Automated Extraction of Concept Matcher Thesaurus from Semi-Structured Catalogue-Like Sources of Data on the Web

Maxim Lapaev
ITMO University
St.Petersburg, Russia
m.lapaev@corp.ifmo.ru

***Abstract*—Ontology design and the process of populating a data-set with knowledge following the chosen or developed ontology to fit the principles of Semantic Web and Linked Open Data is a time-consuming and iterative process, requiring either expert knowledge or a set of tools for data scraping from web. A valid and consistent ontology and knowledge withing the data-set require unification of concepts which means overcoming ambiguity and synonymy of terms which become individuals of ontology. In this paper we spot on techniques used for organising a Russian food product data-set under a light-weight FOOD Ontology and concept matching in particular. Main approaches to data-set concept unification, synonymic term matching and ways to collect dictionaries for matcher are mentioned. The tool for catalogue-like semi-structured resources parsing and thesaurus extraction is developed and introduced for the task of on-the-fly concept matching.**

I. Introduction

Emerged in 1998 Semantic Web conception has developed at a fast pace till nowadays, enriching web with machine-understandable data-sets in all kinds of domain areas and enabling an entity search, content-based navigation and many other useful means of search and exploration which could be impossible without machine logic and reasoning. Semantic Web paradigm is an attempt to overcome limits of Web 2.0 such as ambiguity of concepts and implicit knowledge, its technology stack simplifies data integration, partly resolves the information retrieval dilemma and potentially converts global network into a huge distributed knowledge base, which is currently represented by a fastly growing Linked Open Data Cloud[1].

Whereas the web today is represented by data locked in small data islands which other applications usually cannot access (or data access is accompanied with restrictions up to a sudden change of data access API), Semantic Web technologies are to get rid of closed data, to publish structured data on the web and to draw connections from one data source to data from other data sources. The driven force of Semantic Web technology is a structured and meaningful data corresponding the restrictions and rules of ontology of a domain area which makes the data to become knowledge, i.e. machine-readable, -comprehensible and -processable building block of global knowledge graph suitable for reasoning and new knowledge inference.

A notable part of domain areas including geography, social networking, government, linguistics, media, publications etc. are already covered by Linked Open Data principles within the cloud, however there are still uncovered or not fully covered areas which include food products, in particular, which is the main focus of our work. Food product as an entity has a lot of attributes significant for the domain area some of which are hard to unify for the reason of synonymy and many ways to address to the very same entity (eg. food additives have both E-code and a list of alternative names, ingredients are represented by a free-form text without a real unification of names). Thus, a name unification is to be fulfilled for resolving data-set inconsistency and avoiding multiple individuals pointing to the same concept.

The issue of term synonymy in ontology design and data-set organisation is not a new problem and had been discussed and studied giving a birth to ontology-based knowledge extraction methodologies, however mentioned approaches are based either on already organized thesauri [2], [3] or assume relation of terms instead of complete synonymy [4] and more applicable for issues of ontology merging [5] or cross-ontology linking [6] rather then the process of ontology learning and populating and enriching a data-set.

II. Problem statement

Taking into considerations the purpose of mechanism of matching synonymic concepts and all prerequisites it may be concluded that a kind of thesaurus should be the central element and data source of the matcher. However we meet a lack of already organized thesauri and references in chosen domain area and in addition to that a lack of standards and codices devoted to ingredient naming is a significant problem as well, which assumes the only way to build a core data-source of matcher is to find a number of distributed all over the web sensible and more or less reliable sources of reference data and collect it into a single thesaurus.

For the sake of automation a parser is needed and a DOM-parsing approach is more applicable for the described problem. However, as was concluded, the data is distributed within a number of resources, but not a single resource, a different parser and thesaurus data extractor is required for every single web-catalogue which is quite a challenge, because coding a parser is a time-consuming process including programming and resource structure investigation prior to it. All in all, the time required does not worth the volume of data processed within a single data-source. A unified tool either able to parse every catalogue or to decrease expenses for parsing each catalogue is needed to solve the problem we faced with minimum loss of time which is the goal to be achieved in this study. Thus, a thorough investigation of achievements in automated web-data extraction and an overview of already existing tools has to be done.

The rest of the paper is organised as follows: Section 3 briefly overviews widely-used approaches to extraction of knowledge from unstructured and semi-structured web resources as well as data mining and wide-spread web-scraping tools. Section 4 is focused on techniques and means used for ingredient concept matcher, its algorithms and structure, shortly describes the tool designed and includes experimental verification process on some data-sources chosen as a test input semi-structured data. Section 5 gives an overview of the tool and data processed by the tool within overall process of data-set population.

III. Related works and tools

The topic of data extraction from semi-structured sources of information is not new to knowledge engineering. The ways to mine knowledge have been an object of attention and is still within a single data-source. A unified tool either able to well as extraction tools exist.

