INDIRECT SPATIAL DATA EXTRACTION FROM WEB DOCUMENTS

Dimitar Blagoev, George Totkov, Milena Staneva,
Krassimira Ivanova, Krassimir Markov, Peter Stanchev

Abstract: An approach for indirect spatial data extraction by learning restricted finite state automata from web documents created using Bulgarian language are outlined in the paper. It uses heuristics to generalize initial finite-state automata that recognizes only the positive examples and nothing else into automata that recognizes as larger language as possible without extracting any non-positive examples from the training data set. The learning method, program realization and experiments are presented. The investigation is carried out in accordance and following the rules of EU INSPIRE Network.

Keywords: Automatic Data Extraction, Restricted Finite State Automata, Web Documents, Indirect Spatial Data, INSPIRE network.

ACM Classification Keywords: H.2.8 Database Applications - Data mining; F.1.1 Models of Computation – Finite State Automata

Introduction

“Spatial data” is data with a *direct* or *indirect* reference to a specific location or geographic area [INSPIRE-DSM, 2007]. Our attention in this paper is given to the indirect references included in the text documents written in Bulgarian language.

The indirect reference to a specific location or geographic area may be of different types and formats. Because of this it is difficult to propose a common classification of such information. In the same time, one of the main characteristic of indirect references is the address, i.e. a description of the interconnection of the data with the specific location or geographic area [INSPIRE-DSAD, 2008]. Usually this is a text with common structure – location of properties based on address identifiers, usually by road name, house number, postal code, etc.

In everyday practice there are many kinds of documents containing indirect references to a specific locations or geographic areas. The kernel problem is that EU member countries use different languages and national standards for different types of indirectly given references. The automatic extraction of the references is very important for processing such documents in the INSPIRE network.

In recent years multiple machine learning approaches have been proposed for information extraction [Li et al, 2008]. A large class of entity extraction tasks can be accomplished by the use of carefully constructed regular expressions. Examples of entities amenable to such extractions include e-mail addresses, software names (web collections), credit card numbers, social security numbers (e-mail compliance), gene and protein names (bioinformatics), etc. With a few notable exceptions, there has been very little work in reducing this human effort through the use of automatic learning techniques.

In the context of information extraction, prior work has concentrated primarily on learning regular expressions over relatively small alphabet sizes and usually learning of regular expressions is done over tagged tokens produced by other text-processing steps such as POS tagging, morphological analysis, and gazetteer matching [Ciravegna, 2001].

[Rozenfield et al, 2008] propose approach, which use the immense amount of unlabeled text in the web documents in order to create large lists of entities and relations. Based on this approach the system SRES is a self-supervised web relation extraction system that learns powerful extraction patterns from unlabeled text, using short descriptions of the target relations and their attributes.

The proposed in [Li et al, 2008] learning algorithm ReLIE takes as input not just labeled examples but also an initial regular expression, which provides a natural mechanism for a domain expert to provide domain knowledge about the structure of the entity being extracted and meaningfully restriction of the space of output regular expressions.

In 2004 a team of Prof. William Cohen from Carnegie Mellon University starts creating collection of classes for storing text, annotating text, and learning to extract entities and categorize text called MinorThird [Cohen, 2004]. It contains a number of methods for learning to extract and label spans from a document, or learning to classify spans (based on their content or context within a document). The creating of such collections is a useful tool not only for the particular investigation support, but also for creating common notion for the area as a whole.

An approach for indirect spatial information extraction by learning restricted finite state automata from marked web documents created using Bulgarian language is outlined in the paper. We use heuristics to generalize initial finite-state automata that recognizes only the positive examples and nothing else into automata that recognizes as larger language as possible without extracting any non-positive examples from the training data set.

The proposed approach is a good base for building system from the class of Semi-Automated Interactive Learning (SAIL) systems [IBM, 2009]. In the next chapters the INSPIRE network as possible practical area, the proposed approach, program realization and experiments are presented. Finally, the conclusions and steps for feature work are outlined.

The INSPIRE Network

Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 for establishing an Infrastructure for Spatial Information in the European Community (INSPIRE), was published in the Official Journal of the European Union on 25 April 2007 and was entered into force on 15 May 2007 [INSPIRE Directive, 2007]. The main goal of the Directive is to establish a new common approach for processing the spatial data in all EU member countries.

A simplified view to the processing of data today is shown in the Figure 1. In most cases, each EU member state uses input data according to different, often undocumented or ill-documented data specifications and uses different methods to process the input data to produce more or less similar information relevant for policies within the Community [INSPIRE-DSM, 2007]. For instance two different states "A" and "B" have theirs own specific data specifications and data sets and their own processing methods.

