Automated Extraction of Values of Quantitative Indicators to a Quality Evaluation System Using Natural Language Analysis Tools

Mariya Zhekova[1], George Pashev[2], George Totkov[3], Silvia Gaftandzhieva[2]

[1] *University of Food Technologies, 26 Maritsa Blvd., 4000 Plovdiv, Bulgaria*
[2] *University of Plovdiv "Paisii Hilendarski", 24 Tzar Assen Str., 4000 Plovdiv, Bulgaria*
[3] *National Evaluation and Accreditation Agency, 125, Tzarigradsko Chaussee Blvd., bl. 5, 1133 Sofia, Bulgaria*

Abstract

The objective evaluation of qualitative indicators requires the extraction and processing of huge amounts of data. The paper aims to propose a model for extracting values of quantitative indicators to a quality evaluation system with the help of natural language analysis. It presents an algorithm and the development of a software module, which use natural language analysis to solve the task for extraction of value for evaluation of quantitative indicators from quality evaluation system and answering user questions asked in natural language. All this happens with a natural language interface and free text processing without the help of controls, tools, and a graphical user interface. In the presented software model, the task is to provide an opportunity to extract information by synthesizing a SQL query to a database, using pre-created templates, from a user-asked question in natural language (Bulgarian). The study is part of a study dedicated to the quality evaluation of higher education in Bulgaria.

Keywords

natural language comprehension, natural language synthesis, natural language analysis, generation, question-answer systems, information retrieval

1 Introduction

The quality evaluation of objects in different subject areas is based on a large number of qualitative and quantitative indicators. The objective evaluation of qualitative indicators requires the extraction and processing of huge amounts of data.

The quality evaluation of Higher Education Institutions (HEIs) has to be carried out periodically and to reflect the results of processes and states of objects in different periods. Monitoring of the procedures and activities related to the quality evaluation in higher education involves the collection of a huge amount of data from institutional information and management systems of the HEI (e.g. Student information systems, Systems for candidate student campaigns, E-learning environments, Human resource systems, Academic staff development systems, Research reporting systems, etc.).

In recent years, a number of studies have been conducted to automate (self) assessment and quality evaluation in higher education. For example, the quality system of the University of Graz (Austria) generates a large amount of data allowing quality monitoring [1]. A web-based application for monitoring academic performance in real-time based on business intelligence and service-oriented architecture has been developed in Indonesia [2]. The system of the Arab International University

Education and Research in the Information Society, September 27–28, 2021, Plovdiv, Bulgaria
EMAIL: m_jekova@uft-plovdiv.bg; georgepashev@gmail.com, totkov@uni-plovdiv.bg, sissiy88@uni-plovdiv.bg
ORCID: 0000-0002-2501-4412; 0000-0001-8148-4737, 0000-0001-9413-3565, 0000-0002-0569-9776

CEUR Workshop Proceedings http://ceur-ws.org ISSN 1613-0073

extracts and aggregates data from quality assurance and management systems for training, human resources and finance [3].

Some experiments for automated data retrieval from various information systems to ensure and evaluate the quality of training have been conducted in Bulgaria [4, 5]. Among them are experiments for extraction of data on the attendance and activity of the students from the e-learning system; data on students, curricula of specialities and curricula of courses in the management system of the educational process; data on the competencies of the lecturers from the system for development of the academic staff, etc.

In regards to the external evaluation from NEAA, studies for automated extraction and analysis of data (e.g. for teaching materials, infrastructure, e-learning environment, communication and collaboration tools, student assessment system, flexibility and adaptability of the learning process, student support, team qualification, etc.) for assessing the quality of distance learning [4] have been conducted. Models, methodologies and software tools suitable for automating processes for dynamic quality evaluation in higher education (education, science, management, etc.) are proposed, studied and tested [6, 7, 8]. A general model of a process for dynamic quality evaluation accompanied by architecture and a prototype of a software system based on institutional information infrastructure have been proposed. Based on the model and the software prototype SIDOC a general model for the dynamic accumulation and aggregation of data needed to assess quality in higher education is created [9]. The proposed model is specified in the case of institutional accreditation of Bulgarian HEIs, and experiments for data aggregation in the automated quality assessment have been conducted.

The paper aims to propose a software model for extracting values of quantitative indicators to a quality evaluation system with the help of natural language analysis.

The corresponding computer model for extracting and translating data from a user question in natural language to an SQL query to a database implies building a basic data model, "filled" with grammar content, extracted from various sources and corresponding to modern Bulgarian grammar.

The choice of a relational database model is based on the specific software implementation proposed below, based on the SQL pattern matching & Connection Approach. We are aware that the use of NoSQL baseline can eliminate the conversion of natural language text to SQL query, but then other approaches work, including statistical and machine learning, which is studied in other articles of the authors (together and separately).

