Domain Specific Features Driven Information Extraction from Web Pages of Scientific Conferences

Piotr Andruszkiewicz(✉) and Rafał Hazan

Institute 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 describe information extraction from web pages of scientific conferences. We enrich already known features with our new features specific for this domain and show their importance in the process of extracting information. Moreover, we investigate various data representation models, e.g., based on single tokens or sequences, in order to find the best configuration for the task in question and set up a new baseline over publicly available corpus.

1 Introduction

Up-to-date information about conferences plays a vital role in scientific life. Therefore methods for automatic collection of data on conferences, 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., are important for scientific community.

In order to gather data about conferences, one may extract interesting information from relevant resources. It is easy to obtain data from structured services like WikiCFP. However, regarding data from this kind of sources, there might be the lack of information or outdated information. 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. Calls For Papers (CFPs) have limited range of information, e.g., usually there is no information about sponsors. Moreover, this kind of service provides CFPs that are not updated while changes are made, e.g., submission date extensions. Homepages of conferences provide updated information but in an unstructured way. Due to that fact, the methods of information extraction from unstructured text/web resources need to be employed. To this end, in most cases supervised methods are used. These methods need an annotated data set that will be used for training, optimisation and testing.

Bearing in mind drawbacks of CFPs as a data source, we deal with information extraction from conference web pages. Being more specific, we investigate the already known and new domain specific features for information extraction and check how different models handle extraction of specific entity types. In our

A. Gelbukh (Ed.): CICLing 2017, LNCS 10761, pp. 405–417, 2018.
https://doi.org/10.1007/978-3-319-77113-7_32

experiments we use Support Vector Machine (SVM) and Conditional Random Fields (CRF) and combine them with different data representation models. We verify our statements on publicly available corpus of scientific conferences web pages and make a new reproducible baseline for this corpus.

The remainder of this paper is organised as follows: Sect. 2 presents related work. In Sect. 3 we describe the corpus we use. In Sects. 4 the proposed features are presented. The experimental results are presented in Sect. 5. Finally, Sect. 6 summarises the conclusions of the study and outlines avenues to explore in the future.

2 Related Works

Previous works in the field of information extraction from scientific conferences focused mostly on information extraction from CFPs using different approaches. Extracting information from CFPs has already mentioned drawbacks. In [13] a rule based method was employed to extract date and country from a CFP. A linear CRF was used in [16] 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. We use HTML sourcecode of web pages, including formatting. As in [16] only plain text was used, layout features were based on lines of text, indicating, e.g., first token in line or first line in the text. We take into account, for instance, hyperlinks, blocks, and formating. Thus, our data has much richer layout. In [8] a general platform for performing and assessing information extraction from workshop CFPs was described. In [9] authors focused also on information extraction from CFPs, including those 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 of a user. In contrast to aforementioned works [18] 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. Hence, we could not apply our approach to this set in order to compare the results. Furthermore, we could not recreate this system due to insufficient details in the paper.

In information extraction from documents of rich structure and plain text, many approaches have been proposed, regardless the domain of data. One of them is a rule-based method employed in [3,6]. A Support Vector Machines (SVM) classifier was also applied to extract information from web pages [1]. A variety of Conditional Random Fields (CRF) methods were widely used [1,17,18]. Furthermore, Markov Logic Networks (MLNs) were used for information extraction from web pages [1].

In order to verify the necessity of domain specific features and set a new baseline for publicly available corpus we focused on information extraction from conference web pages.

3 The Corpus

The corpus we use is, to the best of our knowledge, the only one publicly available corpus of annotated scientific conferences homepages. It contains 943 annotated homepages of scientific conferences (14794 including subpages). 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 are annotated: *name* and *abbreviation* of the conference, *place*, *dates* of the conference, *submission*, *notification*, *final version due* dates. We call the last three entities *important dates.* In this paper all mentioned types of entities are considered to be extracted. This corpus is available in public and can be found on the website http://ii.pw.edu.pl/~pandrusz/data/conferences.

4 Preprocessing and Features

Information extraction from web pages is a special case of information extraction, hence it requires specific techniques and approaches. We start the description of our approach from the preprocessing phase. Then we present group of features we developed. The described techniques and features are verified in models we build in order to find the best configuration for the given information extraction task.