The methodology described in the Directive aims for a better understanding of the common user requirements for data in INSPIRE. It focuses on the development of harmonized data specifications for the input data. This way all input data from the different member states will follow the same data specifications and the same processing steps to derive the information. The input data in the member states and their maintenance procedures will typically be more-or-less the same prior to INSPIRE, but in addition the data will be provided by the network services of the member states following the harmonized data specifications [INSPIRE-DSM, 2007]. The updated schema based on the proposed methodology for two states "A" and "B" is shown on Figure 2.

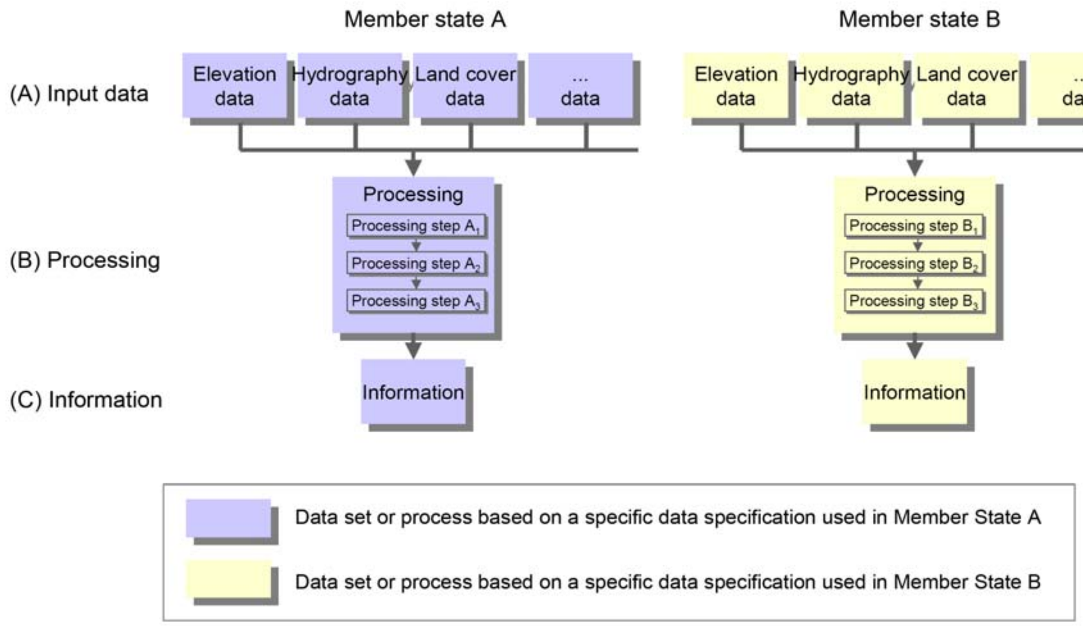


Figure 1. Current situation is "Data stovepipes"

