# Putting Encyclopaedia Knowledge into Structural Form: Finite State Transducers Approach

**Vesna Pajić**

Faculty of Agriculture, University of Belgrade, Nemanjina 6, 11080 Zemun, Belgrade, Republic of Serbia, svesna@agrif.bg.ac.rs

**Summary**

In biology and functional genomics in particular, understanding the dependence and interplay between different genome and ecological characteristics of organisms is a very challenging problem. There are some public databases which combine this kind of information, but there is still much more information about microbes and other organisms that reside in unstructured and semi-structured documents, such as encyclopaedias. In this paper we present a method for extracting information from semi-structured resources, such as encyclopaedias, based on finite state transducers, consisting of two clearly distinguished phases. The first phase strongly relies on the analysis of the document structure and it is used for locating records of data in the text. The second phase is based on the finite state transducers created for extracting the data, which can be modified so as to achieve the preferred efficiency and it is used for extracting the particular characteristic from the text. We show how the two phase method is applied to the text of the encyclopaedia "Systematic Bacteriology". A fully structured database with genotype and phenotype characteristics of organisms has been created from the encyclopaedia unstructured descriptions.

## 1 Introduction

Some of the main activities in different scientific researches are analyzing, processing, comparing or sorting scientific data. Mathematical methods, often used in research, can be applied only to strongly structured data which are usually stored in databases. Those data can be obtained from scientific experiments, but there is often a need to look for some information or scientific facts in literature, scientific articles or encyclopaedia. In these documents, human knowledge resides in an unstructured form, most of it as a free form text, and it is not always easy to find, access, analyze or use. Developing intelligent tools and methods, which give access to document content and extract relevant information, is a key issue for knowledge and information management.

Information Extraction (IE) is one of the main research fields that attempt to fulfil these tasks. It is a part of artificial intelligence which studies and develops techniques used to detect and extract relevant information from larger documents and present it in a structured form. The final output of the extraction process varies; in every case, however, it can be transformed so as to populate some type of database.

The IE field has been initiated by the DARPA's (Defense Advanced Research Projects Agency of USA) Message Understanding Conference in 1987 ([1]). Originally, IE was defined as the task of extracting specific, well-defined types of information from the text of homogeneous sets of documents in restricted domains and filling pre-defined form slots or templates with the extracted information. Since then, the most of IE research has been mainly dealing with extracting named entities (proper names, toponyms, etc.) ([2], [3] and [4]).



doi:10.2390/biecoll-jib-2011-164

Today, IE is not restricted to tasks defined by MUC conference only and may have application in any field of human knowledge and any type of data ([5], [6]).

In biology and functional genomics in particular, the problem of associating phenotypic characteristics of an organism to molecules encoded by its genome is a very challenging one. Anticipating an organism's phenotype based on its genotype (i.e. molecular composition) is important for biodefense. There are several studies that address this challenge (e.g., [7], [8], [9], [10] and [11]), including literature mining for such associations. There are some public databases which combine this kind of information. For example, a database of Complete Microbial Genome, created and maintained by National Center for Biotechnology Information [12] is a comprehensive one, with more than 1250 complete microbial genomes at present. It contains genome data for micro organisms, but also ecological characteristics like habitat, motility, shape and others. Nevertheless, there is still much more information about microbes and other organisms that reside in unstructured and semi-structured documents, such as encyclopaedias, so having methods which could extract information from this kind of text would be a great advantage.

Our contribution to solving this problem is to systematically enrich the database of genotype / phenotype characteristics, by mining semi-structured resources so as to extract information and to provide for as thorough as possible resource, formatted and easily accessible by any other method for revealing such associations.

In our research, we developed a two-phase method for collecting the data from the encyclopaedia text, which is based on the finite state transducers (FST). Finite state transducers are commonly used in Natural Language Processing for different tasks, and idea of using FST for information extraction is not new ([5] and [13]). It has been suppressed lately by methods that rely on probability theory and statistics, which are based on Hidden Markov Models ([14], [15], [16], and [17]) and Conditional Random Fields ([5], [6], [18] and [19]). Systems that rely on probability theory need large sets of annotated text used for training, i.e. for setting the parameters of the system. This kind of training text is not always available to the researcher, so these methods can not always be used. The other disadvantage of systems based on probability is their precision. Even the systems with high precision extract some amount of irrelevant data, which can be a big problem if the extracted data is to be used for further research. The method we present uses FST for pre-processing the text, and also for describing the context of information and extracting the information itself. The great advantage of the method is its reusability and precision.

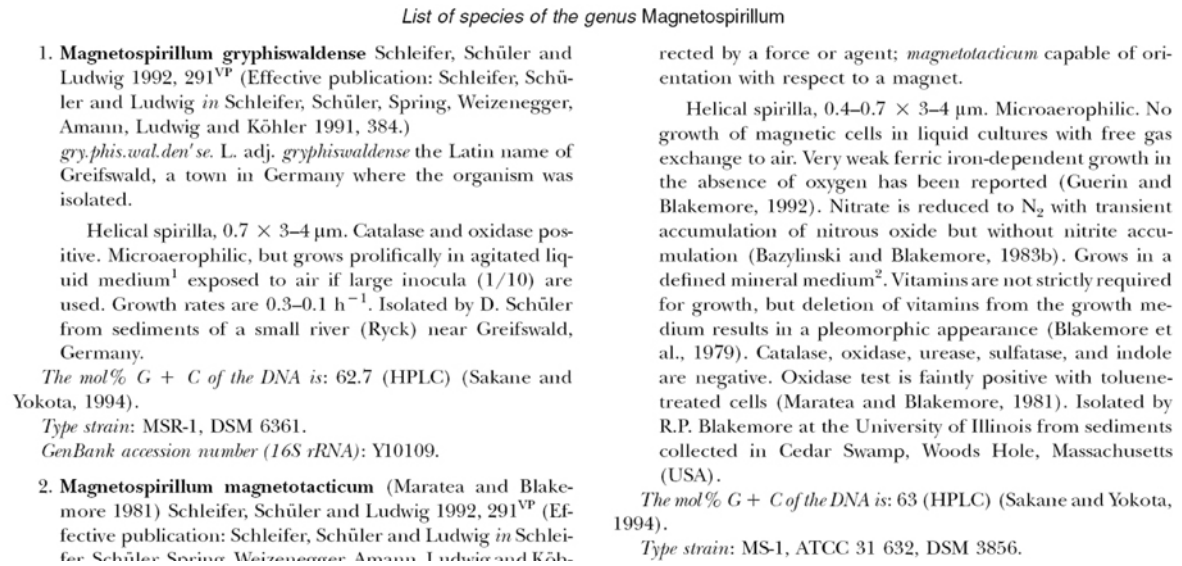
We applied two phase method on text of "Systematic Bacteriology" encyclopaedia [20], which is organized in such a way that can be treated as a semi-structured resource. As a result, we created a relational database containing records of data about microbes, obtained from the encyclopaedia text.