A. Common tools for semi-structured data extraction

Semi-structured data is referred to as an intersection area of the Web and database community as it is usually one of the ways to represent item data from database into a human-readable refined form using tables, styles and other common web representation techniques [7]. However it is still a complicated task to extract this data into a well-defined structural storage. Indeed, a semi-structured data extraction is a way to copy a database or some fields of database table from hosting server into another destination without a direct access to source database which is the main challenge as data was originally not intended for such use-case. Often even the most efficient web-crawling technology cannot replace an investigation and copy-and-paste including a lot of human manual work, and often it may be the only possible solution when the websites for extraction explicitly set up barriers to prevent automation. Thus, indirect and oblique methods emerged having both advantages and disadvantages and not suitable for every source. A number of tools for web data mining already exists and may be divided into few main categories based on approach: text grepping and regular expression matching, HTML parsing, DOM parsing, semantic annotation recognizing in the narrow sense as well as agent-based approach and database approach [8] in the broadest sense. As far as we are concerned with populating a thesaurus with terms and the domain is not covered by Semantic Web approaches yet, a database approach which assumes methods for transforming a semi-structured semantically unannotated web-page data into a rather structured form is of great interest for us. Investigation and comparison of widely-used extraction tools was carried out to check whether we may use one of the tools or a combination of tools to populate thesaurus. Most of existing web crawling tools are based either on stream parsing or DOM-parsing of web pages. Below is a brief overview of widely-used tools:

1) **Automation anywhere**
   - a Web data extraction tool used for retrieving web data effortlessly, screen scraping from web pages or using it for web mining;
   - records data;
   - extracts structured data;
   - extracts semi-structured;
   - has an easy-to-learn user interface.
2) **Web Info Extractor**
   - retrieving unstructured or structured data from web page, reorganizing into local file or saving to the database, placing into the web server;
   - does not support storing data;
   - extracts structured data;
   - extracts semi-structured;
   - user-friendly interface.
3) **Web Content Extractor**
   - the most powerful and easy tools for web scraping, data mining or data retrieval from; the Internet.
   - does not store the data retrieved;
   - extracts structured data;
   - does not support extraction of semi-structured data;
   - complicated user interface.
4) **Screen-scraper**
   - extracting information from web sites, allows a user to scrape structured and unstructured data from websites and format it;
   - lack of data storing mechanism;
   - extracts structured data;
   - extracts semi-structured data;
   - complicated and difficult-to-understand-and-learn user interface.
5) **Mozenda**
   - extracting web data easily and managing it affordably, the users can set up agents that regularly extract, store and circulate data to several destinations;
   - has a data storage mechanism, ability to store in a number of destinations;
   - extracts structured data;
   - extracts semi-structured data;
   - user-friendly interface.

All of the mentioned web data extractors are pretty suitable and convenient to use for the task of data extraction, however some of them lack in storing extracted data which is crucial for us. The other restriction is that all of the tools require a complicated tuning process prior to extraction and still extract data only from the URL list provided (Fig. 1), i.e. unable to pass through catalogues' pages containing terms. Analyzed web extractors are more suitable for the task of tracing a particular list of pages for changes and refreshing information in local storage (eg. price monitoring) rather then automated extraction of concepts from heterogeneous data sources and need an interaction with user. In addition, most of the tools shown to be efficients are enterprise and proprietary and not distributed for free. Unfortunately the tool suitable for automated thesaurus population is still not available and further researches in data mining and a presentation of a web extracting environment or parser generator that allows non-expert users to gain a web mining tool for a given domain and resource simply by providing a set of specifications are to be done.

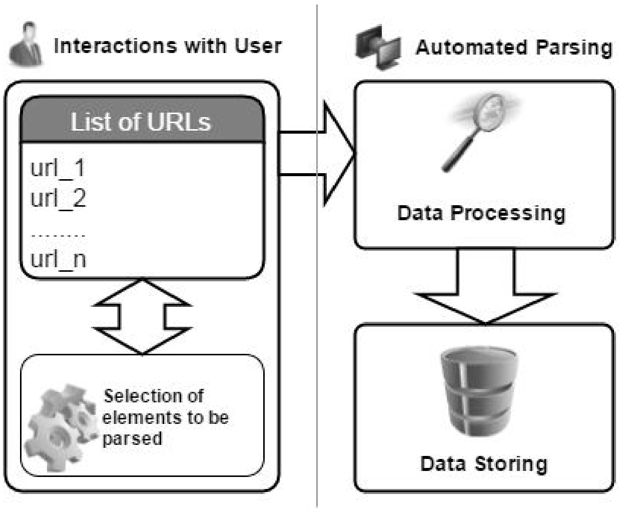


Fig. 1. General semi-automated approach to web-crawling: manual configuration and automated processing

B. Brief overview of researches on data extraction methods

Nowadays, when the amount of data on the web has increased dramatically, but the quality of data is constantly decreasing causing lots of irrelevant or false knowledge, the extraction of data and ontology learning are being roughly discussed and researched. The state-of-the-art researches and overviews deal with subjects of automated data processing and enrichment, entity ranking, content-based search and entity search, entity classification, thesauri and taxonomy organisation and population, data disambiguation and other approaches aimed at increasing data relevance and structuring.

One of the approaches presented by Kejriwal et al [9] is a semi-supervised instance matching using boosted classifiers. The research assumes instance matching by identifying pairs of individuals referring to the same underlying entity using machine learning methods. Authors explain that the system achieves quite significant performance after being trained on a significant amount of samples processed manually. The authors state an approach enabling to decrease the amount of manual labeling up to 2% of overall data available within a resource which is a significant breakthrough in machine learning. Using the tools shown to be efficients are enterprise and proprietary of entity recognition in other domain areas as well [10]. However, as it was mentioned by the authors, the system is semi-supervised, i.e. requires manual processing of entities chosen to be samples for machine learning. Although amount of manually labeled items is relatively small, total efficiency of the method applied to our thesaurus population having small portions of source data distributed over a great number of heterogeneous task is dubious as it leads to a completely manual extraction that is not a wise approach. Furthermore, the boosted classifier approach is criticised by some researchers for the reason that convex potential boosters cannot withstand random classification noise [11].