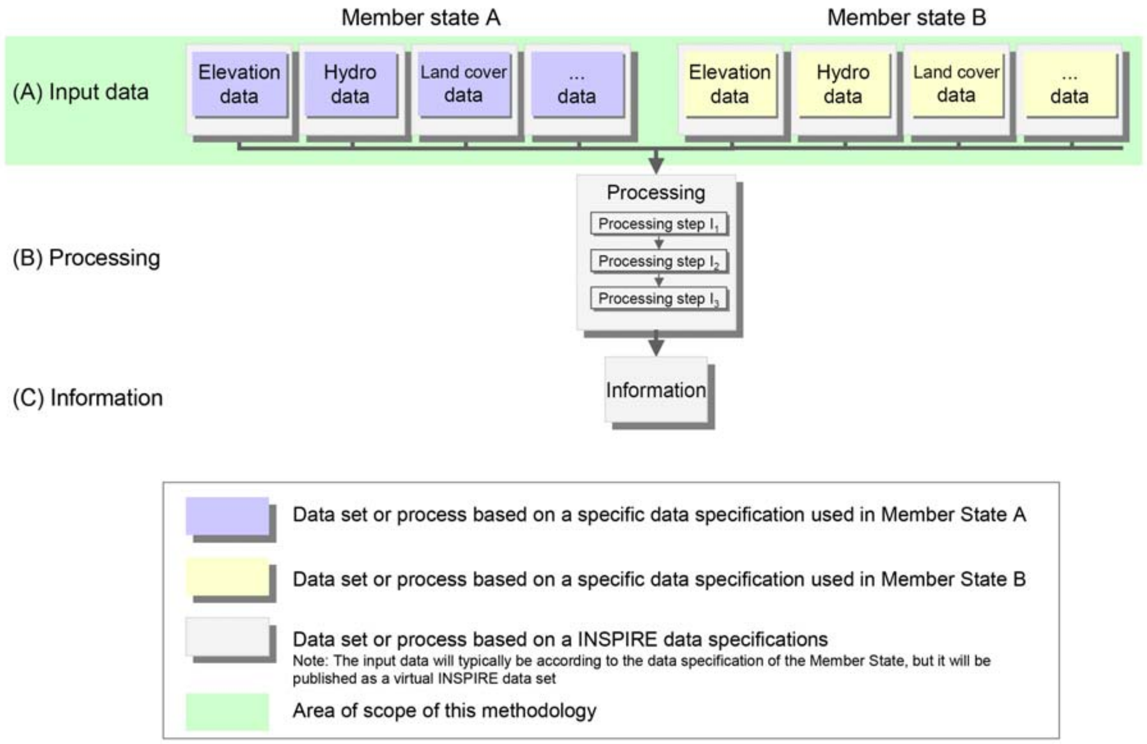

Member state A
Member state B
(A) Input data
(B) Processing
(C) Information
Elevation data
Hydro data
Land cover data
... data
Processing
Processing step I_1
Processing step I_2
Processing step I_3
Information
Data set or process based on a specific data specification used in Member State A
Data set or process based on a specific data specification used in Member State B
Data set or process based on a INSPIRE data specifications
Note: The input data will typically be according to the data specification of the Member State, but it will be published as a virtual INSPIRE data set
Area of scope of this methodology

Figure 2. Target situation: Harmonized data views, eliminating data stovepipes

INSPIRE should be based on the infrastructures for spatial information that are created by the member states and are designed to ensure that:

– spatial data are stored, made available and maintained at the most appropriate level;

- it is possible to combine spatial data from different sources across the European Community in a consistent way and share them between several users and applications;
- it is possible for spatial data collected at one level of public authority to be shared between other public authorities;
- spatial data are made available under conditions which do not unduly restrict their extensive use;
- it is easy to discover available spatial data, to evaluate their suitability for the purpose and to know the conditions applicable to their use.

For these reasons, the Directive focuses in particular on five key areas:

- metadata;
- the interoperability and harmonization of spatial data and services for selected themes (as described in Annexes I, II, III of the [INSPIRE Directive,2007]);
- network services and technologies;
- measures on sharing spatial data and services;
- coordination and monitoring measures.

INSPIRE lays down the legal framework for the establishment and operation of an Infrastructure for Spatial Information in Europe. The purpose of such an infrastructure is to assist policy-making in relation to policies that may have a direct or indirect impact on the environment. *"Infrastructure for spatial information"* means metadata, spatial data sets and spatial data services; network services and technologies; agreements on sharing, access and use; and coordination and monitoring mechanisms, processes and procedures, established, operated or made available in accordance with the Directive [INSPIRE Directive, 2007]:

Every spatial object in a spatial data set needs to be described by a data specification specifying the semantics and the characteristics of the types of spatial objects in the data set. The spatial object types provide a classification of the spatial objects and determine among other information the properties that any spatial object may have (be they thematic, spatial, temporal, a coverage function, etc.) as well as known constraints (e.g. the coordinate reference systems that may be used in spatial data sets). This information is captured in an application schema using a conceptual schema language, which is a part of the data specification. As a result, a data specification provides the necessary information to enable and facilitate the interpretation of spatial data by an application [INSPIRE-TAO, 2007].

The logical schema of the spatial data set may and will often differ from the specification of the spatial object types in the data specification. In this case, and in the context of real-time transformation, a service will transform queries and data between the logical schema of the spatial data set and the published INSPIRE application schema on-the-fly. This transformation can be performed by the download service offering access to the data set or a separate transformation service.

*The main goal of INSPIRE is the "**Interoperability**"* which means the possibility for spatial data sets to be combined, and for services to interact, without repetitive manual intervention, in such a way that the result is coherent and the added value of the data sets and services is enhanced.

One important aspect of this process is the automatic extraction of spatial data and creating the corresponded metadata.

Data Extraction by Learning Restricted Finite State Automata

The approach for indirect spatial information extraction by learning restricted finite state automata from marked web documents contains four main steps:

1. Setting up the hierarchical structure of the data to be extracted. Every element and sub-element which is to be identified has to be specified. The data structure is expressed as a tree of elements and their sub-elements.
2. Scanning and manual tagging the initial documents for the required information.
3. Extracting the examples for the different elements and building an initial parsing grammar.
4. Data extracting from new documents. The user can continue to improve the accuracy of the results by manually correcting the annotations for a particular document and add it to the learning set.

The building of the parsing grammar consists of two sub-steps:

a) combining all positive examples;

b) generalizing the resulting tree into restricted finite state automata.

At the sub-step a) all marked instances of the structured data are flattened in strings containing the text of the main element with special symbols marking the beginnings and ends of the sub-elements and the HTML tags in the case when the text is a web document. Then all the flattened strings from all documents are combined in a single tree. This tree is then used as the initial finite state automata. It recognizes all learned positive examples without misrecognizing negative ones.