2 Basic information structure

The available metadata in the basic structure to which the proposed software model makes its references can be divided into several groups: a relational database of the organization, including general nomenclatures, a corpus of linguistically annotated concepts for the field, a set of control models (compliance models) and an available file system to prove the timeliness of the data. The information handled by the system (e.g. users, procedures, standards, criteria, quantitative and qualitative indicators) is stored in a relational database, together with the attached evidence files and the linguistic corpus. The attached files are organized in a file system with a defined structure. The database store links to the files' names and their paths in the file system and allows searching in the documents later. It is permissible for the files to be read-only to prevent the renaming or deletion of evidence already submitted to a procedure. The linguistic corpus is organized and composed of several ontologies, providing a hierarchical structure of objects in the subject area (words and phrases from indicators). The linguistic corpus is a unified vocabulary with annotations for the accompanying linguistic characteristics (metadata) of the considered fragments.

The separate groups of basic structures can exist and be administered, both together and separately. They can be analysed, compared and integrated. They can be superimposed and restructured if necessary to achieve the desired result. Some of them do not depend on the nature of the subject area, others are semi-automated, and others are annotated and filled in manually.

2.1 Design of the conceptual model of the relational database

In artificial intelligence systems, the presentation of the database in the form of a conceptual model is close to the way concepts are stored in the human brain. The last provides an opportunity for modularity, flexibility and data hierarchy [10]. This characteristic is also the foundation on which the subsequent considerations in the study are based.

The first task in designing a database is all elements, subject to evaluation and accreditation procedures be determined. The set of elements in the subject area "Quality Evaluation in Bulgarian Higher Education" includes objects (curricula, study programmes, disciplines, surveys, etc.), subjects (teachers, guest lecturers, prominent practitioners, etc.), processes and activities (incl. digitalized), information systems and resources (incl. electronic), etc. The individual elements and relationships between them have their life cycle – some elements/relationships are removed or archived, others are adapted or changed, and others are created (intentionally or accidentally) and included in P. We will denote with P the set of states of the elements in the subject area in time. Elements and the relations between them are diverse and have a different set of characteristics (attributes). Together they represent the so-called conceptual map of P.

Data modelling is the first step in the database design process. It is a high-level design or design phase on an abstract level. On this phase should be described *elements of the subject area, which the database contains* (e.g. Teachers, Disciplines, Students, Curricula, etc.), *the relations between objects and subjects* (e.g. Lecturer – PhD students, Lecturer – Disciplines), *data restrictions* (specific data type requirements).

The main element for the development of the natural language tool for automatic extraction is the class. Data, such as columns in a table, data types, calculation fields, and interface fields, are specified by their class in the application code. One class groups elements with similar properties, similar basic parameters, and relative significance in the real world.

Objects such as "*Teachers*", "*Study Programmes*", "*Students*", and others have a place in the developed model. Each of these objects has similar properties, but there is a significant difference in its role and purpose. Accordingly, the data related to these objects are processed differently, despite the similarities (e.g. some of them are persons). On the one hand, these objects can be divided into several separate classes. On the other hand, it is more convenient to work with aggregated data, which improves the system performance and helps in analysis and decision making.

The data used during the development of the model are objects in the field of quality evaluation of higher education – universities, professional fields, students, teachers, disciplines, curricula and programs. It is appropriate for these objects to be grouped in base classes and successors based on hierarchical and classification dependencies between them. Teachers can be grouped into a base class "Teachers" and successor classes "Habilitated", "Non-habilitated", "Occupied a new academic position", "Guest teachers", etc., depending on some of their distinguishing attributes. The exams can be grouped in the base class "Exam" with successors "Candidate-student exam", "State exam", etc.

The summarizing of data based on their significance in the real world is the basis for defining classes of objects in the software implementation, and later as a factor in the formation of the response from the system.

Table 1. List of study programmes distinguished as successors of the base class Study Programmes

Study programmes in professional field/majors from the regulated professions	
SubType	**1:** with new and/or updated curricula – adopted as a result of an analysis of the needs and expectations of various stakeholders (students, business representatives, professional organizations)
	2: conducted in a foreign language
	3: with training conducted jointly with other Bulgarian HEIs
	4: with training conducted jointly with foreign universities
	5: with qualification characteristics and curriculum, to which public electronic access is provided for students and authorized users
	6: new
	7: with diplomas of graduates certified by foreign HEIs
	8: for which information on the prospects for professional realization is available on the website of higher education institution

9: for which the qualification characteristics and the curriculum are published on the website of higher education institution, and for each discipline in the study programme – the annotation and the description of the forms for testing and assessment

10: in the audit of which trainees and employers participated

11: for which data are collected and analysed every school year – available and discussed by authorized users

12: changed as a result of analysis and implementation of an action plan (or package of measures) in the period adopted as a result of external audit recommendations

In the examples given in Table 1, the linguistic fragments are part of various indicators, the quantitative values of which the system can search for answers if the user asks a question correctly. Analogous to the way shown in Table 1, the abstractions in the subject area can be combined into categories groups and then build a linguistic tree of concepts, so-called a tree of indicators for each of which some value is kept in the database of the quality evaluation system.