Another approach to web crawling and matching entities is ranking entities by a query-bases algorithm LDRANK for matching web of data resources with associated textual data [12]. Proposed algorithm is a combination of link analysis and dimensionality reduction. However the algorithm reliability is unknown and cannot be evaluated. The authors use a crowd-sourcing platform to verify and refine data processed by algorithm which is not far from a completely manual data matching. The similar issue as we have was encountered by other research group [13]. They introduce VocBench - an open source web application for populating and editing thesauri. However, the platform proposed has a strong focus on collaboration which has been mentioned above: it is still a great portion of manual work instead of automation. Furthermore developing a crowd-sourcing platform does not worth the task we deal with.

Other works devoted to entity linking and matching and word sense disambiguation [14], [15], [16] concentrate on relational mapping, i.e. propose techniques and solutions to disambiguate and match concepts with an entity from the list of entities that may somehow be related with the considered concept. Proposed methodologies are shown to be efficient in mapping relations, especially within a knowledge base, however one may notice that the techniques proposed still lack in accuracy of linking in case of exact match among semantically unannotated terms.

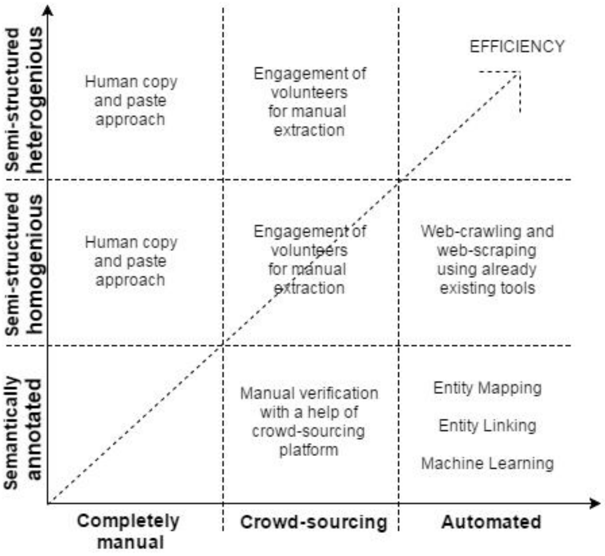

Semi-structured heterogenious
Semi-structured homogenious
Semantically annotated
Human copy and paste approach
Engagement of volunteers for manual extraction
EFFICIENCY
Human copy and paste approach
Engagement of volunteers for manual extraction
Web-crawling and web-scraping using already existing tools
Manual verification with a help of crowd-sourcing platform
Entity Mapping
Entity Linking
Machine Learning
Completely manual
Crowd-sourcing
Automated

Fig. 2. A diagram of existing extraction techniques: X-axis - techniques, Y-axis - type of data source

Along with analysis of data semi-structured extraction techniques and researches in this area an investigation on general efficiency of the use of heterogeneous resources was made. Researches provide an evidence that a moderate number of sources is sufficient to exploit for heterogeneous source data integration [17]. Taking into account all methods analyzed and processed a diagram of extraction techniques was proposed (Fig. 2). It demonstrates the general trend of efficiency as well as already existing method pairs ”resource type - extraction method”. It is shown that two cells of the diagram, one of which is intended for automated heterogeneous semi-structured source processing, are empty which means that there are still gaps in knowledge acquisition approaches and we need to find out and develop a set of techniques and tools to solve our extraction issue which is described in a dedicated section.

IV. Overview of scraping parser generating tool

As far as no existing tool or techniques has satisfied our task requirements and writing a dedicated parser for each web resource containing data to be imported to thesaurus we populate is a really time-consuming procedure, a way to combine possible diversity of data and little time consumption was found which supposes a tool for generating parsers automatically based on user input and data structure. Proposed approach seems reasonable as time consumed by ParsGen (URL: https://github.com/m-lapaev/parsgen) development and testing process essentially falls behind the time possibly required for manual copy-and-paste approach or developing a separate parser for each data source.

A. ParsGen overview

The tool we developed and propose is a web-scraping tool providing functionality of generating a parser for most of catalogue-like web data sources. The tool is developed at the intersection point of Java technologies and web-technologies based on DOM-model parsing. The process of configuration takes not longer than 5-10 minutes of manual work and is based on providing a tool with web-catalogue URL, page walk-around techniques and selected pieces of data to be extracted from every page structure as well as storage settings including database connection settings, table schema and data-types of extracted data. The output of configuration process is a set of parser Java classes (Fig. 3) (Entity class as a model of entities extracted and the Parser class including executive part and page walk-around method as well as a class for interactions with database). The set of classes for any resource is unified so that any parser generated for any catalogue resource have the same interface and can be easily integrated into other pipeline for solving subsidiary tasks giving an ability to inject any generated parsing module into the same work-flow, thus, the parser class set appears to be a plug-in for pipeline work-flow.

