

Table 2: The importance of features groups for entities (F1 measure, the best results marked in bold).

Features	Name	Abbrev.	Place	Date	Submission	Notification	Final ver. due
All	**0.36**	**0.76**	**0.67**	**0.80**	**0.60**	0.46	**0.65**
Without local	0.09	0.55	0.66	0.33	0.50	0.35	0.52
Without offset	0.33	0.68	0.62	0.67	0.00	0.00	0.00
Without layout	0.26	0.52	0.54	0.71	0.58	0.48	0.60
Without dict.	0.33	0.74	0.55	0.69	0.56	**0.49**	0.58

and locations of conferences are usually short. *Date* indicates dates that are present on a web page. Other types are: *Number* - assigned for numbers, e.g., 12, 1st; *acronym* indicates words of the following shapes: AAaa, AaAA, AAa, AA, AaaAaa AaaAA, AA1AA, AAaAA, *punctuation marks*, *special char* represents nonalphanumeric chars that are not punctuation marks. Other words are marked with *standard word* type and represent words that probably are not interesting in the case of information we want to extract.

3.2. Offset Features

Predecessor represents features calculated for the word that precedes a current word. We assume that we take into account only *type of a word* feature for one predecessor. *Successor* feature is constructed in the analogous way. Important dates of conferences represented as lists or tables are easy to understand for humans and hard to process by machine learning algorithms. We can find dates on the left, right, below or above a description of a date. We created date surrounding words feature to help machine learning algorithms in important dates extraction. It describes a date by up to six words before a date. If a date is followed by a colon then it contains up to six words after a date. The words from *date surrounding words* feature are used to calculate features for a current word. We create these features only for dates, because we do not want to increase the number of features too much.

3.3. Layout Features

Block feature informs about the blocks a word belongs to. We assign a separated value for each of the following blocks: head title, title, subtitle, paragraph, list, and table as the distribution over blocks differ for entities of interest. The number of a paragraph for a word is represented by *Paragraph number* feature. We consider only first six paragraphs because more than half of interesting entities is present in these paragraphs based on the corpus. This feature is important for conference names and abbreviations, dates and locations of conferences detection as these entities often occur at the beginning of a web page, according to our corpus. The important dates usually lie in further parts of a web page.

One of the subpages of a main conference homepage may contain entities of interest. Therefore, subpages are added to the training data. We restrict subpages to only those accessible by links with the following names: index, home, call for papers, registration, important dates. Moreover, each word from subpage is indicated by *subpage* feature containg anchor text, e.g., SUB=index.

Words modified by the following HTML tags: STRONG, B (bold), U (underlined), and FONT (use different fonts) are marked with *Emphasised* feature. This feature is meant for dates of a conference as they are more often underlined. Abbreviations and names of conferences do not show this regularity.

Links (A HTML tag) are represented by *Hyperlink* feature. This feature surprisingly indicates rather links to other conferences than information important in our task. Statistics calculated on our corpus confirm that.

3.4. Dictionary Features

Within *location* features, for a location in a web page a LOC=true, for a country COUNTRY=true and for a city CITY=true features are created. To calculate these features, gazetteer from ANNIE module of GATE (Kenter and Maynard, 2005) is used and location names from the corpus are added. The aforementioned features are helpful in conference location extraction.

Words that have not been found in the dictionary are marked with *Out of dictionary* feature. Our dictionary of English words contains 112505 words. This feature is designed for abbreviation extraction because this type of entity has the highest fraction of words not found in the dictionary.The feature suggests also location entity as it has the second highest value of not being found in the dictionary.

We created word dictionaries for place and date, name and abbreviation of a conference. They contain words that occur the most in sentences containing an important entity of a given type. Feature *promising surrounding words* marks words from sentences that contain at least one word from the dictionary. As the dictionaries are not mutually exclusive, *promising surrounding words* feature indicates that a word is important rather for entity extraction than for a specific entity type.

3.5. Multi-token Sequences

While describing features for our model, we assume that a single token; that is, a word, a number, or a nonalphanumeric character, is considered as a base object used by a model and assigned with one of interesting entity types, including *other* that means an object is not of one of the interesting entity types. This leads to a case when a sequence of tokens may have different entity types assigned even they are one entity of, e.g., conference name type. For instance, a sequence International Conference on Artificial Intelligence & Applications may have the following entity types assigned: International - conference name, Conference - conference name, on - other, Artificial - conference

name, and so on. Therefore, we expand a base object of a model to be a sequence of tokens that groups words forming one instance of entity. While detection of dates is an easy task, finding sequences that represent other named entities is not trivial. Hence, we prepared a heuristic algorithm customised for finding token sequences on conference web pages that is based on the following rules: each sequence consists of words that begin with a capital letter; these words may be separated by one word that starts with small letter; sequences are found within a sentence; a sequence cannot be separated by comma, dash nor colon. For example, words 'International Conference on Advancements in Information Technology' is treated by this algorithm as one sequence.