Let's look at an example on indicator 4.2. Relative amount (in percentages) of courses (relative to their total number) conducted in a foreign language.

If a user question is asked on indicator 4.2. e.g. – "What is the percentage of disciplines conducted in a foreign language?" – from the system is expected to derive a numerical value of the number of disciplines studied in a foreign language, which is divided by the total number of disciplines in the professional area.

In the developed model the number of the basic concepts are the mechanisms for obtaining the relative share or the percentage content of the searched subcategories of objects.

Let us denote by N_{ind} the set of quantitative indicators. The calculation of the values of quantitative indicators of a given type, as well as the corresponding evidence to them, follow a scheme specific to the respective type.

The indicators are pre-categorized and typed according to the method of calculation and obtaining their numerical value.

Specifically, this example and the grammatical structure of the user string thus set show that these are quantitative indicators of Type 1, namely:

Relative share (in percent) of the number of elements in the set S (relative to their total number in the subset S of PN / SRP), which have property P (for which P is a true predicate).

The value of an indicator of this type in which a percentage or relative amount is sought is determined as follows:

$Val(n) = 100\frac{|\{e\in S:P(e)=1\}|}{|S|}$, where n – quantitative indicator, Val (n) – its numerical amount.

Table 2. List of indicators and their numerical dimensions

ID	Indicator Name	Value
1	Number of candidate-trainees with 1st desire for a study programme for one announced place	12
2	Number of incoming mobility of trainees lasting at least 1 month	4
3	Number of scientific forums and creative events with the participation of students in the professional field/major from the regulated profession organized by HEIs	8
4	Study programmes	32
5	Study programmes with diplomas of graduates certified by foreign universities	2
6	Disciplines	212
7	Disciplines conducted in a foreign language	7
8	Trainees	450
9	Trainees enrolled with foreign diplomas	7
10	Trainees who participated in forms of practical training according to the curriculum, conducted in a real work environment (including under the project "Student Internships" of the Ministry of Education and Science)	16
11	Trainees who participated in outgoing mobility abroad for at least 1 month	3
12	Trainees with publications or creative performances in the field of the professional field/major from the regulated profession	29
13	Trainees who participated in scientific forums or creative projects (conferences, round tables, seminars, etc.)	29
14	Trainees involved in research projects	2
15	Trainees in the last school year, incl. interrupted	103

16	Trainees who in the next school year continue their education in a higher course of the study programme	350
17	Graduates	
18	Graduates who have continued their studies at the educational qualification degree or educational and scientific degree under a joint program with another HEI	14
19	Research projects	12
20	Projects with the participation of trainees	20

2.2 Design of the linguistic corpus for the subject area

Nowadays, a database in an information system is considered one of the fundamental sources of information. The data stored in it is accessible using SQL queries. To retrieve information from the database, the field of knowledge that it covers must be represented formally and declaratively. The representation includes a glossary (or list of constants) for denoting terms (words and phrases) in the field, labelling with metadata, and placing constraints on the set of terms, logical relations (correspondences) that limit their interpretation, and how they relate to each other. The first step in the process of natural language processing is the identification of the connection between the recognized language objects. Then follows their transformation into SQL fragments, from which to generate a query to the database. The automation of these steps is one of the tasks for the integrity of the various data models needed to solve the natural language comprehension problem.

Building a tree (vocabulary) of concepts is feasible, as the structure recreates real objects and situations for the subject area. The concepts network is filled with standard terminology from the dictionary of the contemporary Bulgarian language. The concepts network allows data integration and avoids duplication.

The Bulgarian language consists of linguistic objects that form the linguistic reality [11]. In the PhD research, the linguistic corpus (within the conceptual graph model of the subject area) is upgraded with a level of abstraction through a system of metadata that reflects the properties and relationships between language objects. According to Boyadzhiev, the concept is a sign, an image, and it is created through the perception of real phenomena and facts and analysis of their signs [12].