Main idea of the tool is to make it possible to code a parser without real coding and to allow users having a little programming knowledge to get use of heterogeneous catalogue-like resource data extraction as well as to reduce time consumption for thesaurus mining for out project by bringing the process to simple mouse clicking and selection of data we need at an example page with next automated extraction. So, the complete process of extraction looks as follows:

1) choice of data source: a catalogue-like web resource;
2) analysis of data we need;
3) providing the tool with database connection settings;

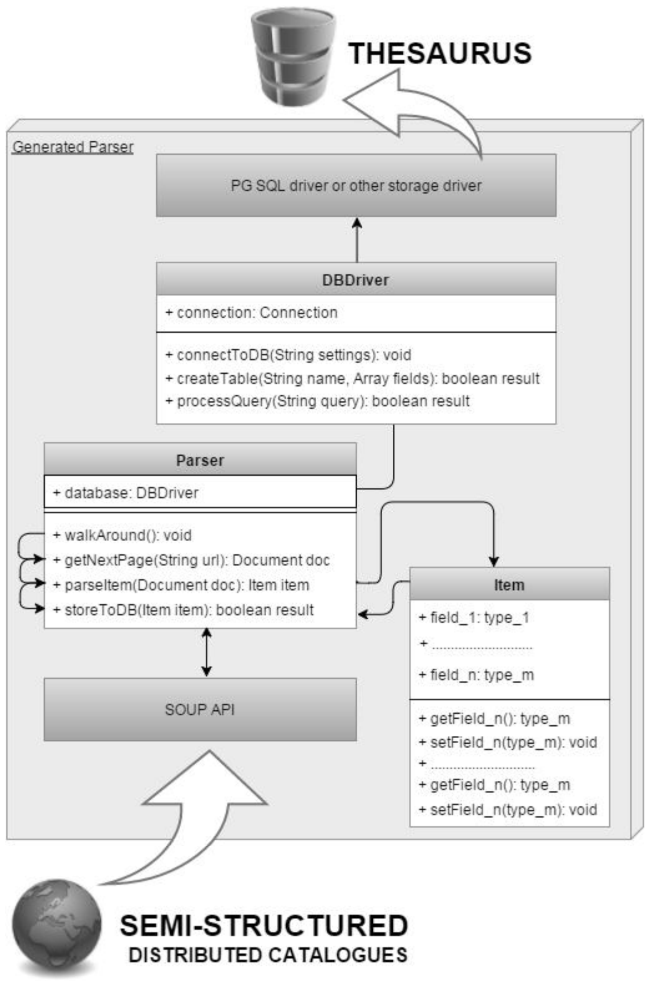

THESAURUS
Generated Parser
PG SQL driver or other storage driver
DBDriver
+ connection: Connection
+ connectToDB(String settings): void
+ createTable(String name, Array fields): boolean result
+ processQuery(String query): boolean result
Parser
+ database: DBDriver
+ walkAround(): void
+ getNextPage(String url): Document doc
+ parseItem(Document doc): Item item
+ storeToDB(Item item): boolean result
Item
+ field_1: type_1
+ ....
+ field_n: type_m
+ getField_n(): type_m
+ setField_n(type_m): void
+ ....
+ getField_n(): type_m
+ setField_n(type_m): void
SOUP API
SEMI-STRUCTURED
DISTRIBUTED CATALOGUES

Fig. 3. Model of generated by the tool parser Java-classes and its class relation

4) constructing a database table by providing the tool with field names, data-types and length restrictions;
5) selecting an example page which may be any resource page of data-source;
6) selecting nodes containing required data within a document tree displayed in the screen;
7) selecting exclusion methods from the list of available methods;
8) repeating step 7 for each data item we need;
9) providing the tool with page walk-around techniques;
10) proceeding to next step when a parser is already generated and database tables are already created;
11) launching the parser and waiting while all pages are examined and all required data is extracted into a database;
12) proceeding to next resource.

As one may conclude, ParsGen is not a web data extractor, but a tool that collects analysed data from user and generates a parser from the user-provided configuration based on his/her preferences and needs and at the same time it is just what we need to process a number of distributed and heterogeneous concepts and alternative synonymic terms as the time required for generating a parser is dramatically less than for coding and debugging the same parser manually.

The tool provides a choice of four data-types for database fields at a database construction step which are the most common ones for extracted data: integer numbers, real floating-point numbers, text strings (character varying in terms of database, can be restricted to a provided length) and, finally, boolean data-type. Moreover, a non-null-value restriction may be selected for any of the fields in case some significant fields are not provided for some entities within a semis-structured source of data. As well as data-type definition, the tool provides a choice of extraction method among few options which are:

- by attribute values or class value prefix;
- by attribute value or class value suffix;
- by attribute value or class value containing a specified string entry;
- by structure items' ids;
- by structure items' names;
- and by style containing a specified property.

We also offer a page walk-around method among four possible choices: analyze pages having numbered URLs within a specified numeric interval; analyze all pages matching a regular expression within the 'table of content' page, analyze all pages provided by the user and, finally, recursive analysis of URLs matching a regular expression provided starting from specified page. These are all the information required from the user. In next subsection we demonstrate one of the use-cases which was used as one of test-cases at the validation and verification stage.