4.1 Preprocessing

In the preprocessing phase we use Snowball stemmer [15] in order to reduce the number of features. Furthermore, we remove stopwords to reduce information noise in the data. We create our own stoplist by dividing words into two groups; namely, *far words* that are farther more than four words from the annotated entity in the data and *close words* that are closer than far words. We consider a word to be a stopword if it does not provide additional information and is in the far words group but not in the close group. The stoplist consists of 21095 words. Moreover, words which occur once or twice in the training set are also considered stopwords. This reduces words that come from wrongly parsed words or named entities that occur very rarely. This way of stoplist preparing reflects the specificity of the domain we are working with. Names of conferences often consist of words such as "the", "and", "on" that are commonly assumed to be stopwords. In this case we cannot remove them because we will not be able to extract a proper name of conferences.

Web pages often contain a lot of unnecessary information, e.g., advertisements, HTML code, menus, copyright notes, thus a specialised library can be used to clean an analysed web page. However, in the case of scientific conference web pages there are not many advertisements and unnecessary information. Hence, we use standard library, Boilerpipe [11], to extract a main article or paragraphs from a web page. We do not remove any other text from the web page to avoid removing important elements by mistake.

4.2 Features

In our approach we distinguish the following group of features: local, offset, layout, and dictionary features. Within these groups we enriched already known features with new features that to the best of our knowledge have not been used for information extraction before.

Local Features. Local features are calculated based on a current word we are analysing. The first and commonly used feature is a *word*. We do not create features for words from stoplist and those that contain nonalphabetic characters. Furthermore, we use part of speech (POS) tags for a current word provided by *Penn Pos Tagger* from *factorie* package [14]. Next feature is *short word* that is assigned with a value *true* if a word contains from 2 to 5 characters. This feature is designed for extraction of acronyms of conferences. 74% of conference's acronyms contain from 2 to 5 characters. *Shape of a word* is the next feature. The feature contains 'a' (for small letters), 'A' (for capital letters), and '1' (for numbers). If there are more than two the same characters in a row, the sequence is reduced to two the same characters. The example values for this feature are: AaaAA (WebET), Aaa (International), 1aa (5th), AA (NAACL), 11 (2016).

Last but not least is a *type of a word* feature. We distinguish eight types of words. *Date* represents whole dates that can be found on a web page. *Short phrase* is assigned to words that are part of sequence of length of one or two words (for more information about sequences please refer to Sect. 4.3, the example is the named entity with two words, for instance, Carl Brunto). *Long phrase* represents words of sequences that consist of at least three words. The reason behind the distinction between short and long phrases is that conference names are usually not short phrases but location of conferences usually are. Other types are: *Number* - assigned for numbers, e.g., 23, 3rd; *acronyms* are words of the following shapes: AA, AaaAaa AaaAA, AA1AA, AAaa, AaAA, AAa, AAaAA; *punctuation marks*, *special char* - all nonalphanumeric chars that are not punctuation marks, e.g., @, *.

All other words are of the type *standard word*. They represent words that probably do not contain interesting information we want to extract.

Table 1. The distribution of the interesting entities over blocks of a web page.

Entity	Name	Abbrev.	Place	Date	Submission	Notification	Final ver. due	Other
Head title	0.18	0.11	0.03	0.03	0.00	0.00	0.00	0.00
Title/subtitle	0.23	0.09	0.04	0.08	0.03	0.02	0.02	0.01
Paragraph	0.50	0.60	0.68	0.61	0.36	0.29	0.25	0.42
Table/list	0.05	0.14	0.16	0.21	0.44	0.51	0.51	0.14
Other	0.40	0.60	0.09	0.07	0.24	0.18	0.22	0.44

Offset Features. *Predecessor* represents features based on the word that precedes the current word. We take into account only one predecessor and *type of a word* feature. *Successor* is calculated for a word that follows the current word. We consider one word ahead and *type of a word* feature.

Sections with important dates of a conference are often organised with lists or tables. Though it is a convenient way for a human, machine learning algorithms poorly deal with learning patterns that occur on scientific conference web pages, because dates are placed on the right, left and even above and below the description of a date. In order to ease the process of learning, we bring into being *date surrounding words* features that extract the description of a given date in a way presented in Algorithm 1. Words returned by the aforementioned algorithm are used to create features for a current word, however, only for dates in order not to increase the number of features too much.

Algorithm 1. Extracting words surrounding a date

```
Data: a list item or a table cell // input text with a date
Result: a description of a date // words surrounding a date

if a date is followed by a semicolon then
    return up to six words after a date
end
if a date is preceded by a semicolon then
    return up to six words before a date
end
if a date is in a short (less than 100 words) list item or a table cell then
    return up to six words before and up to six words after a date that are
    within a list item or a table cell
end
else
    return up to six words before a date
end

General conditions that need to be met:
  – returned words must come from the same sentence as a date,
  – if a returned sequence of words contains a different date then choose a
    subsequence that starts from the first word and ends at the word before the
    first date in a sequence.
```

Layout Features. *Emphasised* feature indicates words that are modified by the following HTML tags: STRONG, B, U, and FONT which means that they are bold, underlined, or use different fonts. The underlined words are more often dates of a conference, however, names of conferences and abbreviations do not correlate with use of aforementioned HTML tags.