## 2 Horizontal and vertical problem of information extraction

The aim of our research was to extract ecological and genome characteristics about micro organisms from the encyclopaedia "Systematic Bacteriology" [20] and to create a relational database containing those data, which is to be used for future biological research. An example of a part of the encyclopaedia text containing organisms' descriptions with data we wanted to extract is given in Figure 1.



doi:10.2390/biecoll-jib-2011-164



*List of species of the genus* Magnetospirillum

1. **Magnetospirillum gryphiswaldense** Schleifer, Schüler and Ludwig 1992, 291[VP] (Effective publication: Schleifer, Schüler and Ludwig *in* Schleifer, Schüler, Spring, Weizenegger, Amann, Ludwig and Köhler 1991, 384.)
*gry.phis.wal.den'se.* L. adj. *gryphiswaldense* the Latin name of Greifswald, a town in Germany where the organism was isolated.

Helical spirilla, 0.7 × 3–4 μm. Catalase and oxidase positive. Microaerophilic, but grows prolifically in agitated liquid medium[1] exposed to air if large inocula (1/10) are used. Growth rates are 0.3–0.1 $h^{-1}$. Isolated by D. Schüler from sediments of a small river (Ryck) near Greifswald, Germany.

*The mol% G + C of the DNA is*: 62.7 (HPLC) (Sakane and Yokota, 1994).

*Type strain*: MSR-1, DSM 6361.

*GenBank accession number (16S rRNA)*: Y10109.

2. **Magnetospirillum magnetotacticum** (Maratea and Blakemore 1981) Schleifer, Schüler and Ludwig 1992, 291[VP] (Effective publication: Schleifer, Schüler and Ludwig *in* Schleifer, Schüler, Spring, Weizenegger, Amann, Ludwig and Köhrected by a force or agent; *magnetotacticum* capable of orientation with respect to a magnet.

Helical spirilla, 0.4–0.7 × 3–4 μm. Microaerophilic. No growth of magnetic cells in liquid cultures with free gas exchange to air. Very weak ferric iron-dependent growth in the absence of oxygen has been reported (Guerin and Blakemore, 1992). Nitrate is reduced to $N_2$ with transient accumulation of nitrous oxide but without nitrite accumulation (Bazylinski and Blakemore, 1983b). Grows in a defined mineral medium[2]. Vitamins are not strictly required for growth, but deletion of vitamins from the growth medium results in a pleomorphic appearance (Blakemore et al., 1979). Catalase, oxidase, urease, sulfatase, and indole are negative. Oxidase test is faintly positive with toluene-treated cells (Maratea and Blakemore, 1981). Isolated by R.P. Blakemore at the University of Illinois from sediments collected in Cedar Swamp, Woods Hole, Massachusetts (USA).

*The mol% G + C of the DNA is*: 63 (HPLC) (Sakane and Yokota, 1994).

*Type strain*: MS-1, ATCC 31 632, DSM 3856.

**Fig 1. Part of the encyclopaedia containing description of specific microbes**

During the extraction process, we had to resolve two problems. The first one was to determine the data records the information refers to, that is to locate the pieces of text containing the information about a specific microbe. This problem is usually called the Vertical Problem or VP ([6]).

The second problem was to extract particular information from the data record. For example, in encyclopaedia we used in our research, the individual attributes of records are located in the free form description. Those descriptions contain different kind of data about an organism, such as their shape, habitat, Gram stain, type strain, G + C content of the DNA, GenBank accession number etc. Those data (attributes) are the target for our research. Not every attribute were located in every description. For instance, only 76 out of 643 species had information about Gram stain property. The main task of our research was to extract as many of these information as possible and convert them into the fully structured form, i.e. put them into the relational database. We refer to the problem of extracting these attributes as Horizontal Problem or HP ([6]).

The structure of encyclopaedic text, i.e. how it is organized into chapters, is such that it allows drawing some conclusions about the data based on chapter titles, paragraphs and other parts of the text. Based on that fact, but also having in mind that we will process some other documents in the future looking for the same type of information, we decided to develop a method that solves the vertical and horizontal problems separately.

Resolving the vertical problem depends on the structure of the encyclopaedia. We therefore analyzed the way the text is organized and developed an algorithm that breaks the encyclopaedia text into smaller pieces, where each one of them contains information about only one organism. As an output, we would than have a database with names of organisms and their descriptions in a free form text. In that way we would solve the vertical problem.

On the other hand, as a start point for solving horizontal problem we would have a database with the organism descriptions. Bearing in mind that it is very important that all extracted data are relevant, so they can be used for future biological research, we decided to describe separately context of each attribute, i.e. each characteristic of an organism, we wanted to extract. For that purpose we used finite state transducers, which are going to be explained in Section 3.

The method we propose is intended for information extraction from text resources whose structure can be used for resolving the vertical problem. This approach, in which the problem

of information extraction is divided into two sub-problems being solved separately and independently of each other, is justified by the possibility of reusing the transducers.

# 3 Finite state transducers

Finite state transducers (FST) are finite state machines ([21], [22]) which define relations between two sets of strings by means of a transformation of one string into another. Finite state transducers are being used in many fields of computational linguistics. Their use is justified from the standpoint of linguistics as well as from the standpoint of computer science. From the linguistics point of view, FST are adequate for describing relevant local phenomena in language research and for modelling some part of a natural language, such as its phonology, morphology or syntax. Some examples of adequate representation of different linguistics phenomena by finite state machines are given in [23]. From computer science point of view, use of finite state transducers is motivated by time and space efficiency. Time efficiency is achieved by using deterministic finite state machines. The size of the output of deterministic machines depends mostly on the size of an input, so they can be considered to be optimal ([21] and [22]). Space efficiency is achieved by minimizing deterministic machines [24].