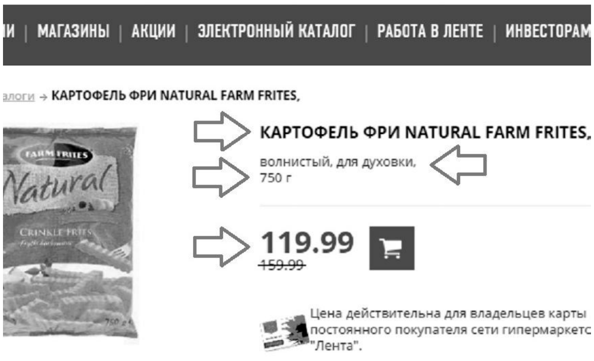
B. ParsGen validation and verification

In order to test the tool a set of catalogue-like semi-structured sources was analyzed and data extracted. As an example we are providing an extraction process of special offers catalogue of one of chain of supermarkets well-known in St.Petersburg and in the territory of Russia. As a first stage of experiment we analyzed content of an ordinary catalogue's page to find out which fields can be extracted for each entity and found out that we may want to store the following data: good's name, good's old price, good's special price and brief description (Fig. 4).

As far as the data required is stated, we define that two of the fields are of type String and two elements are of floating-point type. No we may launch ParsGen and start configuration process which includes database settings (Fig. 5) and table fields restriction (Fig. 6) as the first stage.

As soon as the construction process is over, the table structure is displayed to let the user make sure that all restrictions are provided correctly. If not, editing, removal and addition of fields are possible. If all data is provided correctly, database table is generated (Fig. 7).

Now, when environment structure is defined, environment is generated and an example page is provided, a page node tree is displayed. The user traverses the tree and selects the nodes containing required data and applies one of supported extraction options for each selected node (Fig. 8). Generated code as well as result of code execution is presented to check whether option is applied correctly. Option choice is to be done for every data field provided at database table construction stage.


И | МАГАЗИНЫ | АКЦИИ | ЭЛЕКТРОННЫЙ КАТАЛОГ | РАБОТА В ЛЕНТЕ | ИНВЕСТОРАМ
FARM FRITES
Natural
CRINKLE FRITES

…алоги → КАРТОФЕЛЬ ФРИ NATURAL FARM FRITES,

КАРТОФЕЛЬ ФРИ NATURAL FARM FRITES,

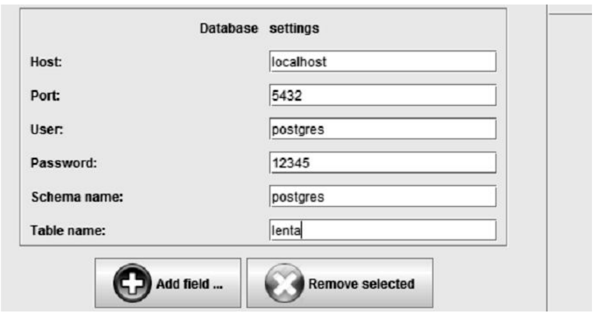
волнистый, для духовки, 750 г

119.99

~~159.99~~

Цена действительна для владельцев карты постоянного покупателя сети гипермаркетс "Лента".

Fig. 4. General look of the catalogue item page



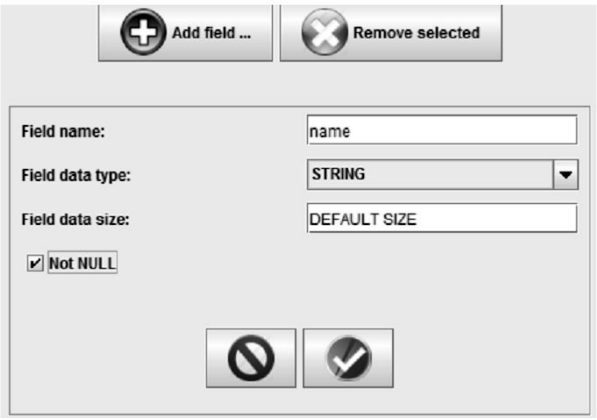
Database settings

Host:	localhost
Port:	5432
User:	postgres
Password:	12345
Schema name:	postgres
Table name:	lenta

Fig. 5. Database connection settings form

The next screen of the wizard (Fig. 9) is presented to user to display all generated extracting code and let the user check whether unoptioned fields are left. As soon as page walk-around techniques and settings are provided, parses classes (pipeline work-flow plug-in) is generated and may be launched to start the automated extraction process.

Experimental verification based on applicability analysis has shown that tool coped with provided tasks successfully: generated classes (URL: https://github.com/m-lapaev/genclasses.git) had been compiled with no errors and populated a database with information on requested entities from around 10 various catalogues including other supermarket catalogues, IKEA catalogue and food additives on-line reference. The other significant feature is that generated code is well-structured and human-readable, i.e. is represented not as a mess of symbols, but as code following Java code convention in most of the cases which was, actually, intended for better manual validation. As a results for presented here experimental verification, a total of 500 special offer goods were extracted which did not confront with the available in catalogue information on the day of test. All fields appeared fields at a database construction step which are the most some items did not have brief description in catalogue which was confirmed during manual overview of URLs causing questions. The tool was also concluded to be efficient in terms



