Model of process and model of natural language processing system

M Zhekova¹ and G Totkov²

¹ PHD student, Department of Computer Science, University of Plovdiv „Paisii Hilendarski“, Tzar Asen str.24, Plovdiv, Bulgaria

² Prof., Department of Computer Science, University of Plovdiv „Paisii Hilendarski“, Tzar Asen str.24, Plovdiv, Bulgaria

m_jekova@uft-plovdiv.bg , totkov@uni-plovdiv.bg

Abstract: Natural language processing is the set of methods for making human language accessibility from computers. An approach that facilitates search in database even for casual non-technical literate users is to use a natural language interface (IEE) for databases. It is important for these IEEs to be intelligent and reliable enough to interpret the most complex sentences most intelligently and effectively enough to deliver results in the shortest possible time. The advantage of natural language interfaces is that they are accessible to a wide range of end users in order to "speak" to the database. Introduction. The solution to the problem of natural-language understanding can be accomplished in two steps: linguistic and database processing. In the first stage of language processing, the natural language query is linked and translated to the corresponding SQL query by appropriate production rules, methods and functions. In the second processing stage, the database is accessed and manipulated and the SQL query is executed by the system. The study presents a research in the field of computational linguistics on a theoretical process model and a natural language processing system model. The process model itself represents knowledge and reflects the conceptual organization of human memory, as well as its flexibility and clarity.

1. Introduction

Natural language processing is a branch of machine learning that focuses on enabling computers to interpret and process human languages in both speech and text forms. The effectiveness of machine learning applications is limited to having the right set of complete, correct, clear and accurate data, as relying on human knowledge base specialists to design, customize and refine models and the inability of these uniquely created models to generalize for each domain area, for all business processes, functions and even geographical regions.

The main purpose of the study is to propose and explore methods and models suitable for creating a system with natural language interface (NLI) for processing user queries in Bulgarian for a given domain area.

Achieving this goal involves developing an NLI for database that upgrades existing systems based on keywords, rules, semantic networks, frames and scripts, making them more automated, improving the quality of the rules. The intent is to create an easy and friendly environment for interacting with computers in natural language and for coping with the task of retrieving a knowledgebase response through a user question in natural Bulgarian.

One of the tasks of natural language processing is to build an abstract conceptual model of the existing database and SQL generating model, which together with the linguistic component and according to the specifications and principles of knowledge modeling, return the answer to the user's query. When creating a computer model of the process of natural language processing of Bulgarian text, it should be taken into account that for the objects and their relations there is already a database
(basic data or basic information base) and tools for their processing in order to be able to extract information about them and it does not necessarily depend on the specific methodology proposed. The existing relational-type information database is the connecting element in identifying and processing text-query in Bulgarian and the tools described below to generate a SQL query.

2. Related work

In most computer information systems, communication with the user is accomplished through a graphical user-friendly interface rather than in a natural way. Any database application interaction requires either a graphical user interface (GUI) or structured query language experience to a database [1]. However, structured query languages are difficult to understand for non-experts. GUI needs to be filled in forms, fields, buttons, menu selections, validation procedures, etc. Instead, NLIs allow the user to communicate directly with the database, in the form of commands that translate a question in their natural language into a database query.

The term 'natural language interface' (NLI) represents a kind of computer-human interface where language forms such as verbs, phrases and clauses act as parameters in the formation of database retrieval requests. Natural language software interfaces allow you to ask questions that are formulated in the user's native language. The NLI must be able to understand human language after typing and translate it into the language that databases understand in order to receive the information sought. Working with NLI is more intuitive than conventional GUI where the user works with menus, tables, and filter fields. Very often menus have inflexible options and sometimes do not meet the specific needs of the user.

In order for a software system to understand a user query, it is necessary to introduce the term "natural language processing". The processing involves converting a sequence of words into a semantic representation that can be used later in the processing. This includes the use of morphology, syntax and semantics. Morphology is the study of the structure and content of word forms. Processing ends with identifying the keywords and generating the meaning of querying the database.

Examples of resource grammars for the Bulgarian language are the syntactic corpus - BulTreeBank; The Chooser system for defining different annotation schemes, used to access detailed information of the proposed meanings through synonymous definitions and semantic relations; The BulNet semantic network of synonymous sets and their semantic relations; knowledge base on semantics of verbs Seminvest et al.