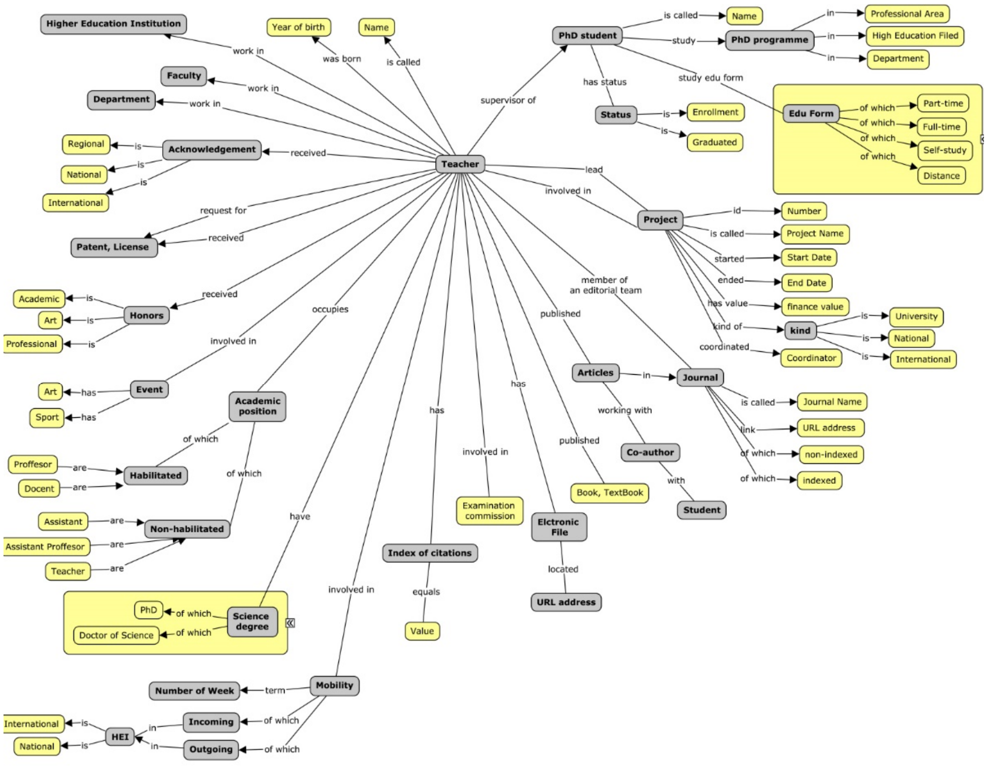


Figure 1. Graph with words and phrases, part of a criteria system for quality evaluation of HEIs

Not every language unit names a concept. Official words (prepositions, conjunctions and particles), interjections and pronouns do not correlate with the concepts, and no questions can be asked about them. The official words only connect the words in the sentence. Interjections are side, immutable words that express feelings and resemble sounds. Pronouns, in turn, only replace and indicate objects and persons. For this study, we exclude the three groups of linguistic units from reasoning. In the abstract linguistic corpus, a list of the so-called immutable parts of speech is in table dbo.Lexicon_Prepositions, in dbo.N_PrepositionType is the type nomenclature of these invariant parts of speech (pronoun, preposition, union, particle). In the application code, there is a method that checks whether any of the received in the input user question tokens is a pretext.

Table 3. Nomenclature of official, unchangeable words

ID	BG	Stopwords	ID	BG	Stopwords	ID	BG	Stopwords	ID	BG	Stopwords
1	аз	i	24	горе	up	47	има	have	70	с/със	with
2	на мен	me	25	долу	down	48	има	has	71	относно	about
3	мое	my	26	в/във	in	49	имал	had	72	против	against
4	себе си	myself	27	вън	out	50	имам	having	73	между	between
5	ние	we	28	в/на/към	on	51	правя	do	74	във	into
6	ние	our	29	от/за	off	52	прави	does	75	през	through
7	наши	ours	30	върху	over	53	правила	did	76	по време	during
8	себе си	ourselves	31	под	under	54	прави	doing	77	по-	more
9	ти	you	32	отново	again	55	един/една	a	78	най-	most
10	вие	your	33	нататък	further	56	един/една	an	79	друг	other
11	ваши	yours	34	тогава	then	57	-ът	the	80	малко	some
12	вие	yourself	35	веднъж	once	58	и	and	81	като	such
13	самите	yourselves	36	тук	here	59	но	but	82	не	no
14	той	he	37	там	there	60	ако	if	83	нито	nor
15	него	him	38	кога	when	61	или	or	84	негов	itself
16	негово	his	39	къде	where	62	защото	because	85	те	they
17	негов	himself	40	защо	why	63	като	as	86	на тях	them
18	тя	she	41	как	how	64	докато	until	87	техен	their
19	нея	her	42	всички	all	65	докато	while	88	техни	theirs
20	нейно	hers	43	някой	any	66	от	of	89	тях самите	themselves
21	нейн	herself	44	заедно	both	67	на	at	90	какво	what
22	то	it	45	поотделно	each	68	при	by			
23	негов	its	46	няколко	few	69	за	for			

The vocabulary resource, as a set of words and phrases, in addition to the annotation with basic grammatical classes and categories, also offers to fill in semantic ones bearing tags/markers with information about the use, meaning and significance of the words combined in synonymous lines. For example, if the terms X and Y are interchangeable with each other (i.e. a result which contain X can be accepted in response to a request containing Y, this will be because they have the same meaning). In this way, classification of terms uniting them into grammatical classes, categories, synonymous lines, semantic classes of equivalence and others, can be built automatically.