Hyperlink feature distinguishes words that are presented as links (A tag). Contrary to the first impression this feature is a good indicator of not being the important information to extract in our case; that is, correlation shows that hyperlinks more often lead to other conferences.

Block feature indicates a block a word belongs to. A separate value is assigned for each block. Considered blocks are head title, title and subtitle, paragraph, table, and list. Table 1 shows the distribution of the entities of our interest over blocks on a web page. Names and abbreviations of conferences, locations, and date occur mostly in paragraphs. Names and abbreviations are placed also in head title and title/subtitle. Dates of submission, notification and so on usually are provided in tables and lists, however, paragraphs also carry that information.

Paragraph number feature indicates the number of a paragraph a word belongs to. We count only the first 6 paragraphs as more than half of interesting entities are contained in these paragraphs according to the corpus. This feature helps in detection of conference names and abbreviations, dates and locations of conferences because as the corpus confirms these entities often occur at the beginning of a web page. The important dates usually occur further in a web page.

Entities we are looking for can be found on one of the subpages of the main conference web page. Thus, we add subpages to the training data, however, we restrict subpages to those that can be accessed through links with the following names: index, home, important dates, call for papers, registration. Furthermore, each word from subpage gets *subpage* feature that contains anchor text, e.g., SUB=home, SUB=index.

Dictionary Features. Detection that word(s) represent a location is helpful for conference location extraction. Hence, we used gazetter from ANNIE module of GATE [10] to add location names from the corpus. Each location found in a text generates a *location* feature, LOC=true. Moreover, each country gets feature COUNTRY=true and city CITY=true.

Out of dictionary feature indicates that a current word has not been found in our custom dictionary of English words that contains 112505 words. This feature is intended to help in abbreviations extraction as the percentage of words not found in the dictionary is the highest for conference abbreviations (0.89). The percentage for location (0.75) is also high, hence it is suggested to a model by this feature (for name it is only 0.23 and 0.14 for other words).

Promising surrounding words feature indicates whether there is at least one word from a given dictionary in a sentence a current word belongs to. We use dictionaries for the following types of entities: name and abbreviation of conference, place and date created based on the most frequent words that occur in sentences that contain an important entity. The dictionaries are not mutually exclusive, hence the *promising surrounding words* feature indicates whether it is an important entity rather than an entity is of a specific type.

4.3 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 a base object used by a model and assigned 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 if 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 a trivial one. 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 any of the chars for this set: ‘,-:’. 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; (2) calculate features for each word separately and use all features; (3) combine features for

Table 2. The importance of features groups for types of entities extraction.

Features	Measure	Name	Abbrev.	Place	Date	Submission	Notification	Final ver. due
All	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
No local features	Precision	0.10	0.51	0.72	0.58	0.64	0.47	0.67
	Recall	0.09	0.58	0.60	0.23	0.41	0.28	0.43
	F1	0.09	0.55	0.66	0.33	0.50	0.35	0.52
No offset features	Precision	0.36	0.73	0.69	0.67	-	-	-
	Recall	0.30	0.64	0.57	0.68	0.00	0.00	0.00
	F1	0.33	0.68	0.62	0.67	0.00	0.00	0.00
No layout features	Precision	0.33	0.63	0.68	0.79	0.62	0.53	0.67
	Recall	0.22	0.45	0.45	0.65	0.55	0.44	0.54
	F1	0.26	0.52	0.54	0.71	0.58	0.48	0.60
No dict. features	Precision	0.35	0.77	0.70	0.70	0.61	0.56	0.70
	Recall	0.32	0.72	0.46	0.67	0.52	0.44	0.57
	F1	0.33	0.74	0.55	0.69	0.00	0.00	0.58

all words into one feature. For example, feature *word* is calculated according to the second approach and, e.g., 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.

5 Experiments

In our experiments we divide the corpus into training and test sets according to the proportion of 70/30. For the SVM model the training set is used to perform cross validation in order to find the best parameters, then the model is trained on the whole training set using these parameters.

For a web page, as an extracted entity we choose the only one instance of 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.

Table 3. The results of entities extraction with regard to different models (the best F1 results marked in bold).