The basic property of FST is that they produce some output and this property determines the way transducers are being used in natural language processing (NLP). This property makes them very suitable for information extraction process, too. Also, they can be visually presented by graphs, which makes them convenient for human use. FST is being used in computational linguistics for morphological parsing, describing orthographic rules, describing inflectional rules etc. Detailed review of theoretical and practical use of finite state transducers in natural language processing is given in [2], [13], [22], [25], [26], [27], [28] and [29].

Finite state transducers can be very complex and difficult to maintain, which, in practice, leads to some problems. For example, if someone tries to describe the language syntax by a finite state machine, the corresponding graph would be very immense, and finding some particular information, such as noun phrases, would be time consuming and impractical. So, instead of one big graph, we use a collection of sub-graphs. This method has a strong theoretical background in the theory of Recursive Transition Networks (RTN). RTN are extension of context free grammars ([22], [30] and [31]). The arcs in RTN are labelled with corresponding grammars, while the states are labelled arbitrarily.

There are several computer tools for linguistic research based on FST and RTN ([32], [33] and [34]), which can be used for different tasks in text processing.

# 4 The structure of the resources used

## 4.1 Software system for linguistic tasks

In our research, we used Unitex [33] as a tool for creating and applying FST graphs, and also for pre-processing the text. Unitex is a collection of programs developed for analyzing natural language text using linguistic resources and tools. The reason we choose Unitex is because of its well designed and functional GUI for creating graphs, but one of the main reasons was possibility to use linguistic resources, such as electronic dictionaries and grammars.

Electronic dictionaries contain simple and compound words, together with their lemmas and a set of grammatical codes. They are constructed by teams of linguists for different languages as well as computer scientists (for English language [35], [36], [37] and [38], for French

language [39], [40] and [41], for Serbian language [42] and [43]). Unitex uses electronic dictionaries in DELA format, where each entry is a line of text terminated by a new line, which conforms to the following syntax:

```
apples,apple.N+conc:p
```

The first word (`apples`) is an inflected form of the entry and it is mandatory. In the former example it is followed by the canonical form (lemma) of the entry. This information may be left out if the canonical form is the same as the inflected form. The following sequence of codes (`N+conc`) gives the grammatical and semantic information about the entry. In the former example, code `N` stands for noun and conc indicates that this noun designates a concrete object. Code `p` stands for plural.

After applying these resources to a text, Unitex creates separate files with simple words, compound words and unrecognized words. Those files are than used in a search process, so one can refer to the dictionary entry from the Unitex by using lexical masks. For example, if some text is pre-processed by applying dictionaries, then the user can use a query `<be.V>` which matches all entries having `be` as the canonical form and the grammatical code *V*. Thus, all occurrences of the verb *to be* (*am*, *is*, *being* etc.) will be recognized by this query.

Using this kind of linguistic resources is the main advantage of the Unitex system, because the researcher can define classes of words and phrases with quite simple patterns, just by using information from the dictionary.

## 4.2 Examples of transducers

### 1.2.1. Example 1: The RTN for extracting information about genome size.

Figure 2 shows several graphs used for extracting information about genome size, created in Unitex software.

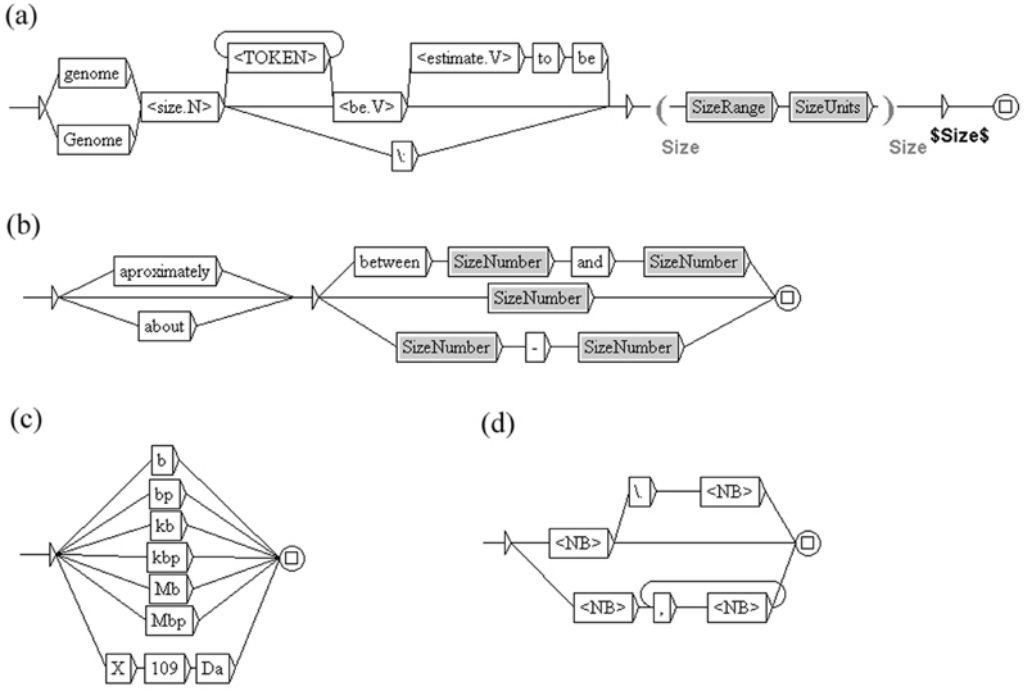



**Fig. 2. (a) Transducer for extracting information about the genome size; it contains calls to sub graphs SizeRange and SizeUnits; (b) SizeRange sub graph for describing possible ways of specifying size value; (c) SizeUnits sub graph which recognizes units for size of the genome; (d) SizeNumber sub graph for describing different formats of numbers**

The main transducer graph is shown in the Figure 2.(a). It describes the context in which the information is expected to be found in a text and defines which part of the text has to be extracted as data. The context of information has been established based on analysis of the encyclopaedia text.