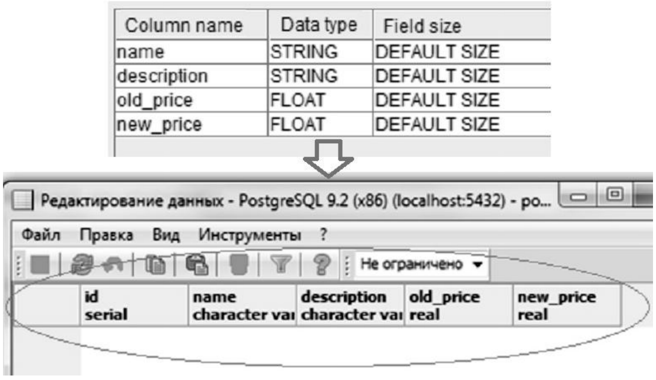
Field name: name

Field data type: STRING

Field data size: DEFAULT SIZE

☑ Not NULL

Fig. 6. Constructing a table: fields' restrictions settings



Column name	Data type	Field size
name	STRING	DEFAULT SIZE
description	STRING	DEFAULT SIZE
old_price	FLOAT	DEFAULT SIZE
new_price	FLOAT	DEFAULT SIZE

Редактирование данных - PostgreSQL 9.2 (x86) (localhost:5432) - po...
Файл Правка Вид Инструменты ?
Не ограничено

id	name	description	old_price	new_price
serial	character vai	character vai	real	real

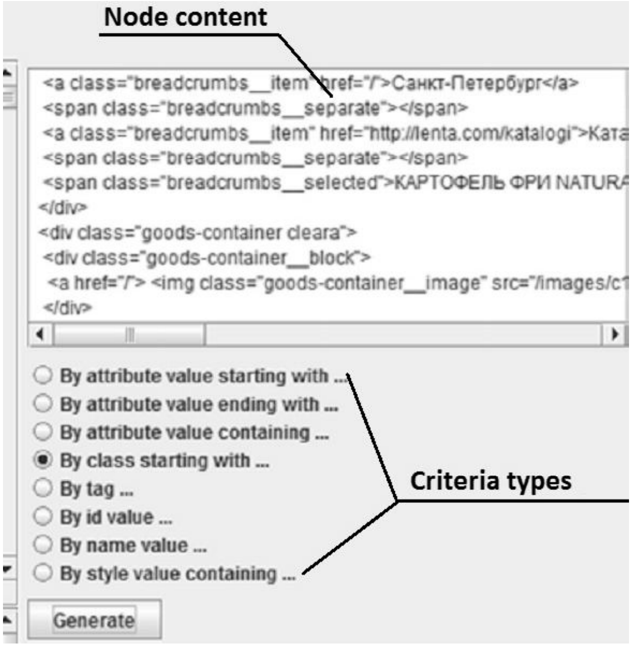
Fig. 7. Generated database table with further entity extraction

of time consumption of extraction task. Though, we did not address the issues of ethics and copyright, thus, they have to be considered and tool must be used with care not to violate laws of authorship.

V. THESAURUS-DRIVEN DATA-SET POPULATION OVERVIEW

The tool developed and described above was used for data-set refinement by means of thesaurus (or glossary) of ingredients and food additives collected from various data-sources [18]. The main issue of data-set and graph of FOODpedia project (URL: http://foodpedia.tk) is concept inconsistency and rather low degree of linking, especially for the reason of synonymy of ingredient items and E-additives which may be overcame only by means of concept matching.

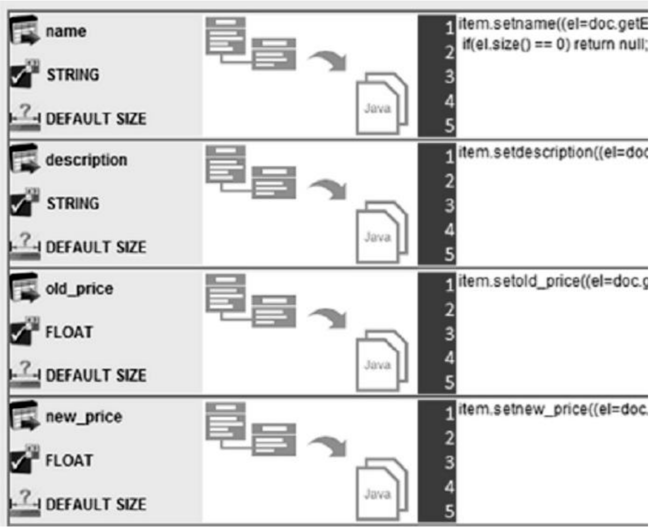
General idea of the approach is to introduce a separate matching module into main pipeline of food data extraction work-flow with thesaurus as a core unit of matching module. Concept matching not only will refine terms and bring them to unity but will increase interlinking with other data-sets such as AGROVOC and DBpedia as the FOODpedia data-set of over 63000 food products met on the shelves of supermarkets in Russia still contains 22803 ingredient terms without references to corresponding concepts wich is not possible without concept matching.



```
<a class="breadcrumbs__item" href="/">Санкт-Петербург</a>
<span class="breadcrumbs__separate"></span>
<a class="breadcrumbs__item" href="http://lenta.com/katalogi">Ката
<span class="breadcrumbs__separate"></span>
<span class="breadcrumbs__selected">КАРТОФЕЛЬ ФРИ NATURA
</div>
<div class="goods-container cleara">
<div class="goods-container__block">
 <a href="/"> <img class="goods-container__image" src="/images/c1
</div>
```

○ By attribute value starting with ...
○ By attribute value ending with ...
○ By attribute value containing ...
● By class starting with ...
○ By tag ...
○ By id value ...
○ By name value ...
○ By style value containing ...