Later in the natural language analysis, if a word from the user query text is missing in the tree of concepts, it can be replaced with another from the same synonymous order in which it occurs or another word belonging to the same class. The word/phrase recognition strategy works by matching keywords (classes). This strategy increases the relevance of the answer received.

Comparing recognized parameters from the Bulgarian language text is a process of creating a parallel database of dependencies. The parallel database corresponds to two different models – the linguistic (indicator tree) and the relational model of the existing database. The dependencies obtained in this correspondence add integrity to the overall process of natural language recognition and processing. A kind of data transformation or mediation between the source is performed – these are the primary data obtained from the user question in Bulgarian language and the purpose, which is to extract an answer from the database. This can be done if there are links between the formal language (SQL) that manipulates and retrieves data from a database and the primary data (explicit and implicit), namely words and phrases identified by the question or hidden, encoded in the grammar structure of the information question

2.3 A set of conformity control models

The set of conformity control models is developed based on a methodology for developing a system of matching templates for queries in natural language [13].

As part of the tasks set here, after a linguistically established system for describing and classifying the concepts included in the scope of the subject area, are the creation of a methodology for recognizing, tagging and comparing them. Some templates for correspondence between the recognized language units and SQL constructs and the control models are already described in the previous paper [13], e.g. control models in the form of correspondence tables, with the help of which it is possible to convert one language resource into another according to a given scheme. An algorithm for converting user search into an understandable relational database query is proposed.

At the beginning of the study, a small number of more general (abstract) rules, or a model with less accuracy, were created. Subsequently, the model is detailed and the number of tables with rules increases. This allows a finer classification of the studied phenomenon and the achievement of better results in the synthesis of a query to the database.

Tables with the models for correspondence of assignment signs – equality or equivalence, the signs for mathematical operations such as +, -, *, /, for logical operations – OR, AND, NOT are created. The aggregate functions max (), min (), count (), sum () are written in tables too, from which, once recognized in the text, their SQL construct will be taken. Characters often used for comparison <,>, <=,> =, ==, <>, predicates such as BETWEEN 'value' AND 'value', IN, NOT IN, LIKE, IS NULL, IS NOT NULL, as and rules for merging tables such as union (), intersect (), sort (), group by, order by are also written as rules (tags) in a parallel database of matching tables. Table 4 shows sample templates that present the clauses and expressions used in different ways and turn them into rules, which will form the SQL query string later.

Table 4. Correspondence table between language objects and SQL query constructs

ID	TokenBGName	TokenENName	SQL_tag	ID	TokenBGName	TokenENName	SQL_tag
1	и	and	and	**20**	по-голямо от	bigger than	>
2	както и	also	and	**21**	по-малко или равно	less than or equal to	<=
3	заедно със	together with	and	**22**	по-голямо или	greater than or	>=
4	в допълнение	in addition	and	**23**	различно от	different from	<>
5	или	or	or	**24**	еквивалентно	equivalent	=
6	не и	no and	not	**25**	равно	equal	=
7	не включва	does not include	not	**26**	съвпада със	coincide with	==
8	поне	at least	min	**27**	абсолютно равно	absolutely equal to	==
9	минимално	minimum	min	**28**	избери	select	select
10	най-малко	the smallest number of	min	**29**	извлчеи	extract	extract
11	най-много	at most	max	**30**	намери	find	find
12	максимум	maximum	max	**31**	покажи	show	show
13	най-голямо	the highest	max	**32**	сумирай	summarize	summarize
14	максималния брой	the maximum number	max	**33**	покажи списък	show list	list
15	най-голямото	the largest number of	max	**34**	открий	find out	find
16	групирани по	grouped by	group by	**35**	кои са	who are	select
17	обединени	united	group by	**36**	когато	when	when
18	сортирани по	sorted by	sort by	**37**	където	where	where
19	по-малко от	less than	<	**38**	с/със/ заедно	with / together with	with

Natural language processing automatically separates tokens, phrases, values and relations, reference to database elements (table names and their attributes, relationships between table rows, attribute values, their data types, etc.), as well as specific text constructs of concepts specific to "request for reference" texts. Based on formal features and correct differentiation of predicative definitions in the text, an analysis of the recognition and classification of language parameters and their transformation into an SQL query is performed.

3 Program implementation of the data extraction module

Text processing in the Bulgarian language is a dynamic and cyclical process in which the input text goes through successive steps of formalization and structuring. Each step corresponds to a certain level of perception and is realized within a separate function for analysis and processing. The dynamics of the process follows from the possibility of paraphrasing and correcting the query, while the cyclicity results from the successful execution of steps to achieve a certain result. The natural language comprehension methodology and the conversion of user text into an SQL query is an iterative process. The key activities are iterations, not individual stages in processing. With each new user request (iteration), all activities are performed, each performed at a different level of detail.