The algorithm that uses this transducer reads word by word from the original description. Every read word is used for passing through the transducer. For example, when the algorithm reads the word "genome", it can start passing the transducer. The next word or phrase should match the noun "size" in any inflectional form in order to proceed with passing. The information about different forms of the word "size" is obtained from electronic dictionaries. The main transducer (Figure 2.(a)) also uses lexical masks such as `<be.V>` and `<estimate.V>`, which recognize any inflected form of the verbs *to be* or *to estimate*, in order to describe the context in which the information about genome size could occur. Only if the sequence of read words matches the path of the transducer, it is recognized by the transducer, and the output is produced. The output is defined by parentheses in the transducer.

The main transducer graph calls two sub-graphs, named *SizeRange* and *SizeUnit*, shown in Figure 2.(b) and Figure 2.(c). The graph *SizeRange* describes contexts corresponding to different ways of expressing the size of a genome sequence in encyclopaedia text, such as "*between 2240 and 3787*", "*1256–1276*" or "*approximately 4061*". The graph *SizeUnits* describes possible units of genome size. The graph *SizeNumber* describes different ways to refer to a number in a text. The special symbol <NB> matches any contiguous sequence of digits.

The following phrases are recognized by the transducer from the Figure 2. Data extracted and put into the database are in bold. So, the transducer recognizes the whole phrase, but it extracts only marked piece of information.

*"genome sizes of four G. oxydans strains were estimated to be **between 2240 and 3787 kb**"*
*"genome size of R. prowazekii is **1,111,523 bp**"*
*"genome size of R. africae is **1.248 kb**"*
*"genome size of R. australis is **1256–1276 kbp**"*
*"genome size is **2.62 X 109 Da**"*
*"Genome size: **2.73 X 109 Da**"*
*"genome size is **1.713 Mbp**"*
*"genome size was estimated to be **approximately 4061 kb**"*
*"genome size of all the classical strains examined was **about 3000 kb**"*

### 1.2.2. Example 2: The RTN for extracting information about Gram stain property

Figure 3 shows graphs used for extracting information about Gram stain property. They will recognize, among others, the following phrases as well:

"*Gram positive*", "*gram-positive*", "*Gram +*", "*Gram-pos.*", "*gram negative*", "*Gram -*"

These phrases will be recognized by one of several possible paths of the graph. Depending on the information stored in the text, these paths will produce output "*pos*" for the Gram positive bacteria, or "*neg*" for Gram negative bacteria. The output of the transducer represents the actual information we wanted to extract and it will be stored in the database.

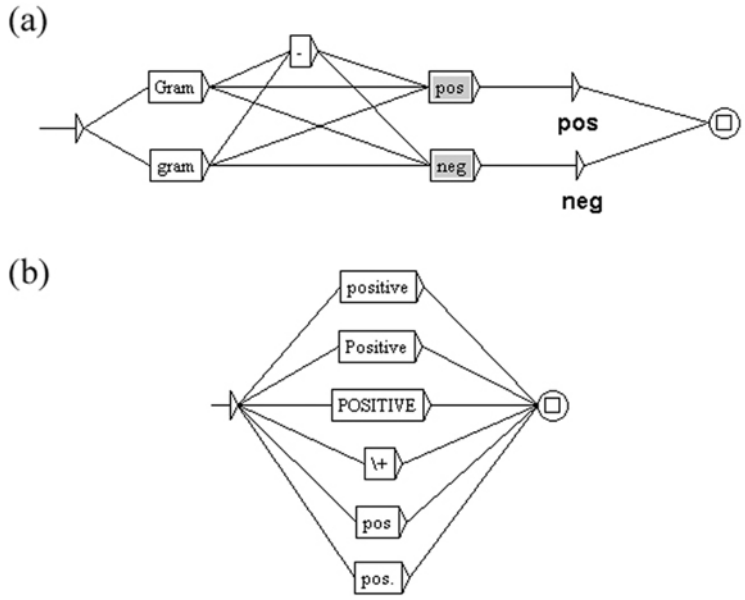



**Fig. 3. (a) A graph representing the transducer for extracting information about Gram stain property of an organism; (b) a sub graph called "pos" which describes possible tokens that refer to Gram positive property**

### 4.3 Semi-structured resource: Encyclopaedia

The aim of our research was to extract information about genome and ecological characteristics of microbes from a free form text and put them into relational database. As a resource, we used electronic format of encyclopaedia "Systematic Bacteriology" [20]. The structure of the encyclopaedia is such that it is possible to be used for information extraction process, so we treated it as a semi-structured document.

The text was given to us in the `.pdf` format. The first step was to convert it to `.txt` format. During this conversion some information about the structure of the document was misinterpreted. This applies to information about table data and some paragraphs. For example, header and footer of the pages were converted to paragraphs. Nevertheless, the main content of the document preserved its structure.

As already mentioned, we treated the given document as a semi-structured one. This means that the algorithm for locating pieces of text containing data strongly relies on the structure of the encyclopaedia. Therefore, the analysis of this structure was one of the key points of our research.

The content of the document is as follows. The chapters of the encyclopaedia correspond to systematic categories of the Bacteria super kingdom. For instance, the first chapter of the book is a description of the class *Alphaproteobacteria*, the next one is a description of the first order (*Rhodospiralles*) of the current class, followed by the chapters about families of that order. Each chapter with the family description is followed by the chapters of the genera in that family. The excerpt from the content is given in the Figure 4.

Descriptions of the species, containing information we want to extract, are given inside the chapters about genera, located at the end of the chapters. The part of the text containing species description is preceded by the line beginning with "*List of species of the genus ...*" There is a different number of species descriptions for different genera, but each one of them begins with the number, followed by the name of the species and description in a free form. The described structure of the document was used to discover data records, as will be explained in the Section 5, where each record corresponds to one systematic category.