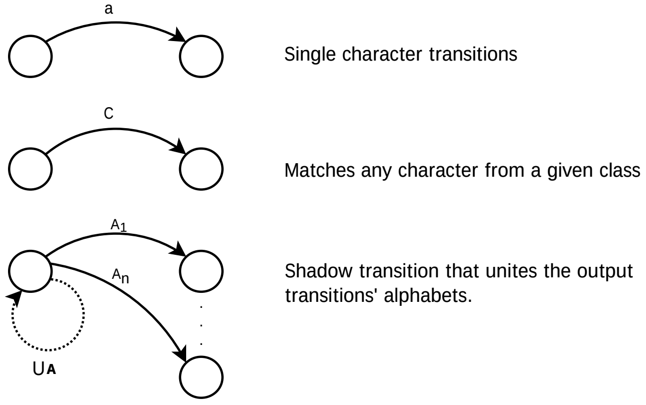

a
Single character transitions
C
Matches any character from a given class
A1
An
UA
Shadow transition that unites the output transitions' alphabets.

Figure 3. Elements of the restricted finite state automata [Baltes, 1992]

The sub-step b) is the generalization of the automata using heuristic methods for combining states and extrapolating the transitions' characters into predefined alphabets. After each generalization the automata is checked for consistency by re-scanning the learning texts and if the extracted data differs from the initial (manually annotated) data the modification is rolled back. There are many ways in which the finite state automata can be generalized [Baltes, 1992]. To prevent the computational complications that arise from this condition we use restricted finite state automata. The building elements that are used in these automata are (Figure 3):

- single character transition;
- matching any character from a given class;
- shadow transition that unites the output transitions' alphabets.

In addition to the automata generalization heuristics described later in the section the generalizator employs the use of a custom character class list. The class list specifies what characters belong to a given class and how many of them have to be present in a state's output transitions before class generalization is attempted. Table 1 shows one sample list which includes classes for both English and Bulgarian letters.

Table 1: Sample character class list

Min	Characters	Class
3	abcdefghijklmnopqrstuvwxyz	English lowercase
3	ABCDEFGHIJKLMNOPQRSTUVWXYZ	English capitals
3	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ	English
3	абвгдежзийклмнопрстуфхцчшщъьюя	Bulgarian lowercase
3	АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЮЯ	Bulgarian capitals
3	абвгдежзийклмнопрстуфхцчшщъьюяАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЮЯ	Bulgarian
1	0123456789	Digits
1	\b \t \n \r	White space characters
1	' " “ „ ” « » ‘ ’ ,	Quotes

The generalization algorithm (Figure 4) in sub-step b) is done in the following way:

1. Class generalization (top-down) which tries to generalize as much as possible output transition characters for a given state into classes;
2. State merging (bottom-up) with character comparison tries to merge a state with one of its next possible states if the two states have identical characters and classes on their output transitions. If it is successful, the two states are merged into a single state and a shadow transition is added over the union of the other output transitions' alphabets. Further testing is made to find the upper repetition limit for the newly formed state;
3. State merging (bottom-up) without character comparison, essentially same as above except it does not require two states to have comparable characters and classes on their output transitions. If it is necessary, character transitions are merged with class transitions. This operation is more prone to making erroneous generalizations or one that block the further generalization of upper states therefore it is performed after the previous generalizations;
4. Character and class merging (bottom-up) tries to merge a character transition in a given state with a class or another character transition in the same state resulting in a transition over a new class which was not predefined in the classes list;

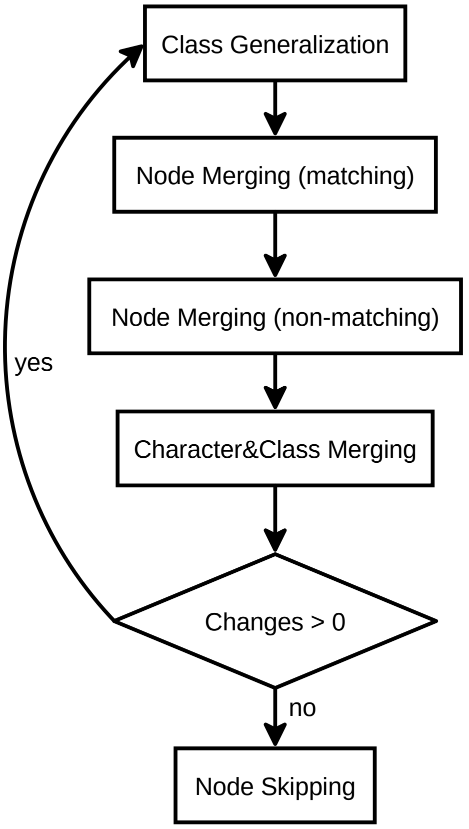


Figure 4. Generalization algorithm

5. State skipping (bottom-up) which tests if all output transitions on a given state can be skipped thereby advancing onto all sub-states without matching any of the transitions. Every output transition to another state is complemented with an epsilon transition (one that matches the zero-length string) to the same sub-state.