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

5.1 Importance of Features

In our first group of experiments we verify 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 the groups of features contain sparse features, 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 we perform experiments with all groups of features but one. The results of the experiments with SVM (Table 2) show that the most important are *local* features. Lack of them causes the highest drop in accuracy of the results (more than 20 p.p. for *name* and *abbreviation*, almost 50 p.p. for *date* in F1). These features generate almost half of feature functions. This group contains *type of a word* feature and its absence makes extraction task harder for each type of interesting entities. Furthermore, lack of *shape of a word* and *short word* features decreases accuracy for abbreviation extraction. Only *place* noticed slightly drop of accuracy.

Lack of *offset* features reduces mostly the accuracy of conference *date*, about 13 p.p. in terms of F1, and *important dates* are not discovered at all. It is due to lack of *date surrounding words* features that characterise important dates well. This group generates high number of feature functions also.

Layout features help in extraction of *name* and *abbreviation* of a conference. They are also important for *place* and *date* of a conference, however, to lower extend. Within this group of features *block* and *paragraph number* features are the most important ones. These entities often occur in head title. They may be provided also in a title or a subtitle of a web page. If these entities are missing in aforementioned block, it is almost sure that they appear in the first or in a few first paragraphs of a web page. This information is carried over by mentioned features.

As we expected *dictionary* features play the most important role for *place* detection as a *location* feature is a key for this entity type.

To sum up, each group of features carries some information that is important (at least for one of) interesting entity types. Thus, we could say that it is crucial to prepare features that are specific for a given domain. As we have shown, lack of some features may reduce the accuracy for some entity types to zero, for instance, the lack of *offset* features for *important dates*. In the domain of web pages of scientific conferences *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, that is necessary for *important dates* extraction. *Layout* features generate important features functions that inform about a place within a web page a given word is located. They help in 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.

5.2 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 all mentioned groups of features and preprocessing described in Sect. 4.1.

SVM Model. As a base model we use Support Vector Machine (SVM) [4] with linear and radial basis function (RBF) kernel that is defined as follows:

$K(x, y) = e^{-\xi||x-y||^2}$. We use LibSVM implementation [2]. For multiclass classification we employ one versus the rest approach [5]. 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 the first model assigns a label to each single token independently and mentioned entities consists of several tokens. We try to help 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 p.p. decrease in F1 for *abbreviation* detection comparing to the single token SVM.

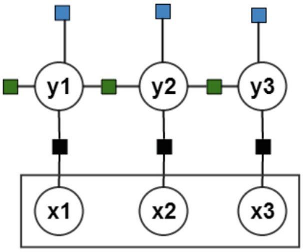

Fig. 1. Linear CRF structure.

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 [7].

CRF Model. In the experiments, we also use Conditional Random Fields, CRF [12]. Figure 1 presents the structure of CRF model which is a linear one with three different templates of factors. 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. Equation 1 shows the formula of our CRF model, where $Z(\boldsymbol{x})$ is a normalisation factor.

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

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\,(\sum_{j=1}^{n}(\sum_{i=1}^{m_1} \alpha_{1,i} f_{1,i}(y_j)+ \sum_{i=1}^{m_2} \alpha_{2,i} f_{2,i}(y_{j-1}, y_j) + \sum_{i=1}^{m_3} \alpha_{3,i} f_{3,i}(y_j, \boldsymbol{x}, j))) \tag{1}$$

In our experiments we used CRF that operates on single tokens (Lin. CRF in Table 3) and sequences (Lin. CRF seq. in Table 3). Single tokens CRF 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 of label (SVM lacks this feature). However, for entities that do not consist of several consecutive words we have not observed the improvement in the results; on 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 discovers them on a 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, like for SVM.

In case of *name* sequences discovery, which is not so perfect as for *important dates*, 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, dates are extracted better with models based on sequences than single tokens. For *place* the winner is SVM on both single tokens and sequences (only 1 p.p. difference), however, all other models are not worse than 8 p.p. in terms of F1. The single token models outperforms sequence models for *name* and *abbreviation*. The single token SVM obtains the best results for *abbreviation*, however, the sequence CRF is not far behind (0.82 vs. 0.80 in terms of F1). Furthermore, the results of all models in *abbreviation* extraction are within the difference of 6 p.p., hence, results from all models do not differ much. The reason behind may be that *abbreviation* is a single token entity and sequence models do not leverage their properties in this case. Surprisingly, *name* entity is handled the best with the single token CRF, despite having more than one token. This is probably due to lower accuracy of the algorithm that discovers conference name token sequences for the sequence models compared to date discovery (*dates* are extracted the best with sequence models).

Concluding the analysis of the obtained results, different models may be used for specific entity types in order to achieve the best cumulative results.

6 Conclusions and Future Works

In this paper we investigated information extraction from scientific conference web pages by verifying the applicability of different types of features and various models.