Fig. 8. Node content representation and criteria type selection



name
STRING
DEFAULT SIZE

```
item.setname((el=doc.getE
 if(el.size() == 0) return null;
```

description
STRING
DEFAULT SIZE

```
item.setdescription((el=doc
```

old_price
FLOAT
DEFAULT SIZE

```
item.setold_price((el=doc.g
```

new_price
FLOAT
DEFAULT SIZE

```
item.setnew_price((el=doc.
```

Fig. 9. Fields and generated code representation

The most difficult to match are lexically messy and unanalysable concepts and terms such as E330, e330, e-330, citric acid and other alternative names, E100, e100, e-100, curcumin and other alternative names and lots of other examples with a large set of alternative names. Application of thesaurus-driven matching has shown fare results and increased percentage of referencing to other data-sets from 70% up to 85% only by matching E-additives and will increase even more after matching ingredients other than food additives. A pipeline extraction process (Fig. 10), especially matching process was observed with a help of tracing of matches found (Alg. 1) and analyzing them for correctness. The observations evidences of fare and valid matching of raw entries with URIs within FOODpedia resources and mapping them to DBpedia exactly matching URIs which witnesses of both disambiguation and concept matching as well as entity mapping to be correct.

Algorithm 1 Matcher tracing

```
MATCHED: ANATO --> E160B
link to
   <http://foodpedia.tk/resource/E160b>
   skos:exactMatch <http://dbpedia.org
        /page/Annatto>
MATCHED: SOY LECITHIN --> E322
link to
   <http://foodpedia.tk/resource/E322>
   skos: <http://dbpedia.org
        /page/Lecithin>
MATCHED: BIXIN --> E160B
link to
   <http://foodpedia.tk/resource/E160b>
   skos:exactMatch <http://dbpedia.org
        /page/Annatto>
MATCHED: NATURAL DYE CURCUMIN --> E100
link to
   <http://foodpedia.tk/resource/E100>
   skos:exactMatch <http://dbpedia.org
        /page/Curcumin>
MATCHED: CHLOROPHYLLIN --> E140
link to
   <http://foodpedia.tk/resource/E140>
   skos:exactMatch <http://dbpedia.org
        /page/Chlorophyll>
MATCHED: BETA-CAROTENE --> E160A
link to
   <http://foodpedia.tk/resource/E160a>
   skos:exactMatch <http://dbpedia.org
        /page/Carotene>
MATCHED: YELLOW SUNSET --> E110
link to
   <http://foodpedia.tk/resource/E110>
   skos:exactMatch <http://dbpedia.org
        /page/Sunset_Yellow_FCF>
```

VI. Conclusion and future work

Populating a knowledge base and ontology learning is a long and iterative process requiring application of all sorts of techniques. Some of the methods are already proposed by researchers, others are still under research, however each of existing techniques fit only particular types of problems and still there is no method or tool suitable for every problem of knowledge extraction and refinement. This paper presented a brief overview of existing techniques and proposed an uncommon data extraction and knowledge structuring method and its technical aspects. The analysis of data-set from a semi-structured homogeneous data source has shown a drawback of knowledge consistency caused by synonymy and ambiguity: many lexically non similar terms match the same concept. The drawback have been overcome by introducing a thesaurus-driven concept matching component into a main pipeline of extraction process.

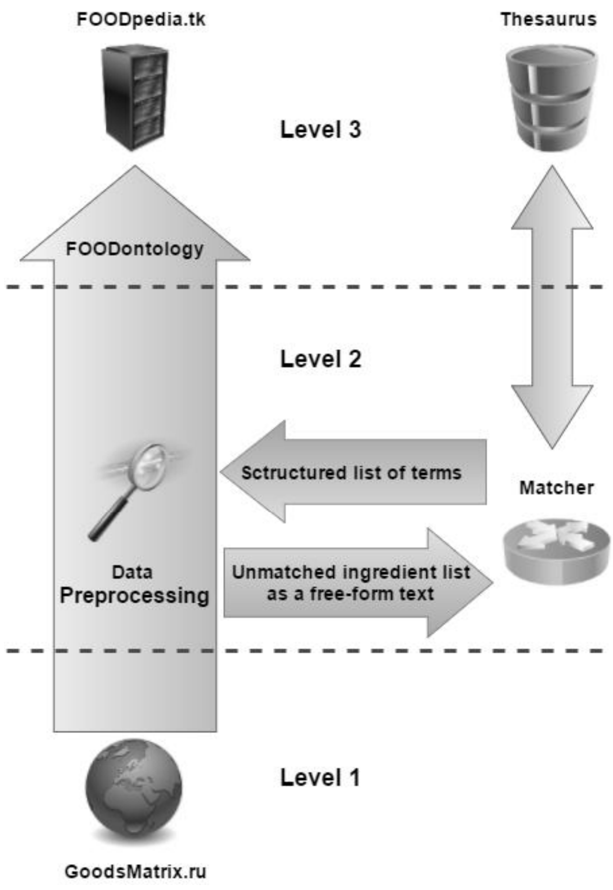


Fig. 10. Extraction pipeline with introduced thesaurus-driven matching mechanism, Three levels are: 1 - semi-structured and unstructured textual data, 2 - structured data, 3 - structured database and knowledge base

In this study we showed that introduced techniques are applicable for the issue described and have a fare efficiency which have been proved numerically. Along with the techniques we introduced a tool that dramatically simplifies process of thesaurus extraction from heterogeneous data sources and verified tools effectiveness experimentally for a number of on-line catalogues. Validation and verification has shown that using the introduced tool for generating parser rather than coding a separate parser for each data source sufficiently decreases time consumption required for knowledge acquisition.