3. Model of a process for natural language processing of Bulgarian text

Figure 1 presents a general model of a dynamic process for natural-language processing of a user-asked-query set in Bulgarian. The development of the process in Figure 1 is presented with possibilities for changing the text of the query and the methodology for synthesizing a SQL query, as a result of the accumulation of new metadata for the received language units and the flow of different subprocesses for natural language analysis and synthesis, following different principles and methods in the domain area in question.

The schematically presented model of interaction between the basic elements of the natural language processing process underlies the design and creation of an automated natural language interface system to the database for any domain area. Designing, creating and using electronic lexical resources (grammars) is a necessary preparatory step in the natural language processing process indicated in Figure 1 with S_Intro. As a result of the digitization of linguistic resources, it can be argued that the linguistic units annotated therein are updated periodically.

The design of a methodology for synthesizing a SQL query to the database depends on the nature of the language units and the results obtained from the next step of the process indicated in Figure 1 with S_Transform. The methodology of the process of forming a pattern-question for generating a response from the system, which is the final stage in natural language processing, denoted by S_Synthese, takes the form of rules, instructions and directions.
Figure 1. A common model for the natural language request processing process

The correct execution of the process of processing the user request involves performing one of the following actions:

a) After normalization and segmentation of the received request in the first stage S_Transform, and after receiving a list of consecutive values of the specific database and concepts from the specific information base, the process goes to the next stage S_Synthese;

b) If the list of DB values does not correctly generate a question pattern (the corresponding values do not "match"), they do not have matching that will lead to a response from the system, the process returns to the beginning and the user must ask a clarifying question.

c) Upon receipt of the relevant elements of the request, a question-pattern is formed in S_QForm, suitable for the chat-bot question-answer system.

d) Following the question-pattern, a SQL query-pattern related to the database is received.

e) In S_GenAns a reference is generated, extracted from a question in Bulgarian.

This process can be formalized and developed as a theory that combines syntactic, semantic analysis and code generator into one. The article introduces a new approach to semantic analysis based on syntactic-semantic analysis and transformation rules. The algorithm annotates the set of concepts recognized by the input query with their semantic representations. A major innovation of this algorithm is its integration with state-of-the-art natural language comprehension techniques.

4. Model of a natural language processing system

One of the main tasks in the development is to create a model of an automated system for natural language processing in Bulgarian. For this purpose, it is necessary to propose and investigate relevant conceptual and computer models for the elements of Fig. 1. This means that the general model of Fig. 1 will be further developed by incorporating computer models of the linguistic corpus. It is also necessary to propose a software implementation model (computer model) for the S_Analyse modules (with S_Segm and S_Norm sub-modules) and S_Synthese (with the corresponding sub-modules).

The problem of responding to a user's request in natural language is solved using a set of different rules and criteria of the proposed methodology. In the end, the final response from the system is accompanied by a number of tools additional information resources that help to analyze and generate a SQL request to the DB, ie. Response = Request + Resource. It should be noted that preparing a response from the database requires some co-ordination with the other submodules, procedures and processes.
4.1. **Creation of a specific (for the given database) information base**

The information base (computer model) includes systems and tools that differ in the nature of the processes and processing they perform over their data; technology, maintenance, etc. The existing relational type database is preceded by an analysis of the information structure of the organization within which it operates. The analysis made on the existing information base is made in order to identify the accessible structures of objects, phenomena, processes, relations and tools for their automated processing.

The modeling of each relational database scheme and its relation to the natural language allows:
- interpreting naturally language queries and generating correct SQL requests.
- adding metadata to natural language concepts to enable natural language interpretation of user queries, thus increasing the language coverage of the proposed software.

![Diagram](image)

**Figure 2.** Identification of objects and relationships extracted from the existing database - digitization of language resources

Designing, modeling, and incorporating the elements derived from the analysis is an essential step in the development of the conceptual computer model. At this stage, the process of pairing the linguistic component and creating a computer model is not automated. In the future, however, the need and possibility of automating the matching process between an existing domain-specific database and the proposed compliance database model is discussed.

Terminology, factology and specialized knowledge presentation are important elements in the process of analysis and acquisition of knowledge. Terms such as the linguistic designation of specialized knowledge are inextricably linked to their presentation, storage, transmission, and extraction. Concepts are a core element of both the conceptual model and the knowledge base. Therefore, the concept is crucial to the presentation of knowledge. At any time from the perception of an event, the space around it is also perceived, [2] including objects, participants and objects present in it. Specialist linguists (experts) are involved in the process of collecting, annotating and creating the language component in S Intro stage.