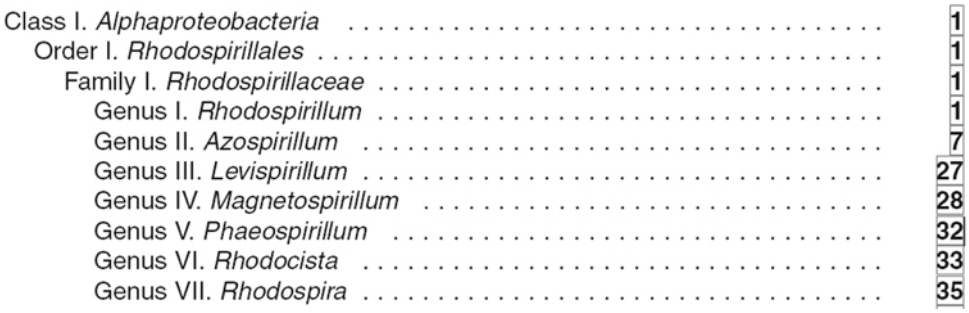



**Fig. 4. The excerpt from the content of the encyclopaedia**

## 5 The two phase FST method

The method we used for extracting the information from the free form encyclopaedia text distinguishes two phases of IE process. Both phases were implemented through a specially designed software system using programming language Java.

The first phase strongly relies on the structure of the document from which the extraction is to be done, i.e. on the structure of the encyclopaedia. During the first phase, the main goal is to locate pieces of the text in which the information about one record is situated. Those pieces of text are being put in a relational database, for further analysis. In this way the vertical problem of information extraction is resolved.

In our research, having the "*Systematic Bacteriology*" as a resource, we used the fact that each chapter of the text corresponds to one systematic category (*Class, Order, Family and Genus*). We developed an algorithm which reads content of the encyclopaedia line by line. Each line in the content has the same structure: the first word is the systematic category, followed by a number and a dot, and ending with the name of the category (e.g. "*Family I. Rhodospirillaceae*"). The algorithm reads the first word and uses this value as a name for the table where the record should be inserted e.g., "Family". The name of the category was used as a value for the field *FamilyName* of the record, e.g. " *Rhodospirillaceae"*. Order in which different categories appear in content was used to establish the connections between the different tables. At the end of this process, we had a database with data in several tables: *Class*, *Order*, *Family*, *Genus* and *GenusIncSed* (for "*Genus Incertae Sedis*", taxonomic group where its broader relationships are unknown or undefined). For example, the structure of the *Family* table is shown in the Table 1. The field *OrderID* is used to establish the relationship with the corresponding record in the *Order* table, i.e. with order where particular family belongs.

**Table 1. The structure of the table Family**

| Field Name | Data Type |
|---|---|
| ID | Integer |
| OrderID | Integer |
| FamilyName | Text |

A table with species descriptions was created in the next step. At first, we had to pull out the following information from the text and put it into the database:

- the name of the species
- does it belong to *Genus* or *Genus Incertae Sedis*
- the name of the genus it belongs to
- the description of the species

Having in mind that our final goal was to extract particular attributes about bacteria species, the table *Species* was created with fields as shown in the Table 2.

**Table 2. The structure of the Species table**

| Field Name | Data Type |
|---|---|
| ID | Integer |
| GenusID | Integer |
| SpeciesName | Text |
| SpeciesDesc | Text |
| Size | Text |
| GC | Text |
| GenBankNmbr | Text |
| TypeStrain | Text |
| Gram | Text |
| Habitat | Text |

In the first phase we were focused to fill only first four fields (i.e. *ID*, *GenusID*, *SpeciesName* and *SpeciesDesc*), while the remaining fields were filled in the second phase.

In order to populate the table *Species*, the algorithm tries to find the text "*List of species*". This text was followed by a list of species description, which was the target for our search. The main goal was to locate and extract these parts of the document. Each description starts with the number followed by the dot and the name of the species, e.g.

*1. Rhodovulum sulfidophilum (Hansen and Veldkamp 1973) .....*

This fact is used by the algorithm in order to determine records for the table Species, i.e. to identify beginning and ending of one species description. Unfortunately, during the conversion from `.pdf` to `.txt` format, some paragraphs was misinterpreted, so there were some lines which begin with a number and a dot, but which do not represent the beginning of the species description. In order to resolve this problem, the algorithm was modified in a way that it uses the fact that the first word in the name of the species is the name of the genus it belongs to. Therefore, the algorithm expects to find the name of the current genus after the dot. Only lines which fulfill this expectation are treated as the beginning of a new species description.

This kind of modification has led to excellent efficiency of the algorithm. Every species' description existing in the document was found and put into the database. This kind of fine improvements of the algorithm is possible by analyzing the structure of the resource. It is up to the researcher to modify the algorithm to reach the preferred level of the efficiency.

Here we present the first phase algorithm we used for resolving the vertical problem of information extraction from the encyclopedia "Systematic bacteriology", but we want to stress that this algorithm strongly depends on the structure of the text resource, and therefore it will be different for other documents.

```
set file to the encyclopedia content file
while not end of file
       read the line from the file
       if line starts with the name of some taxonomic category
              read the first word and set as category
              read the second word and set as name
              insert the record in the table corresponding the category
       end if
end while
set file to the encyclopedia text
while not end of file
       read the line from the file
       if line starts with "List of species"
```

```
                set desc to true
        end if
        if line starts with the name of some category
                if line starts with “Genus”
                        set the name of the Genus to be current
                end if
                if newSpecies
                        insert the speciesDesc and the speciesName
                                to the database
                        set speciesDesc to empty string
                        set speciesName to empty string
                end if
                set desc to false
                set newSpecies to false
        end if
        if desc
                if line starts with number followed by dot
                        if the first word after dot not equal current genus
                                break
                        end if
                        if newSpecies
                                insert the speciesDesc and the speciesName
                                        to the database
                                set speciesDesc to empty string
                                set speciesName to empty string
                        end if
                        set newSpecies to true
                        set speciesName to the first two words after the dot
                        set speciesDesc to the rest of the line
                else
                        appent speciesDesc with the line
                end if
end while
```

After the first phase had been finished, we had the database containing the data about species (their names – field *SpeciesName*, belonging genus – field *Genus* and free form descriptions – field *SpeciesDesc*). The part of the data in the table Species is shown in the Figure 5.