For sequences with at least two words we need to calculate features in one of the following ways: 1) calculate features for the first word only; 2) calculate features for each word separately and use all the features; 3) combine features for all words into one feature. For example, feature *word* is calculated according to the second approach and International Conference on Mechanics has the following features W=International, W=Conference, W=on, W=Mechanics. Third approach is used for POS features, e.g., 'Workshop on Applications of Software Agents' has a feature POS=INNNNNS.

4. Experiments

In our experiments we divide the corpus into training and test sets according to the proportion of 70/30. For Support Vector Machine (SVM) model (Cortes and Vapnik, 1995) cross validation is performed on the training set in order to find the best parameters, then the model is trained on the whole training set. We use LibSVM implementation (Chang and Lin, 2011). For multiclass classification we employ one versus the rest approach (Fan et al., 2008). For a web page we choose the only one entity of a given type that has the highest score among those indicated by an algorithm. Only *location* entity may have two instances because usually a country and a city is provided on a web page as a *location* of a conference.

4.1. Importance of Features

In our first group of experiments we verify using SVM how important the groups of features customised for information extraction from scientific conferences web pages are. We want to show how domain specific features influence the final results. As features in groups are sparse, a model with only one group of features would obtain very low accuracy and the comparison of models built with only one group of features would not be reliable. Therefore in each iteration we analyze all groups of features but one, in order to estimate how relevant is the group which was left out.

The results of the experiments are shown in Table 2. For all entity types but *Notification* we obtain the best results for all types of features included. For *Notification* we achieve the best results for the case without dictionary features, however, the results for all types of features case are not far behind (0.49 vs 0.46 in terms of F1). The results show that each group of features carries some information that is important for (at least one) interesting entity type. Thus, we could say that it is crucial to prepare features that are specific for a given domain. As the obtained results have shown, lack of some features may reduce the accuracy for some entity types to zero, for instance, lack of *offset* features for *important dates*.

For scientific conference web pages *local* features identify more general objects, such as dates and named entities that contain desired information. *Offset* features describe surroundings of a word, its context, which is necessary for *important dates* extraction. *Layout* features generate important features functions that inform on the localisation of a given word within a web page. They help in the case when an entity is not placed in the main text of a web page. *Dictionary* features improve the results mostly by its *location* feature that indicates potential places where a conference is held.

4.2. Models Comparison

Having the influence of features verified, we investigate the applicability of different models with regard to variations of their basic objects used; namely, single tokens, and sequences. In this set of experiments we use preprocessing and all the groups of features mentioned in Section 3..

For SVM model we start with comparison of single tokens and sequences used as basic objects that the model is working with. The results for linear SVM classifier run on single tokens as basic objects[1] are shown in the first row of Table 3. The accuracy of the model, also linear SVM, that uses sequences as basic objects is presented in the second row in the same table. The single token SVM performs significantly poorer than sequence SVM for *name* of a conference and *important dates*. The reason behind is that first model assigns a label to each single token independently and mentioned entities consists of several tokens. We try to ease SVM with this task by incorporating *offset* features, however, it seems that it is not enough to help single token SVM with extraction of entities that consist of several consecutive words. By providing the SVM already extracted potential sequences we overcome this problem. For sequence SVM we observe also 6 percentage points (p.p.) decrease in F1 for *abbreviation* detection, where linear SVM performs the best.

We present only the results of linear SVM because the non-linear SVM with RBF kernel function has not obtained significantly better results. Therefore, we stay with linear one due to less complexity and shorter training time. Our model has a high number of features, hence there is no need to increase the dimensionality by applying a kernel function (Hsu et al., 2003).

In our experiments we also use Linear Conditional Random Fields, CRF (Lafferty et al., 2001) with three different templates of factors. The first template connects factors with an input variable and an output variable. The second represents the relation between consecutive output variables. The third has only one argument that is an output variable. Single tokens CRF (Lin. CRF in Table 3) significantly outperforms both SVM models in *name* extraction (0.57 versus 0.36 and 0.15 in F1) due to the fact that it models sequences

[1] It means that the model assigns a label; that is, a type of entity, to a single token.