Let's look at Figure 3. It shows some of the tables in dBUniversity - a relational database in higher education. It consists of objects - 'student', 'teacher', 'discipline', 'specialty', etc. and relationships between them, based on the phenomena and processes taking place in a higher education institution. In the reasoning and examples in the exposition we will be based on this database.
4.2. Creating a conceptual database model

Creating a conceptual model that describes the semantics of the restricted language of a particular database. The existing relational database is transformed into a conceptual network of nodes (concepts) and edges (relationships), which later model query texts from the specific database.

Through the abstract conceptual model model, we schematically represent the structure of the existing relational database. The model is generated automatically by retrieving table names, values, and attributes based on the database schema and is independent of domain area. Initially, the database schema is analyzed to retrieve table names, attributes, data types, and relationships between tables. Subsequently, the structure of all tables and finally the relationships are modeled.

Conceptual modeling is the activity of formalizing aspects of the physical and social world around us for the purpose of understanding and communication. A network of connections is created, mimicking the way the human mind works.

Developing a conceptual data model also includes information about current business processes or workflow analyzes within an organization. After creating a conceptual data model that users are happy with, the next step is translating it into a schema that implements the relevant data structures in the database. This process is often called logical database design, and output is a logical data model, expressed as a schema. Data in a database is usually managed by database management systems (DBMSs). Access to the database is facilitated through a special interaction language called SQL. To overcome the complexity of communication between lay developers and the database management language - SQL, many researchers use natural languages. The idea of using natural language has prompted the development of a new type of processing method called natural language processing to database systems.

In order to model the structure of a table in a database, classes, attributes and properties are defined that describe it. The properties, attributes, and classes used to describe the structure of a relational database are presented in the following table, with the name of each database table representing a TableN class and the columns of each database table representing a ColumnN class. Subsequently, ColumnN class representatives contact the corresponding TableN class representatives using the hasColumn property. The data type name describing each attribute is added to the corresponding attribute via the DataType property [3].
Table 1. Classes and properties for modeling the structure of each database table

| Name               | Property      | Description                                                                 |
|--------------------|---------------|-----------------------------------------------------------------------------|
| TableN             | Class         | A table name is introduced as a member of TableN                            |
| ColumnN            | Class         | The column name is introduced as a member of ColumnN                         |
| hasColumn          | Attribute Property | Relationship between a TableN and a ColumnN                               |
| isColumnOf         | Attribute Property | The inverse property of hasColumn                                            |
| relatesColToTable  | Relation      | Relationship between two tables                                             |
| DataType           | Datatype Property | Specifies the data type of each attribute                                   |

Figure 4. Scheme of the table Student and its attributes

After the structure of all tables is modeled as shown (see Table 1), the relationships between the tables in the database are followed. Figure 5 illustrates the relationship between two tables that is implemented through the relationsColumnToTable property. This property associates an attribute that is a representative of a ColumnN class with a table that is a representative of the TableN class. The TableN class representative must be a primary key in one table, and the ColumnN class representative must be a foreign key in another table (see PersonalData.ID and Student.PersonalDataID in Fig. 5).

The role of the representatives of the individual classes is to classify the objects according to their class. Proper modeling of table-to-table relationships is key to combining multiple table records when generating an SQL query.

Based on the proposed model, a hierarchy of classes and attributes, data types and relationships are dynamically defined for each table and attribute. This necessitates a system with a natural language interface to provide opportunities for:

- creation and management of a basic object (class, attribute property, relation, data type);
- defining roles and identifying objects in the database;
- defining procedures for checking different objects.

Modeling of information extracted from the database schema as a result of the classes and properties described (see Table 1) is performed dynamically and does not depend on the domain area. The user specifies only the database and the configuration is performed automatically. Following the conceptual scheme thus modeled, the user can further develop and customize with a set of natural language tools and metadata to describe each of the objects in the database.
Figure 5. Scheme of relations between two tables - Student and PersonalData, representatives of the class TableN

After modeling the conceptual scheme, the relationship between it and the natural language is determined. For this purpose, a set of natural language words is identified that describes all table and column names in the database that store information that could be requested by users.

The conceptual network is created to answer templates of questions that are centered around a problem that it seeks to solve. This will be accomplished by constructing data structures describing facts, concepts, and valence relationships from the database that match the names of the tables, attributes, and relationships between them, which we will call an abstract syntax tree or concept tree.