Species : Table

| | ID | Genus | GenusI | SpeciesName | SpeciesDesc | Size | GC | GenBankNmbr | TypeStrain | Gram | Habitat |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | Genus | 1 | Rhodospirillum rubrum | (Esmarch 1887) Molisch 1907, 25AL (Spirillum rubrum Esmarch 1887, 230.) rub'rum. M.L. neut. Adi. rubrum red. Cells are vibrioid | | | | | | |
| | 2 | Genus | 1 | Rhodospirillum photometricum | Molisch 1907, 24AL pho.to.me'tri.cum. Gr. n. phos light; Gr. adj. metricus measuring; M.L. neut. adi. photometricum light measuring. | | | | | | |
| | 3 | Genus | 2 | Azospirillum lipoferum | (Beijerinck 1925) Tarrand, Krieg and Do¨bereiner 1979, 79AL (Effective publication:Tarrand, Krieg and Do¨bereiner | | | | | | |
| | 4 | Genus | 2 | Azospirillum amazonense | Magalha~es, Baldani, Souto, Kuykendall and Do¨bereiner 1984, 355VP (Effective publication: Magalha~es. Baldani. Souto. | | | | | | |
| | 5 | Genus | 2 | Azospirillum brasilense | Tarrand, Krieg and Do¨bereiner 1979, 79AL (Effective publication: Tarrand, Krieg and Do¨bereiner 1978, 979.) bra.si.len'.se. M.L. | | | | | | |

**Fig. 5. The look of the table “*Species*” after the first phase is finished**

In the second phase, the system takes an unstructured text with information about a record from the database and analyzes it with FST graphs. For every attribute (kind of information) we wanted to extract, we created separate transducer using the Unitex. Each transducer recognizes the context of some information and produces the output which represents the information itself. This output is inserted into the database. The approach of fulfilling the second phase is represented in the Figure 6.

The use of transducers for the tasks of describing context and extracting information was motivated by the fact that in biological text there is a limited number of possible phrases for describing some properties of an organism. For example, there are no many different ways to tell that some bacteria are a Gram negative one. Therefore, by designing a transducer which recognizes a part of the text about Gram stain and produces the output “*positive*” or “*negative*”, depending on the information in the text, we can process not only descriptions from the encyclopaedia, but also we can process any other textual resource about bacteria.

As mentioned in Section 3.1., we used the Unitex software system for creating this kind of graphs, and also for pre-processing the text. Beside the pre-processing tasks which are required by the Unitex's programs for locating patterns in the text, such as normalization and tokenization of the text, we used the possibility of applying linguistic resources to the text and applied English electronic dictionary to the descriptions. This way we could use lexical masks in transducers.

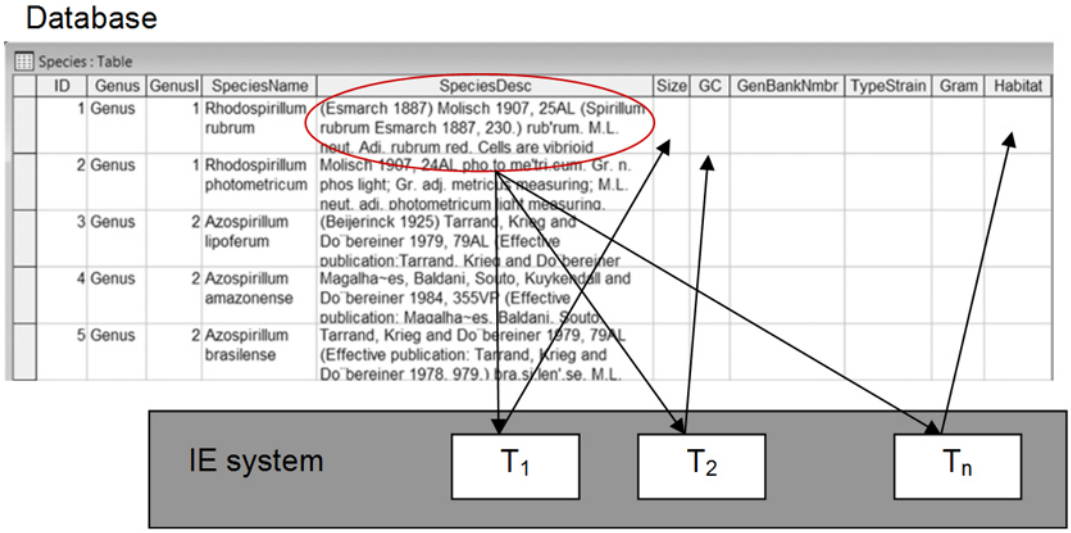



**Fig. 6. Using transducers $T_1$, $T_2$, ..$T_n$ for information extraction – illustration of approach**

The algorithm of the second phase was as follows:

```
for each record in the table "Species"
        read the description from the record
        for each transducer ti
                apply transducer ti on description
                insert the output of the transducer ti into attribut ai column
        end for
end for
```

## 6 Results and evaluation of the method

The encyclopaedia "Systematic Bacteriology" contains descriptions of 643 species of bacteria, grouped by the genus they belong to. As mentioned in the Section 5, the algorithm we used for the first phase of the proposed method was very efficient and it extracted all of the 643 descriptions. The reason for achieving such a good efficiency is thorough analysis of the document structure. The initial algorithm was tuned and modified by the researchers until it has reached such an excellent level of efficiency.