The algorithm for extracting information from a system with a natural language interface to the database includes these steps:

Step 1. Check in the input field for a valid text string – if there is one, take its value.

Step 2. Perform graphical analysis – the text is divided into sentences, using delimiters such as [\.! \?].

Step 3. The tokenization process divides the text of the question into separate tokens and gives a unique number – ID of each identified token.

Step 4. Check if each received token exists in the dictionary/grammar of the subject area.

Step 5. Remove from the question words that do not contain content – such as prepositions, conjunctions, interjections, pronouns, etc.

Step 6. Check if each received token is a table name or a field in the database.

Step 7. Syntactic and semantic analysis at different levels, where each token is characterized by grammatical characteristics belonging to certain categories, parts of speech, classes and checking for synonyms.

Step 8. Check for closeness between the received names of columns and the received names of tables from the database.

Step 9. Check for the presence of connection and display clauses, aggregate functions, conditional clauses and related mathematical operators, numbers, dates and values of attributes forming SQL constructs of database queries.

For the software implementation, an integrated environment for the development of software applications of the company Microsoft – Visual Studio 2019 and the modern, widespread high-level programming language, with numerous libraries and technological frameworks – C # has been chosen. When developing the software module, the best techniques and practices in the field of object-oriented programming are followed.

Programmatically, the relations between the objects in the database can be entered using the fragment, as a collection of the type Dictionary <TKey, TValue>, the elements of which are stored in Key-ValuePair objects, where the key is the name of the relation and value are the participants in it:

```
public Dictionary<List<string>, string> connections;
connections = new Dictionary<List<string>, string>()
   {
      { new List<string>(){"students", "teachers" }, "" },
      { new List<string>(){"disciplines", "students" }, "" },
      { new List<string>(){"disciplines", "teachers" }, "" },
      {new List<string>(){"disciplines","students", "teachers"}, ""},
    };
```

The key identifies the names of the relations. These names are unique and do not accept a null value. The value keeps the participants in the associative relations, which may be duplicated and/or missing. The values are accessed via connections [key].

The code demonstrates the type in which the elements of the collection are stored:

```
public struct KeyValPair
   {   public string key;
       public List<string> val;
       public string tableName;  }
```

Similarly, a limited set of predicate-argument structures of objects that can be the subject of inquiries has been created for the subject area.

The process of graphical analysis starts with the extraction of a list of consecutive language elements from the input string, which represent its significant elements such as tokens and SQL-constructs (join clauses, aggregate functions, conditional clauses and related mathematical operators), punctuation marks, numbers, dates and values of database attributes accompanied by their characteristics. The task of graphical analysis is to recognize the boundaries of individual sentences, words, phrases and other fragments of the input text request. Signals such as spaces, capital letters, punctuation marks denoting boundaries between sentences and constituent parts of sentences, paragraph indents and others serve as formal separators for defining the boundaries of text elements.

The implementation of segmentation and normalization in the application code is performed in the body of the Result class. It contains many methods and functions for linguistic analysis, declarations of lists for naming constants needed to determine the type of tokens received, e.g.:

```
public enum TokenEnum {TOKEN_TABLE_NAME, TOKEN_COLUMN_NAME, TOKEN_NONE, TOKEN_VAL };
```

Firstly when the user enters a text request in the Bulgarian language, its content is written in the variable txtQuestion. Then if the text is composed of several sentences, the splitIntoSentences method performs their segmentation based on punctuation marks at the end of a sentence. The method accepts the user question from the user natural language interface and returns a list of segmented sentences.

```
string txtQuestion = Session["txtInput"].ToString();
public static List<string> splitIntoSentences(string inputText)
{
    string[] sentences = Regex.Split(inputText, @"(?<=[\.!\?])\s+");
    return new List<string>(sentences);
}
```

The segmentation process continues at the sentence level. For each sentence, the splitIntoWordsAndNormalize method is applied. The method takes a sentence as an input parameter and returns a list of tokens (List <List <TokenItem>> tokens), which later will be normalized to their base forms.

```
List<List<string>> words = new List<List<string>>();
foreach (string item in sentences)
    {
        words.Add(splitIntoWordsAndNormalize(item));
    }
List<List<TokenItem>> tokens = new List<List<TokenItem>>();
```

The check starts based on the punctuation marks found in the text. In the fragment below, the text is divided into separate sentences. Each token is checked to see if it is a noun or a verb and normalized to its base form:

```
var punctuation = sentence.Where(Char.IsPunctuation).Distinct().ToArray();
var words = sentence.Split().Select(x => x.Trim(punctuation));
List<string> resultList = new List<string>();
      foreach (string item in words)
      {
        string word = item;
        word = word.ToLower();
        word = word.Trim();
```

```
  if (isVerbOrNoun(word))
  {
    if (word[word.Length - 1] == 's')
      word=word.Remove(word.Length - 1, 1);
  }
  resultList.Add(getNormalFormOf(word));
}
return resultList;
```