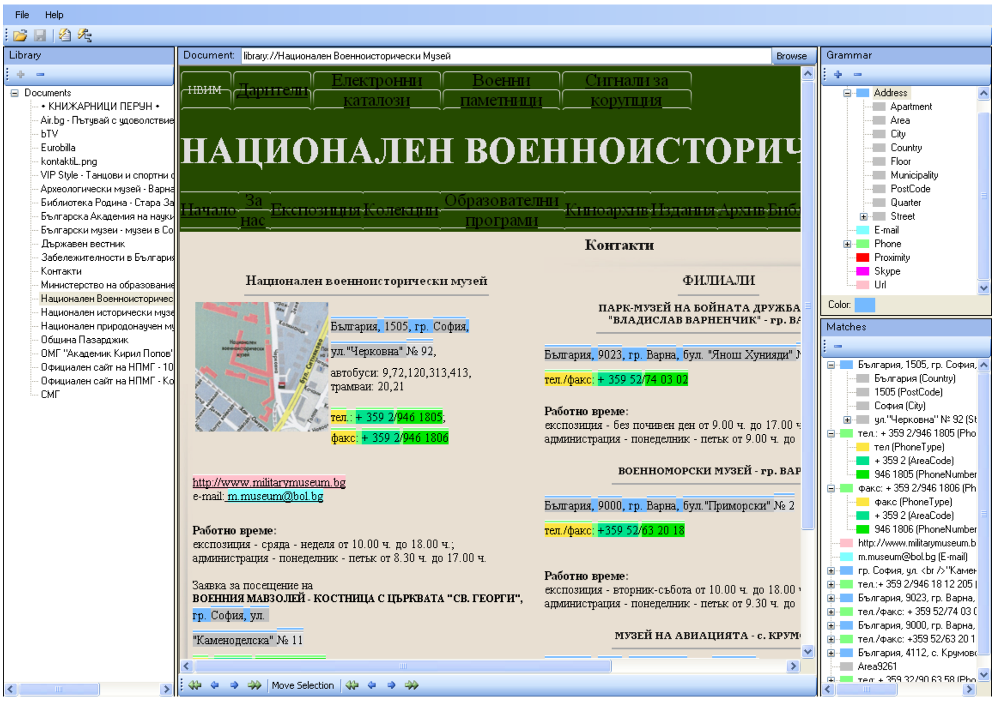
After every change of the automata a test run is performed with the new automata over the learning data set. If the result differs from the initial ones the attempted transformation is rolled back. All generalization and merging steps (without state skipping) are repeated until there are no more states which can be merged and/or generalized.

Once the module's learning phase is complete a parsing grammar is being generated. This grammar employs regular expressions for data extraction and generates structured XML output containing all elements found in the parsed documents. The sub-elements of the hierarchical data structure are encoded as named groups in the regular expressions. This grammar can then be used for performing background batch processing on a large number of documents or to analyze the produced regular expressions and make inferences for the structure of the elements of interest.

Program Realization

The presented algorithm has been realized in the experimental system InDES. The system contains two separate modules: a graphical user interface (GUI) and a command-line learning and extracting module. The GUI is developed in C#.NET and employs the embedded Internet Explorer browser component to display the web documents. The learner and extractor are written in C++ for increased performance and smaller memory footprint. Both modules use the same html preprocessing routine for cleansing the given web documents. The cleansing's purpose is to normalize or eliminate characters in the input document without changing the structure of the contained information or the way it appears on the screen. This includes but is not limited to Unicode character normalization where explicit character codes are replaced with their respective characters and JavaScript removal (since the current version of the system does not execute JavaScript prior to learning or extracting).

A screenshot of the GUI with a loaded web document is given on Figure 5. In the left, the sub-screen for selecting the web documents contain some already connected web sites and corresponded documents. At the right the generated grammar and founded matches are shown. In the center of the screen the current document with market texts is presented.



Documents
- • КНИЖАРНИЦИ ПЕРУН •
- Air.bg - Пътувай с удоволствие
- bTV
- Eurobilla
- kontaktiL.png
- VIP Style - Танцови и спортни
- Археологически музей - Варна
- Библиотека Родина - Стара За
- Българска Академия на науки
- Български музеи - музеи в Со
- Държавен вестник
- Забележителности в България
- Контакти
- Министерство на образование
- Национален Военноисторичес
- Национален исторически музе
- Национален природонаучен му
- Община Пазарджик
- ОМГ "Академик Кирил Попов"
- Официален сайт на НПМГ - 10
- Официален сайт на НПМГ - Ко
- СМГ

Document: library://Национален Военноисторически Музей

НВИМ Дарители Електронни каталози Военни паметници Сигнали за корупция

НАЦИОНАЛЕН ВОЕННОИСТОРИЧ

Начало За нас Експозиция Колекции Образователни програми Киноархив Издания Архив Библ

Контакти

Национален военноисторически музей

България, 1505, гр. София,
ул."Черковна" № 92,
автобуси: 9,72,120,313,413,
трамваи: 20,21

тел.: + 359 2/946 1805;
факс: + 359 2/946 1806

http://www.militarymuseum.bg
e-mail: m.museum@bol.bg

Работно време:
експозиция - сряда - неделя от 10.00 ч. до 18.00 ч.;
администрация - понеделник - петък от 8.30 ч. до 17.00 ч.