Species : Table

| ID | Genus | GenusI | SpeciesName | SpeciesDesc | Size | GC | GenBankNmbr | TypeStrain | Gram | Habitat |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Genus | 1 | Rhodospirillum rubrum | (Esmarch 1887) Molisch 1907, 25AL (Spirillum rubrum Esmarch 1887, 230.) rub'rum. M.L. neut. Adj. rubrum red. Cells are vibrioid shaped to spiral. 0.8-1.0 lm wide: one complete turn of | | 63.8-65.8 | D30778, M32020 | ATCC 11170, DSM 467, NCIB 8255 | | stagnant and anoxic freshwater habitats that |
| 2 | Genus | 1 | Rhodospirillum photometricum | Molisch 1907, 24AL pho.to.me'tri.cum. Gr. n. phos light; Gr. adj. metricus measuring; M.L. neut. adj. photometricum light measuring. TABLE BXII. .2. Carbon sources and electron | | 64.8-65.8 | AJ222662 | ATCC 49918, DSM 122, NTHC 132 | | stagnant and anoxic freshwater habitats that |
| 3 | Genus | 2 | Azospirillum lipoferum | (Beijerinck 1925) Tarrand, Krieg and Do bereiner 1979, 79AL (Effective publication:Tarrand, Krieg and Do bereiner 1978. 978 (Spirillum lipoferum Beiierinck 1925. | | 69-70 | M59061 | BR11080, Sp 59b, ATCC 29707, DSM 1691 | neg | |
| 4 | Genus | 2 | Azospirillum amazonense | Magalha~es, Baldani, Souto, Kuykendall and Do bereiner 1984, 355VP (Effective publication: Magalha~es, Baldani, Souto, Kuykendall and Do bereiner 1983. 417.) | | 67-68 | Z29616, X79735 | BR 11142, Am14, Y1, ATCC 35 119, DSM GenBank | | Soil and tissues + + + mainly of nonlegumes |

**Fig. 7 Table Species with data after the second phase of IE process**

After the second phase is finished, we had data about microbes extracted and inserted into the database. The table *Species*, previously shown in the Figure 5 in Section 5, at the end of the second phase looked like shown in the Figure 7. The extracted information was inserted in the corresponding fields of the database.

In order to evaluate efficiency of transducers, we manually analyzed species description in order to calculate precision and recall of the method. Precision and recall are common and frequently used measures of efficiency of an information extraction system. In information extraction context, precision and recall are defined in terms of a set of extracted information ($S_{ext}$) and a set of relevant information ($S_{rel}$) that is located in the text. Precision and recall are then calculated based on the following formulas:

$$precision = |S_{ext} \cap S_{rel}|/|S_{ext}|$$

$$recall = |S_{ext} \cap S_{rel}|/|S_{rel}|$$

Precision is, therefore, measured with the amount of relevant data. That means that we don't want to have GenBank ID number read, but incorrectly identified as, for example, genome size. Precision, in our case, was the highest possible, i.e. all of the extracted information was relevant. This is a consequence of the fact that transducers were designed by human experts to extract particular attributes, and therefore they recognize only sequences of text in which specific information is stored.

Recall differs for different transducers, depending on the complexity of the information context. For example, the transducer for Gram stain property was very efficient; it properly extracted attributes from all descriptions which had that kind of information. Some other transducers, especially those for extracting information that occur in a complex context, weren't that efficient. For example, the initial transducer for genome size extracted 14 out of 18 information about genome size. Some expressions weren't recognized by this transducer, such as:

*"genome has a size of 1,231,204 bp"*

*"genome size is distinctly larger (1.49 X 109 Da)"*

*"genome is 1,257,710 bp in size"*

Nevertheless, with slight modification of the transducer and by extending it in order to recognize the former expressions as well, the recall can be increased. This is a key point and a major advantage of methods based on FST over methods based on probability. Efficiency of methods based on FST can be increased to the preferred level by modifying transducers. This fact, together with the fact that transducers can be reused for other resources in the same domain makes this method justified and suitable to use for IE tasks.

## 7 Conclusion

In this paper we successfully applied the proposed two phase method for information extraction on encyclopaedia containing different kind of biological data. We treated the encyclopaedia as a semi structured resource and used the structure of the document to conclude the facts about relationships between the data. The second phase of the method involves creating and applying transducers to the text from which the information is to be extracted. We showed that using this method is very efficient, especially when applied to a text from some specific science or domain, in which case the transducer has to describe relatively simple context of information. The use of transducers is also justified by their reusability in some other text of the same domain.



doi:10.2390/biecoll-jib-2011-164

The advantages of the proposed two-phase FST-based method is its conceptual simplicity, efficiency, possibility to adjust precision of the transducers, reusability of the transducers and no need for large sets of training data.

We hope that this method will attract more attention from the research community in the future, and that spreading its use will lead to creation of transducers collections, which can be reused by other researchers for collecting data from different documents containing biological data. We plan to make our collection of transducers, as well as databases with extracted information, available to others.

## References

[1] R. Grishman, B. Sundheim. Message Understanding Conference –6: A Brief History. Proceedings of COLING'96, Copenhagen, Denmark, pp. 466-471, 1996.

[2] N. Friburger, D. Maurel. Finite-state transducer cascades to extract named entities in texts. Theoretical Computer Science, 313:93 – 104, 2004.

[3] D. Maynard, K. Bontcheva, H. Cunningham. Towards a semantic extraction of Named Entities. In Recent Advances in Natural language Processing, Bulgaria, 2003.

[4] S. Sekine, E. Ranchhod. Named entities: Recognition, classification and use. John Benjamins Publishing Company, Amsterdam, Netherlands, 2009.

[5] G. Burns, D. Feng, E. Hovy. Intelligent Approaches to Mining the Primary Research Literature: Techniques, Systems, and Examples. Computational Intelligence in Medical Informatics, Studies in Computational Intelligence, Springer Berlin / Heidelberg, p. 17-50, 2008.

[6] D. Feng, G. Burns, E. Hovy. Extracting Data Records from Unstructured Biomedical Full Text. In Proceedings of the EMNLP conference, Prague, Czech Republic, 2007.

[7] C. S. Goh, T. A. Gianoulis, Y. Liu, J. Li, A. Paccanaro, Y. A. Lussier, M. Gerstein. Integration of curated databases to identify genotype-phenotype associations. BMC Genomics, 7:257, 2006.

[8] K. Jim, K. Parmar, M. Singh, S.Tavazoie. A cross-genomic approach for systematic mapping of phenotypic traits to genes. Genome Res, 14(1):109-115, 2004.

[9] J. Korbel, T. Doerks, L. J. Jensen, C.Perez-Iratxeta, S. Kaczanowski, S. D. Hooper, M. A. Andrade, P. Bork. Systematic association of genes to phenotypes by genome and literature mining. PLoS Biol, 3:134-134, 2005.

[10] N. J. MacDonald, R. G. Beiko. Efficient learning of microbial genotype-phenotype association rules. Bioinformatics, 26:1834-1840, 2010.

[11] M. Tamura, P. D'haeseleer. Microbial genotype-phenotype mapping by class association rule mining. Bioinformatics, 24:1523-1529, 2008.