Features	Measure	Name	Abbrev.	Place	Date	Submission	Notification	Final ver. due
Lin. SVM	Precision	0.14	0.79	0.74	0.72	0.41	-	0.32
	Recall	0.16	0.86	0.59	0.79	0.06	0.00	0.08
	F1	0.15	**0.82**	0.66	0.76	0.11	0.00	0.13
Lin. SVM seq.	Precision	0.38	0.76	0.75	0.80	0.66	0.54	0.71
	Recall	0.34	0.75	0.60	0.80	0.54	0.40	0.59
	F1	0.36	0.76	**0.67**	0.80	0.60	0.46	**0.65**
Lin. CRF	Precision	0.74	0.75	0.66	0.82	0.73	0.25	0.56
	Recall	0.47	0.82	0.53	0.69	0.09	0.01	0.14
	F1	**0.57**	0.78	0.59	0.75	0.17	0.02	0.22
Lin. CRF seq.	Precision	0.61	0.77	0.66	0.82	0.67	0.63	0.70
	Recall	0.40	0.84	0.56	0.82	0.57	0.40	0.50
	F1	0.48	0.80	0.61	**0.82**	**0.61**	**0.49**	0.58

Table 3: The results of extraction for entities (the best F1 results marked in bold).

of label (SVM lacks this property). However, for entities that do not consist of several consecutive words we have not observed improvements; even contrary, we notice small decrease for *place* and *date*. Surprisingly, single token CRF cannot handle *important dates* extraction like in the case of single token SVM. However, sequence CRF (Lin. CRF seq. in Table 3) discovers them on the comparable level to sequence SVM. Both models based on sequences handle *important dates* significantly better because the sequence discovery algorithm extracts potential entities, that may have different formats, very well. Moreover, sequences also help CRF in *date* extraction (the best obtained results), like for SVM. Sequences discovery for *name* is not as good as discovering sequences for *important* dates. That is why we observe 9 p.p. decrease in extraction of that entity for CRF based on sequences compared to the one based on single tokens. However, sequences slightly increase CRF results for *abbreviation* and *place*.

Summarising, linear CRF based on single tokens outperforms other models for *name*. Linear SVM, also based on single tokens, obtains the best results for *abbreviation*. Dates are extracted better with models based on sequences than single tokens. For *place* the winner is SVM on both single tokens and sequences (SVM on sequences outperforms SVM on single tokens by only 1 p.p.), however, all other models are not worse than 8 p.p. in terms of F1. Thus, different models may be used for specific entity types in order to achieve the best cumulative results.

5. Related work

Previous works in that field focused mostly on information extraction from CFPs using different approaches. Extracting information from CFPs has drawbacks mentioned in Section 2.. In (Lazarinis, 1998) rule based method was employed to extract date and country from a CFP. Linear CRF was used in (Schneider, 2006) in order to extract seven attributes about conferences from CFPs with the use of layout features. However, in this approach only plain text of CFPs was used and layout features were based on lines of text, indicating, e.g., first token in line or first line in the text. We use HTML sourcecode of web pages, including formatting. Thus, our data has much more richer layout. In (Ireson et al., 2005) a general platform for performing and assessing information extraction from workshop CFPs was described. The platform was used in Pascal Challenge on Evaluating Machine Learning for Information Extraction. The organizers of the challenge provided a standardised corpus of CFPs, a set of tasks, and methodology for evaluation. The results of the challenge can be found in the aforementioned paper. Issertial and Tsuji (2011) focused also on information extraction from CFPs, including that which come via e-mails. They used rule-based methods to extract information about conferences from conference services, like WikiCFP, and combined them in one system in order to facilitate the process of finding conferences that are of interest for a user. In contrast to aforementioned works (Xin et al., 2008) extracted information about conferences from web pages with Constrained Hierarchical Conditional Random Fields. However, the set of homepages used in experiments has not been published. We created the annotated corpus, performed extraction and made both the corpus and the results public in order to encourage researchers to improve the baseline for this corpus.

In information extraction many approaches have been proposed. One of them is a rule-based method employed in (Ciravegna, 2001; Hazan and Andruszkiewicz, 2013). Support Vector Machines (SVM) classifier was applied to information extraction from web pages also (Andruszkiewicz and Nachyla, 2013). A variety of Conditional Random Fields (CRF) methods were widely used (Tang et al., 2008; Wu and Weld, 2010; Li et al., 2011; Rocktäschel et al., 2013; Wang and Feng, 2013; Andruszkiewicz and Nachyla, 2013; Cuong et al., 2015). Constrained CRF applied in (Xin et al., 2008) allows a miner for specifying constrains for extracted entities. Furthermore, Markov Logic Networks (MLNs) were used in information extraction from web pages (Andruszkiewicz and Nachyla, 2013).