Заявка за посещение на
ВОЕННИЯ МАВЗОЛЕЙ - КОСТНИЦА С ЦЪРКВАТА "СВ. ГЕОРГИ",
гр. София, ул.
"Каменоделска" № 11

ФИЛИАЛИ

ПАРК-МУЗЕЙ НА БОЙНАТА ДРУЖБА "ВЛАДИСЛАВ ВАРНЕНЧИК" - гр. ВА

България, 9023, гр. Варна, бул. "Янош Хунияди" №
тел./факс: + 359 52/74 03 02

Работно време:
експозиция - без почивен ден от 9.00 ч. до 17.00 ч
администрация - понеделник - петък от 9.00 ч. до

ВОЕННОМОРСКИ МУЗЕЙ - гр. ВАР

България, 9000, гр. Варна, бул."Приморски" № 2
тел./факс: +359 52/63 20 18

Работно време:
експозиция - вторник-събота от 10.00 ч. до 18.00 ч
администрация - понеделник - петък от 9.30 ч. до

МУЗЕЙ НА АВИАЦИЯТА - с. КРУМ

Grammar
- Address
  - Apartment
  - Area
  - City
  - Country
  - Floor
  - Municipality
  - PostCode
  - Quarter
  - Street
- E-mail
- Phone
- Proximity
- Skype
- Url

Matches
- България, 1505, гр. София,
  - България (Country)
  - 1505 (PostCode)
  - София (City)
  - ул."Черковна" № 92 (St
- тел.: + 359 2/946 1805 (Pho
  - тел (PhoneType)
  - + 359 2 (AreaCode)
  - 946 1805 (PhoneNumber
- факс: + 359 2/946 1806 (Ph
  - факс (PhoneType)
  - + 359 2 (AreaCode)
  - 946 1806 (PhoneNumber
- http://www.militarymuseum.b
- m.museum@bol.bg (E-mail)
- гр. София, ул. <br />"Камен
- тел.:+ 359 2/946 18 12 205 (
- България, 9023, гр. Варна,
- тел./факс: + 359 52/74 03 0
- България, 9000, гр. Варна,
- тел./факс: +359 52/63 20 1
- България, 4112, с. Крумово
- Area9261
- тел: + 359 32/90 63 58 (Pho

Figure 5. Screenshot of the program system InDES

Experiments

We have made several types of experiments over different web documents in Bulgarian language for extracting elements such as addresses, phones and e-mails from randomly chosen companies' and social institutions' web pages containing contact information. To test the ability of the proposed method to extract new information we used a set of 100 pre-marked web documents in Bulgarian language for three types of elements: addresses, phones and e-mails with corresponded sub-elements. The reason for such choice is that main way for representing the indirect space information is using the addresses [INSPIRE-DSAD, 2008]. In other words, the extracting of the address is the first step for the processing the indirect space information. This information may contain different elements, represented by text sequences which are connected to any specific location or geographic area by the addresses. These elements need to be extracted, too. This means that the system need to have possibility to extract addresses as well the other types of elements from the given thematic area. To simplify the experiments, the phone numbers and e-mails are taken as such elements.

The experiments were provided following the steps of the proposed approach.

At first step the hierarchical structure of the data to be extracted was set up as it is shown on Figure 6. The structure consists of:

- addresses with sub-elements "Country", "Area", "Municipality", "City", "Post Code", "Quarter", "Street", "Floor" and "Apartment";
- phones with sub-elements "Area Code", "Phone Number" and "Phone Type";
- e-mails without sub-elements.

At the next step the data set was chosen. For the purposes of the experiments, the web document set was created using web pages for five categories organizations: companies, schools, museums, municipalities and libraries. The documents were picked out in html format using Google possibilities. For each category were selected first twenty web sites after searching for combination keywords "address" and one of keywords "company", "school", "museum", "municipality", "library" and with restriction "pages in Bulgarian".

Then, all documents from the data set was scanned and manually tagged in accordance with chosen hierarchical structure. Some of the documents are used later as instances in the learning set and other are used as instances in the testing set. At the first experiment we used ten-fold cross validation. At the second experiment the data set was split into learning set and testing set in random principle.

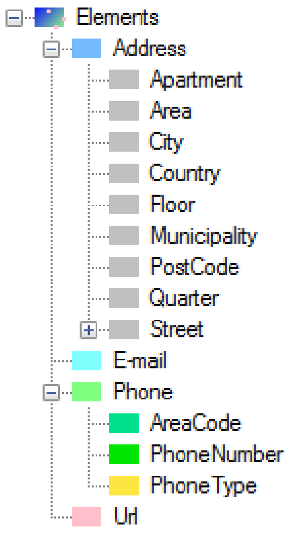


Figure 6. Sample element hierarchy for information extraction

Since the task is to find and extract all data that represents a given element we tested the system using the following criteria:

- Recall – the percentage of manually annotated elements for which an overlapping element is found in the results of the search;

- Precision – the percentage of found elements that overlap with a manually annotated element [Taylor, 1982];
- Accuracy – the average similarity between the original annotated elements and the correctly extracted elements.