[12] National Center for Biotechnology Information, U.S. National Library of Medicine 8600 Rockville Pike, Bethesda MD, 20894 USA

[13] F. Casacuberta, E. Vidal, D. Picó. Inference of finite-state transducers from regular languages. Pattern Recognition, 38(9):1431-1443, 2005.

[14] D. Freitag, A. Mccallum. Information Extraction with HMM Structures Learned by Stochastic Optimization. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, Austin, Texas, USA, AAAI Press, pp. 584—589, 2000.



doi:10.2390/biecoll-jib-2011-164

[15] A. Mccallum, D. Freitag. Maximum entropy markov models for information extraction and segmentation. Morgan Kaufmann, pp. 591—598, 2000.

[16] S. Ray. Representing Sentence Structure in Hidden Markov Models for Information Extraction. In Proceedings of the 17th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, pp. 1273—1279, 2001.

[17] P. Zhong, J. Chen, T. Cook. Web Information Extraction Using Generalized Hidden Markov Model. Hot Topics in Web Systems and Technologies, 2006. HOTWEB '06. 1st IEEE Workshop, p. 1-8, 2007.

[18] F. Peng, A. McCallum, Information extraction from research papers using conditional random fields. Information Processing & Management, 42(4):963-979, 2006.

[19] J. Zhu, Z. Nie, J.R. Wen, B. Zhang, W.Y. Ma. 2D Conditional Random Fields for Web information extraction. ACM International Conference Proceeding Series; Vol. 119 , Proceedings of the 22nd international conference on Machine learning, Bonn, Germany, pp. 1044 – 1051, 2005.

[20] G. M. Garrity. Systematic Bacteriology, Second Edition, Volume Two: The Proteobacteria, Part C: The Alpha-, Beta-, Delta-, and Epsilonproteobacteria. Bergey's Manual Trust, Department of Microbiology and Molecular Genetics, Michigan State University, USA, 2005.

[21] D. Vitas. Prevodioci i interpretatori: Uvod u teoriju i metode kompilacije programskih jezika. Matematički fakultet, Belgrade, Republic of Serbia, 2006.

[22] D. Jurafsky, J. H. Martin. Speech and language processing. Prentice-Hall Inc., 2000.

[23] M. Gross, D. Perrin. Electronic Dictionaries and Automata in Computational Linguistics. In Proceedings of LITP Spring School on Theoretical Computer Science Saint-Pierre d'Oleron, France, May 25.-29., 1987

[24] A. V. Aho, J. E. Hopcroft, J. D. Ullman. The Design and Analysis of Computer Algorithms. Addison Wesley, Reading, MA, 1974.

[25] J. R. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, M. Tyson. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Roche E. and Y. Schabes, eds., Finite-State Language Processing, The MIT Press, Cambridge, MA, pages 383-406, 1997.

[26] A. Kornai. Extended finite state models of language, Cambridge University Press, 1999.

[27] V. Pajic. Finite State Transducers in Web Monitoring. Master Thesis, Faculty of Mathematics, University of Belgrade, Republic of Serbia, 2010.

[28] E. Roche. Finite state transducers: parsing free and frozen sentences, Extended finite state models of language. Cambridge University Press, 108.-120, 1999.

[29] E. Roche, Y. Schabes. Finite-state language Processing. The MIT Press, 1997.

[30] J. M. Sastre, M. Forcada. Efficient parsing using recursive transition networks with output. In Zygmunt Vetulani, editors, 3rd Language & Technology Conference (LTC'07). 5-7 October 2007. pp. 280–284, 2007.

[31] J. M. Sastre. Efficient Parsing Using Filtered-Popping Recursive Transition Networks. Lecture Notes in Computer Science, vol. 5642, pp. 241–244, 2009.

[32] B. Olivier, M. Constant, E. Laporte. Outilex, plate-forme logicielle de traitement de textes ecrits. In Proceedings of TALN'06. Leuven, Belgium, UCL Presses universitaires de Louvain, 2006.

[33] S. Paumier. Unitex 1.2 User Manual, Université de Marne-la-Vallée, 2006.

[34] M. D. Silberztein. Dictionnaires électroniques et analyse automatique de textes. Le système INTEX. Paris: Masson. 1993.

[35] A. Chrobot, B. Courtois, M. Hammani-Mc Carthy, M. Gross, K. Zellagui. Dictionnaire electronique DELAC anglais : noms composés. Technical Report 59, LADL, Université Paris 7, 1999.

[36] G. Klarsfled, M. Hammani-Mc Carthy. Dictionnaire électronique du ladl pour les mots simples de l'anglais (DELASa). Technical report, LADL, Université Paris 7, 1991.

[37] A. Monceaux. Le dictionnaire des mots simples anglais : mots nouveaux et variantes orthographiques. Technical Report 15, IGM, Université de Marne-la-Vallée, 1995.

[38] A. Savary. Recensement et description des mots composés - méthodes et applications. Thèse de doctorat. Université de Marne-la-Vallée, 2000.

[39] B. Courtois. Formes ambiguës de la langue française, Lingvisticæ Investigationes, 20(1) Amsterdam-Philadelphia: John Benjamins Publishing Company, pp. 167 -202, 1996.

[40] B. Courtois and Max Silberztein, editors. Les dictionnaires électroniques du français. Larousse, Langue française, vol. 87, 1990.

[41] J. Labelle. Le traitement automatique des variantes linguistiques en français: l'exemple des concrets, Lingvisticæ Investigationes, 19(1), Amsterdam - Philadelphia: John Benjamins Publishing Company, pp.137–152, 1995.

[42] C. Krstev, D. Vitas. Corpus and Lexicon - Mutual Incompletness. In Proceedings of the Corpus Linguistics Conference, Birmingham, 14-17 July 2005.

[43] D. Vitas, C. Krstev, I. Obradović, Lj. Popović, G. Pavlović-Lažetić. Processing Serbian Written Texts: An Overview of Resources and Basic Tools. In Workshop on Balkan Language Resources and Tools, 21 Novembar 2003, Thessaloniki, Greece, pp. 97-104, 2003.



doi:10.2390/biecoll-jib-2011-164