6. Conclusions and Future Work

To sum up, we created the corpus of 943 annotated homepages of scientific conferences and make it publicly available. Moreover, we performed the experiments with single- and multi-token SVM and CRF for this set in order to set a baseline for this corpus.

In future work, we plan to apply other models, e.g., hierarchical CRF, MLNs, to obtain better results. Especially, we want to focus on *important dates* extraction by experimenting with different models and gathering more instances of these entity types. We also would like to extend our corpus by adding new conferences and annotations, e.g., chairs, committee members, in order to encourage researchers to make experiments on our corpus.

7. Bibliographical References

Andruszkiewicz, P. and Nachyla, B. (2013). Automatic extraction of profiles from web pages. In *Intelligent Tools for Building a Scientific Information Platform - Advanced Architectures and Solutions*, pages 415–431.

Califf, M. E. and Mooney, R. J. (1999). Relational learning of pattern-match rules for information extraction. In Jim Hendler et al., editors, *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA.*, pages 328–334. AAAI Press / The MIT Press.

Chang, C. and Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM TIST*, 2(3):27.

Ciravegna, F. (2001). $(LP)^2$, an adaptive algorithm for information extraction from web-related texts. In *Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*.

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

Cuong, N. V., Chandrasekaran, M. K., Kan, M., and Lee, W. S. (2015). Scholarly document information extraction using extensible features for efficient higher order semi-CRFs. In Paul Logasa Bogen II, et al., editors, *Proceedings of the 15th ACM/IEEE-CE Joint Conference on Digital Libraries, Knoxville, TN, USA, June 21-25, 2015*, pages 61–64. ACM.

Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, C. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Freitag, D. and McCallum, A. K. (1999). Information extraction with HMMs and shrinkage. In *In Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 31–36.

Hazan, R. and Andruszkiewicz, P. (2013). Home pages identification and information extraction in researcher profiling. In *Intelligent Tools for Building a Scientific Information Platform - Advanced Architectures and Solutions*, pages 41–51.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

Ireson, N., Ciravegna, F., Califf, M. E., Freitag, D., Kushmerick, N., and Lavelli, A. (2005). Evaluating machine learning for information extraction. In Luc De Raedt et al., editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 345–352. ACM.

Issertial, L. and Tsuji, H. (2011). Information extraction and ontology model for a 'call for paper' manager. In David Taniar, et al., editors, *iiWAS'2011 - The 13th International Conference on Information Integration and Web-based Applications and Services, 5-7 December 2011, Ho Chi Minh City, Vietnam*, pages 539–542. ACM.

Kenter, T. and Maynard, D. (2005). Using GATE as an Annotation Tool, January.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In Brian D. Davison, et al., editors, *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 441–450. ACM.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley et al., editors, *ICML*, pages 282–289. Morgan Kaufmann.

Lazarinis, F. (1998). Combining information retrieval with information extraction for efficient retrieval of calls for papers. In *20th Annual BCS-IRSG Colloquium on IR, Autrans, France. 25th-27th March 1998*, Workshops in Computing. BCS.

Li, Y., Jiang, J., Chieu, H. L., and Chai, K. M. A. (2011). Extracting relation descriptors with conditional random fields. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 392–400. The Association for Computer Linguistics.

McCallum, A., Schultz, K., and Singh, S. (2009). FACTORIE: probabilistic programming via imperatively defined factor graphs. In Yoshua Bengio, et al., editors, *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 1249–1257. Curran Associates, Inc.

Porter, M. F. (2001). Snowball: A language for stemming algorithms.

Rocktäschel, T., Huber, T., Weidlich, M., and Leser, U. (2013). Wbi-ner: The impact of domain-specific features on the performance of identifying and classifying mentions of drugs. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 356–363.

Schneider, K. (2006). Information extraction from calls for papers with conditional random fields and layout features. *Artif. Intell. Rev.*, 25(1-2):67–77.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). Arnetminer: extraction and mining of academic social networks. In Ying Li, et al., editors, *KDD*, pages 990–998. ACM.

Wang, G. and Feng, X. (2013). Tool wear state recognition based on linear chain conditional random field model. *Eng. Appl. of AI*, 26(4):1421–1427.

Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In Jan Hajic, et al., editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16,*

2010, Uppsala, Sweden, pages 118–127. The Association for Computer Linguistics.

Xin, X., Li, J., Tang, J., and Luo, Q. (2008). Academic conference homepage understanding using constrained hierarchical conditional random fields. In James G. Shanahan, et al., editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 1301–1310. ACM.