For a similarity measure we propose to set the ratio of the length of the overlapping to the length of the union of the original and extracted texts:

$$similarity = \frac{A \cap B}{A \cup B}$$

where A and B are the original and extracted line segments respectively.

1. Ten-fold cross-validation test

Because of the wide diversity in which those elements can occur we split the data into 10 parts and performed a 90% learn – 10% test evaluation testing once each part [Kohavi, 1995]. Table 2 shows the results for each of the three element types.

Table 2: Results for extracting addresses, phones and e-mails without sub-elements

	Count	Recall	Precision	Accuracy
Address	134	51.54%	74.56%	57.25%
Phone	296	82.71%	87.45%	69.41%
E-mail	102	89.96%	97.29%	95.44%

During generalization the number of states in the automata has been reduced on average by 71%, 72% and 90% for addresses, phone numbers and e-mails respectively. The automata could be further compacted by merging common sub-trees.

This experiment shows the ability of the learning method to build generalized automata for parsing web documents. There appears to be a relation between the algorithm's performance and the structural variance of the information to be extracted.

2. Examination trend of reaching satisfactory results with increasing the cardinality of learning set

In other group of experiments the data set was split in two parts in a random principle – 40 instances were used as a learning set and the rest 60 documents were used as a testing set.

The system was learned using respectively 10, 20, 30 and 40 web documents from the learning set (each set contained the documents of the lower learning sub-set). Each time the testing was provided with all documents from the testing set. The test results were analyzed to obtain values for the numbers of fully extracted, partially extracted and elements that should have been but were not extracted. These experiments were provided in order to examine the trend of reaching satisfactory results. For each case multiple randomized runs have been performed to obtain more stable average values. We assume the address is recognized if its sub-elements, given in the text, are recognized. The telephone number is recognized if the system has recognized at least the phone code and the number. In several cases the system has recognized the string as a whole without recognizing its sub-elements. Partial recognizing means that some sub-elements are recognized (in particular one of them), but not the element as a whole. For instance only "town", "street", etc.

Table 3 shows the obtained results.

Table 3: Precision for extracting addresses, phones and e-mails when learning sets were 10, 20, 30 and 40 documents respectively

Number of Learning Documents	Address	Phone	E-mail
10	45.33%	85.54%	93.30%
20	50.43%	81.17%	86.72%
30	54.24%	84.23%	88.73%
40	66.19%	85.57%	93.35%

Figures 7, 8 and 9 shows the trend of increasing learning accuracy with increasing of the cardinality of learning set for elements and sub-elements of addresses and phones respectively.

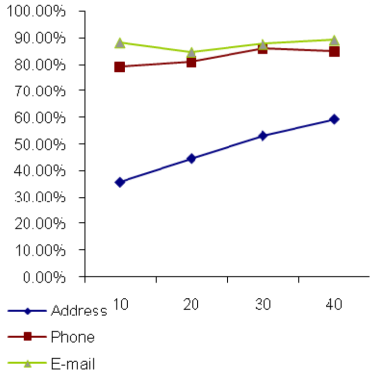


Figure 10. F-measure for addresses, phones and e-mails

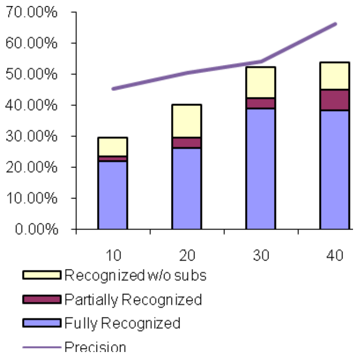


Figure 11. Recall and precision for the addresses

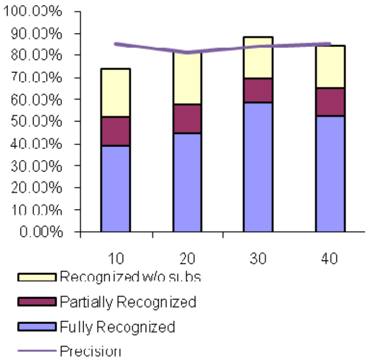


Figure 12. Recall and precision for the phones

In this experiment we found there is a trend for increasing the accuracy of the extractor by increasing the learning data set. As expected, e-mail addresses show the highest recall and precision and achieve high accuracy with a small cardinality of the learning set. The main reason for it is probably the existence of a strict and short structure for an e-mail address which leads to little variety in the different element instances. In that case learning over wider range of documents can actually sometimes prevent the optimal generalization resulting in worse results (Table 3). Bulgarian addresses show the worse results. Given the extremely wide variety of the indirect representation of Bulgarian addresses the results for this element are very promising. Furthermore, by increasing the learning and testing data sets the automata should begin to comprise of the most common cases which will lead to results comparable to the ones for the other two elements.