Despite the fact that designed tool and proposed techniques have shown efficiency, further revision and researches are desirable. The tool is still suitable only for semi-structured existing techniques fit only particular types of problems and data sources of other types are considered. Moreover, analysis of other ways to refine the data-set is needed as well as publication of ontology is desired.

Acknowledgment

We would like to show our gratitude to the all publicly available references and catalogues for sharing the data we used for verification purposes. We are also immensely grateful to developers and contributors of GoodsMatrix.ru and Dobavkam.NET for the data that became a part of FOODpedia existing techniques fit only particular types of problems and Ministry of Education and Science of the Russian Federation (Grant #RFMEFI57514X0101).

References

[1] P. Hitzler, M. Krtzsch, S. Rudolph, ”Foundations of Semantic Web Technologies”, CRC Press, 2009

[2] D. Mouromtsev, P. Haase, E. Cherny, D. Pavlov, A. Andreev, A. Spiridonova, ”Towards the Russian linked culture cloud: data enrichment and publishing.” The Semantic Web. Latest Advances and New Domains. Springer International Publishing, 2015. 637-651.

[3] D. Snchez, M. Batet, D. Isern, A. Valls, “Ontology-based semantic similarity: A new feature-based approach“, *Expert systems with applications*, vol.38(9), 2012, pp. 77187728.

[4] R. Shah, S. Jain, “Ontology-based information extraction: an overview and a study of different approaches“, *International Journal of Computer Applications*, vol.87(4), 2014, pp. 68.

[5] A. Siham, M. Sihem, “Syntactico-semantic algorithm for automatic ontology merging“, *Information Technology and e-Services (ICITeS)*, 2012, pp. 15.

[6] P. Petrov, M. Krachounov, E. A. A. van Ophuizen, D. Vassilev, “An algorithmic approach to inferring cross-ontology links while mapping anatomical ontologies“, *Serdica Journal of Computing*, vol.6, 2012, pp. 309332.

[7] V. Bharanipriya, V. K. Prasad, “Web content mining tools: a comparative study“, *International Journal of Information Technology and Knowledge Management*, vol.4(1), 2011, pp. 211-215.

[8] M. Ambika, K. Latha, “Web mining: the demystification of multifarious aspects“, *International Review on Computers and Software (I.RE.CO.S.)*, vol.9(1), 2014, pp. 135141.

[9] M. Kejriwal, D. P. Miranker, “Semi-supervised instance matching using boosted classifiers“, *The Semantic Web. Latest Advances and New Domains*, vol.9088, 2015, pp. 388-402.

[10] H. Ren, Ze-Nian Li, “Basis mapping based boosting for object detection“, *Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1583 - 1591.

[11] Long Philip M, Servedio Rocco A, “Random classification noise defeats all convex potential boosters“, *Machine Learning (Springer US)*, vol.78(3), 2010, pp. 287304.

[12] M. Alsarem, P.-E. Portier, S. Calabretto, H. Kosch, “Ranking entities in the age of two webs, an application to semantic snippets“, *The Semantic Web. Latest Advances and New Domains*, vol.9088, 2015, pp. 541-555.

[13] A. Stellato , S. Rajbhandari, A. Turbati, M. Fiorelli, C. Caracciolo, T. Lorenzetti, J. Keizer, M. Teresa Pazienza, “VocBench: a web application for collaborative development of multilingual thesauri“, *The Semantic Web. Latest Advances and New Domains*, vol.9088, 2015, pp. 38-53.

[14] A. Rettinger , A. Schumilin, S. Thoma, B. Ell, “Learning a cross-lingual semantic representation of relations expressed in text“, *The Semantic Web. Latest Advances and New Domains*, vol.9088, 2015, pp. 337-352.

[15] A. Moro, A. Raganato, R. Navigli, “Entity linking meets word sense disambiguation: a unified approach“, *Transactions of the Association for Computational Linguistics*, vol.2, 2014, pp. 231244.

[16] W. Shen, J. Wang, J. Han, “Entity linking with a knowledge base: issues, techniques, and solutions“, *Knowledge and Data Engineering*, vol.27(2), 2014, pp. 443 - 460.

[17] G. Wohlgenannt, “Leveraging and balancing heterogeneous sources of evidence in ontology learning“, *The Semantic Web. Latest Advances and New Domains*, vol.9088, 2015, pp. 54-68.

[18] M. Kolchin, A. Chistyakov, M. Lapaev, R. Khaydarova, “FOODpedia: Russian food products as a linked data dataset“, *The Semantic Web: ESWC 2015 Satellite Events*, vol.9341, 2015, pp. 87-90.