During the natural language analysis, morphological analysis is performed. It includes the process of normalization (lemmatization) of the obtained word forms, i.e. reduction of the different word forms to their unified representation (lemma) and their classification according to a set of morphological characteristics.

```
public static string getNormalFormOf(string word)
{
    dynamic obj = GetWordOBJ(word);
    if (IsPropertyExist(obj, "word"))
        { return obj.word; }
     return word;
}
```

The getNormalFormOf (word) method call a method that searches for words one by one in the linguistic database and returns a JSON object. The JSON object contains information about the speech rate of each token, a definition, its synonymous line, information about whether the token is a name of a table, a column in a table or connecting immutable word.

The getNormalFormOf calls the getWordOBJ method. Its definition includes checking whether the corresponding token is a characteristic (property) or value in the JSON object *obj*, obtained from the WordsAPI tool.

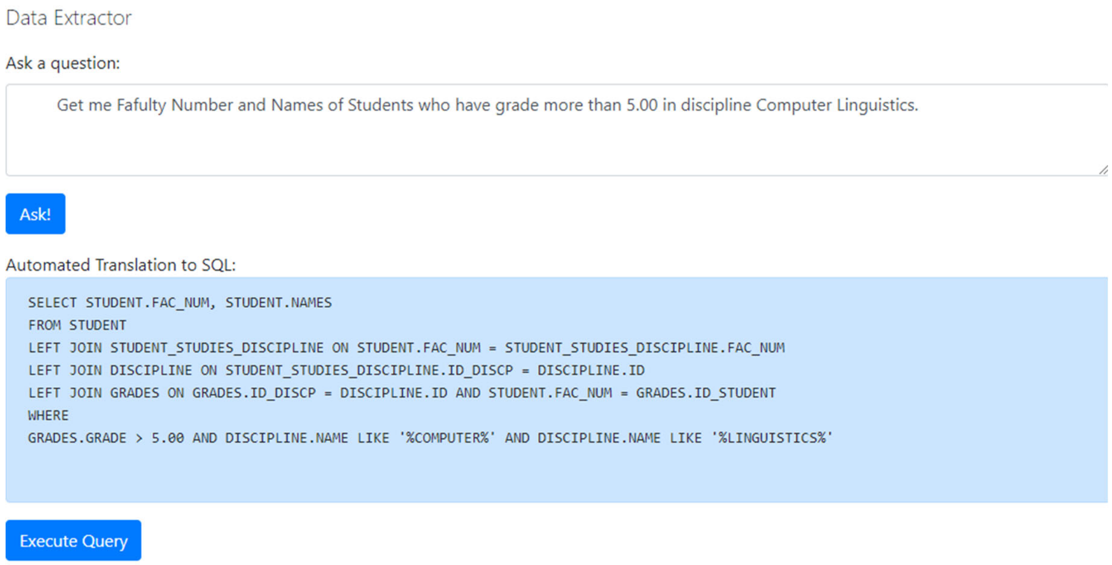
The **WordsAPI** tool is free and can be used with up to several thousand requests per day. It automatically provides information about the base form of the word, part of speech, synonymous line, pronunciation and accent. Finally, the resulting JSON object is deserialized to the variable *obj*. The type of the variable *obj* is dynamic. The object for working with structures *obj* contains generalized information about different data types without the need of converting and parsing between its constituent attributes. The reference to a dynamic object is performed based on the principle – lazy loading (design pattern, which creates an object during the program execution, immediately before use, no prior initialization is required in the application code).

Some common problems are observed in the natural language analysis, e.g. in equally phonetically represented but semantically different words, as well as in the numerical or abbreviated presentation of information. The model cannot make assumptions about such ambiguities. The study assumes that abbreviations and numerical representations are values (parameters) for the query, not arguments. Converting numbers to text representations is also not a subtask in the process of retrieving responses from a database. The last is also true for the issue with redundancies. A solution at this stage is to offer the user to reformulate its search.

The building on the software model continues with the definition of methods for automatic translation and generation of SQL queries from Bulgarian text. The model aims to detect semantic dependencies, work with synonymous rows of the used language units and find the exact fragments from which a formal SQL query is built. Specially designed matching templates are used, which distinguish the syntactic structure of SQL queries.

4 Experimental Setup

The linguistic data extraction module is tested with a sample University database structure, which was initially defined in the tool's knowledge base. In addition, the subject area's custom synonym sets (synsets) were defined. The tool also uses English language synsets, available in wordnet dictionary.