Our results are compatible with [Cohen, 2004]. The differences are coming from different languages and different grammars' structure in the languages.

Conclusion

The aim of the current work was to propose an approach for indirect spatial data extraction by learning restricted finite state automata from marked web documents. The learning method, program realization and experiments were presented. The proposed approach is suitable to cover practical needs for automatic extraction of indirect spatial data from web documents created using Bulgarian language.

The developed system InDES and provided experiments showed that such approach is acceptable and can be used in INSPIRE network.

The main idea of the approach is based on the understanding that the most of indirect spatial information objects are referenced to specific locations or geographic areas using the addresses. In near future, such kind of information will be given following the INSPIRE Data Specification of Addresses [INSPIRE-DSAD, 2008]. It is good standard which is accepted all over the European Community and need to be basis for the further investigation.

The future work involves research in the following directions:

- Adding external knowledge to the system (part-of-speech tagging, named entity lists, word ontology);
- Enhancing the generalization algorithm to identify common sub-trees and merge them if possible;
- more detailed comparison the performance of the realized system with other existing systems like MinorThird which implements various other extractor learning algorithms [Cohen, 2004].

Acknowledgements

This work is partially supported by Bulgarian National Science Fund under the project D 002-308 / 19.12.2008 "Automated Metadata Generating for e-Documents Specifications and Standards".

Bibliography

[Baltes, 1992] J. Baltes. Symmetric Version Space Algorithm for Learning Disjunctive String Concepts. Technical Report 92/469/06, University of Calgary, Calgary, Alta, March 1992

[Ciravegna, 2001] F. Ciravegna. Adaptive Information Extraction from Text by Rule Induction and Generalisation. IJCAI 2001, pp. 1251-1256.

[Cohen, 2004] W. Cohen. Minorthird: Methods for Identifying Names and Ontological Relations in Text Using Heuristics for Inducing Regularities from Data. 2004. http://minorthird.sourceforge.net

[Holzmann, 1991] J. Holzmann. Design and Validation of Computer Protocols. Prentice Hall, 1991, 512 p.

[IBM, 2009] Trainable Information Extraction Systems. http://researchweb.watson.ibm.com/IE/

[INSPIRE Directive, 2007] Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF

[INSPIRE-DSAD, 2008] INSPIRE Data Specification of Addresses http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/DataSpecifications/INSPIRE_DataSpecification_AD_v2.0.pdf

[INSPIRE-DSM, 2007] INSPIRE Data Specifications: Methodology for the Development of Data Specifications http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/inspireDataspecD2_6v2.0.pdf

[INSPIRE-TAO, 2007] INSPIRE Technical Architecture Overview. http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/INSPIRETechnicalArchitectureOverview_v1.2.pdf

[Kohavi, 1995] R. Kohavi. A Study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection. International Joint Conference on Artificial Intelligence IJCAI, 1995.

[Li et al, 2008] Y. Li, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, H. Jagadish. Regular Expression Learning for Information Extraction. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, October 2008. Association for Computational Linguistics, pp. 21–30.

[Taylor, 1982] J. Taylor. An Introduction to Error Analysis. University Science Books, Mill Valley, California, 1982.

[Rozenfield et al, 2008] B. Rozenfield, R. Feldman. Self-supervised relation extraction from the Web. Knowledge and Information Systems, Volume 17, Issue 1, (October 2008), Springer-Verlag New York, Inc. USA, pp. 17-33.

Authors' Information

Dimitar Blagoev – Plovdiv University "Paisii Hiledarsk"i, PhD Student in Computer Informatics Department;
24, Tsar Asen St., 4000 Plovdiv, Bulgaria; e-mail: gefrix@gmail.com

George Totkov – Plovdiv University "Paisii Hiledarski"; chair of Computer Informatics Department;
24, Tsar Asen St., 4000 Plovdiv, Bulgaria; e-mail: totkov@uni-plovdiv.bg

Milena Staneva – Institute of Mathematics and Informatics – BAS; Information Systems Department;
Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria; e-mail: mstaneva@math.bas.bg

Krassimira Ivanova – Institute of Mathematics and Informatics – BAS, Information Systems Department;
Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria; e-mail: kivanova@math.bas.bg

Krassimir Markov – Institute of Mathematics and Informatics – BAS, Information Systems Department;
Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria; e-mail: markov@foibg.com

Peter Stanchev – Kettering University, Flint, MI, 48504, USA / Institute of Mathematics and Informatics – BAS;
chair of Information Systems Department; Acad. G.Bontchev St., bl.8, Sofia-1113, Bulgaria;
e-mail: pstanche@kettering.edu