We designed different groups of features and verified their importance in this task. Based on the empirical results obtained on publicly available corpus we state that domain specific features are necessary for correct information extraction. Additionally entity type specific features are also necessary in order to obtain good results.

Despite having a broad range of features, the considered models (algorithms, representations of base objects for algorithms) achieve different results for different entity types. Thus, it is beneficial to apply specific models for specific entities.

Moreover, with help of our new features we set new baselines values of precision, recall, and F1 for information extraction from a publicly available corpus of scientific conference web pages.

In future work we plan to create a model for multi-token sequence detection and incorporate it in our models. We would also like to apply other models, e.g., MLNs, hierarchical CRF, to obtain better results.

References

1. Andruszkiewicz, P., Nachyla, B.: Automatic extraction of profiles from web pages. In: Bembenik, R., Skonieczny, L., Rybinski, H., Kryszkiewicz, M., Niezgodka, M. (eds.) Intelligent Tools for Building a Scientific Information Platform - Advanced Architectures and Solutions, pp. 415–431. Springer, Heidelberg (2013). http://dx.doi.org/10.1007/978-3-642-35647-6_25
2. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. ACM TIST **2**(3), 27 (2011). http://doi.acm.org/10.1145/1961189.1961199
3. Ciravegna, F.: $(LP)^2$, an adaptive algorithm for information extraction from web-related texts. In: Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining (2001). http://citeseer.ist.psu.edu/481342.html
4. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
5. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008). http://doi.acm.org/10.1145/1390681.1442794
6. Hazan, R., Andruszkiewicz, P.: Home pages identification and information extraction in researcher profiling. In: Intelligent Tools for Building a Scientific Information Platform - Advanced Architectures and Solutions, pp. 41–51 (2013). http://dx.doi.org/10.1007/978-3-642-35647-6_4
7. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification (2003)
8. Ireson, N., Ciravegna, F., Califf, M.E., Freitag, D., Kushmerick, N., Lavelli, A.: Evaluating machine learning for information extraction. In: Raedt, L.D., Wrobel, S. (eds.) Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, 7–11 August 2005. ACM International Conference Proceeding Series, vol. 119, pp. 345–352. ACM (2005). http://doi.acm.org/10.1145/1102351.1102395

9. Issertial, L., Tsuji, H.: Information extraction and ontology model for a ‘call for paper’ manager. In: Taniar, D., Pardede, E., Nguyen, H., Rahayu, J.W., Khalil, I. (eds.) iiWAS 2011 - The 13th International Conference on Information Integration and Web-based Applications and Services, 5–7 December 2011, Ho Chi Minh City, Vietnam, pp. 539–542. ACM (2011). http://doi.acm.org/10.1145/2095536.2095650
10. Kenter, T., Maynard, D.: Using GATE as an annotation tool, January 2005. http://gate.ac.uk/sale/am/annotationmanual.pdf
11. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Davison, B.D., Suel, T., Craswell, N., Liu, B. (eds.) Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, 4–6 February 2010, pp. 441–450. ACM (2010). http://doi.acm.org/10.1145/1718487.1718542
12. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Brodley, C.E., Danyluk, A.P. (eds.) ICML, pp. 282–289. Morgan Kaufmann (2001)
13. Lazarinis, F.: Combining information retrieval with information extraction for efficient retrieval of calls for papers. In: 20th Annual BCS-IRSG Colloquium on IR, Autrans, France, 25–27 March 1998. Workshops in Computing, BCS (1998). http://ewic.bcs.org/content/ConWebDoc/4410
14. McCallum, A., Schultz, K., Singh, S.: FACTORIE: probabilistic programming via imperatively defined factor graphs. In: Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I., Culotta, A. (eds.) 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, pp. 1249–1257. Curran Associates, Inc. (2009)
15. Porter, M.F.: Snowball: a language for stemming algorithms (2001)
16. Schneider, K.: Information extraction from calls for papers with conditional random fields and layout features. Artif. Intell. Rev. **25**(1–2), 67–77 (2006). http://dx.doi.org/10.1007/s10462-007-9019-4
17. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: ArnetMiner: extraction and mining of academic social networks. In: Li, Y., Liu, B., Sarawagi, S. (eds.) KDD, pp. 990–998. ACM (2008)
18. Xin, X., Li, J., Tang, J., Luo, Q.: Academic conference homepage understanding using constrained hierarchical conditional random fields. In: Shanahan, J.G. et al. (eds.) Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, 26–30 October 2008, pp. 1301–1310. ACM (2008). http://doi.acm.org/10.1145/1458082.1458254