Get me Fafulty Number and Names of Students who have grade more than 5.00 in discipline Computer Linguistics.

```
SELECT STUDENT.FAC_NUM, STUDENT.NAMES
FROM STUDENT
LEFT JOIN STUDENT_STUDIES_DISCIPLINE ON STUDENT.FAC_NUM = STUDENT_STUDIES_DISCIPLINE.FAC_NUM
LEFT JOIN DISCIPLINE ON STUDENT_STUDIES_DISCIPLINE.ID_DISCP = DISCIPLINE.ID
LEFT JOIN GRADES ON GRADES.ID_DISCP = DISCIPLINE.ID AND STUDENT.FAC_NUM = GRADES.ID_STUDENT
WHERE
GRADES.GRADE > 5.00 AND DISCIPLINE.NAME LIKE '%COMPUTER%' AND DISCIPLINE.NAME LIKE '%LINGUISTICS%'
```

Figure 2. GUI of the Data Extractor Tool with example Natural Language Query and translated SQL

As shown in Figure 2, the tool first translates the text in natural language to SQL Select query upon clicking the "Ask!" button. The generated SQL Query can be observed and altered by the user. Then the user can click "Execute Query" button to get results from the Database.

In the current example, there are three keywords in the user input text, that refer to table names in the connected SQL database: students, grade, discipline. The table "Discipline", however, is not connected directly to the table "Student", so additional LEFT JOIN with the table STUDENT_STUDIES_DISCIPLINE is automatically added. The text value "Computer Linguistics" is broken into two like clauses: one for each word to maximize the correctness of the returned results.

5 Conclusion

The proposed model is used for automated extraction of values of quantitative indicators from a question-answer quality evaluation system to extract knowledge for the benefit of users. In this type of system, the user asks a question in natural language, and the system may return an answer, a list of answers or fail (if ambiguity has occurred or no data is found that matches the query). Data sources for the extraction of the quantitative data are a base with hierarchical indicators for quality evaluation and their quantitative dimensions and a linguistic tree (including a synonym dictionary) with words and phrases that build the indicators bodies grammarly. The paper presented a model for generating a SQL query from a user string in natural language. Now, the work on model analysis and experimental evaluation is in progress. The data retrieval module can be integrated into systems with a natural language interface and be used in other subject areas to support communication between users.

Acknowledgements

The paper is supported within the National Scientific Program "Young scientists and Post-doctoral students" in accordance with Appendix No. 11 of Council of Ministers Decision No. 577 of 17 August 2018.

References

[11] FINHEEC, Audit of the University of Graz, ISBN 978-952-206-236-9, 2013.

[12] I. Wisnubhadra, Service Oriented Business Intelligence for Monitoring Academic Quality, Proceedings of the 2nd International Conference on DEI, 2013, pp. 136-143.

[13] F. Diko, Z. Alzoabi, F. Alnoukari, Enhancing Education Quality Assurance Using information Systems-QAAS System, International Symposium on Information Technology, ITSIM 08, 2008.

[14] R. Doneva, S. Gaftandzhieva, Automated Data Retrieval in Distance Learning Quality, Proceedings of the 8th National Conference „Education and Research in the Information Society”, Plovdiv, 2015, pp. 83-93.

[15] S. Gaftandzhieva, Automated assessment of student activity in Moodle, Proceedings of the 8th National Conference „Education and Research in the Information Society”, Plovdiv, 2014, pp. 38-48.

[16] G. Totkov, S. Gaftandzhieva, R. Doneva, Dynamic Quality Evaluation in Higher Education (with Applications in e-Learning), First Varna Conference on e-learning and knowledge management, Varna, 2016, pp. 8-23.

[17] S. Gaftandzhieva, Model and System for Dynamic Quality Evaluation in Higher Education, PhD thesis, University of Plovdiv “Paisii Hilendarski”, 2017.

[18] S. Gaftandzhieva, R. Doneva, G. Totkov, Dynamic Quality Evaluation in Higher Education, TEM Journal, Volume 7 (2018) 526-542.

[19] S. Gaftandzhieva, R. Doneva, G. Totkov, Quality Evaluation In Higher Education: Dynamic Data Accumulation And Aggregation, International Journal of Scientific & Technology Research, Volume 9 (2020) 3275-3279.

[20] M. Zhekova, G. Totkov, Frame Model for Presentation of Semantic Rolls and Processes for the Establishment of Natural Language Interface, Proceedings of the National Conference on "Education and Research in the Information Society", Plovdiv 2019, ISSN 2534-8663, p. 42-52.

[21] P. Osenova, K. Semov, Formal grammar of the Bulgarian language, Institute for Parallel Information Processing, BAS, ISBN 978-954-92148-2-6, Sofia 2007.

[22] T. Boyadzhiev, I. Kutzarov, Y. Penchev, Contemporary Bulgarian Language, “Petar Beron” Publishing House, Sofia 1999, 654 pages.

[23] M. Zhekova, G. Totkov, Methodology for establishing a system of conformity patterns in question-answers system with natural-language interface, Proceedings of the National youth forums "Science, Technology, Innovation, Business” Plovdiv 2019, pp. 139-145.