The syntax tree corresponding to the statement in Figure 6. will be represented as a tree with vertices, which are arguments describing the concepts Student and PersonalData, and edges, which are relations of syntactic subordination between words. All values in the table PersonalData (Name, EGN, Nationality, etc.) are inherited from the table Student because Student is a type of Person (Student inherits the PersonalData) and all attributes of the class PersonalData class are attributes of the Student class as well.
4.3. *Creating a Concept Tree for a domain area* – подмодул_S_BulNet

From the concepts in the conceptual network to the existing database, builds a tree of concepts described with its synonymous lines (reduced version of BulNet).

At this stage, potential usage information will be collected and concepts will be modeled in the context of the domain area. The resulting knowledge resource will not be static, irrelevant and represented by unrelated data records. The purpose of the_S_BulNet submodule is to present concepts as part of a more abstract concept or situation where the concept is related to other concepts in a dynamic structure that can be optimally and naturally optimized through the interface of the submodule. The resulting abstract syntax tree is a reduced version of the BulNet linguistic corpus.

The submodule_S_BulNet will semantically describe the concepts that make up the conceptual model of each database. Semantic modeling is here to increase the number of correctly answered user queries.

**Table 2. A set of attributes describing a Concept of a domain area**

| Attributes / Metadata | Concept |
|-----------------------|---------|
| Synsets               | Synsets |
| Description / Lexical meaning | Description / Lexical meaning |
| Part of speech        | Part of speech |
| Semantic class / category | Semantic class / category |
| Core                  | Core |
| Basic form / Lemma    | Basic form / Lemma |
| Hiperonym             | Hiperonym |
| Hiponym               | Hiponym |
| Принадлежи / Отнася се | Принадлежи / Отнася се |
| Question words        | Question words |

The feature tables that describe linguistically the concepts at this stage are based on the properties of the language units. The linguistic units are annotated with a syntactic category, part of speech, lexical meaning, synonyms and equivalent values, hyperonyms, hyponyms, etc. (see Table 2). Linguistic annotation data are presented in the form of key-value. Each category receives a single value derived from predefined sets.

A. Morphosyntactic characteristics are part of the morphological annotation. There are two main approaches to automatic morphological annotation.

B. Lexical characteristics describe the information lexical meaning of words, systemic lexical relations - synonyms, hyperonyms, antonyms, meronyms, etc.

C. Semantic characteristics attribute lexical meaning, semantic roles, semantic frames

Attributes of a class are inherited from all its heirs by the type-subspecies relationship, and the successors may have additional characteristics. The_S_BulNet submodule contains basic information about concepts in the domain area, such as semantic and syntactic features, but also contains compatibility (links) with other concepts in the concept tree.

The explicit specification and conceptualization, described by terms organized in taxonomies, their definitions and attributes, as well as related validation methods and value constraints, is intended to answer consumer questions. The conceptual description depends on both the domain and the tasks...
to be solved. The concepts correspond to the roles played by the objects in the domain area in the process of performing a particular activity.

4.4. Building a conceptual model for correspondences – подмодул S_FrameNet

From the edges of the conceptual network, a set of control models (reduced version of FrameNet) is obtained.

For a natural language processing system to be effective, it must have the necessary and sufficient knowledge to understand user questions and translate them into question patterns and SQL queries to the knowledge base. In order to interpret natural language user issues, an information system with a natural language interface must know the connection between the natural language (submodule S_BulNet) and knowledge of the database schema (submodule S_Transform).

The specialized knowledge resource - submodule S_BulNet, as part of the specific information base, aims to facilitate the analysis and acquisition of knowledge. Submodule S_FrameNet, which is the next step in the stage of creating a specific information system, provides descriptions of relationships and content in which the concepts are considered as part of a process or event. Because the acquisition and understanding of knowledge requires a specific situation, therefore horizontal relationships that determine the purpose, task, meaning and outcome of the work and use of an object are as important as vertical and specific relationships [2].

In part 4.2. we have specified that for the creation of a semantic network of concepts structured and grouped by different criteria, nodes are synonymous sets that encode equivalence relations between several linguistic units, and edges express semantic relations between objects located in the nodes. The nodes correspond to the concepts / objects in the knowledge base, and the edges indicate the connections between them. The combination between two nodes and an arc is called a statement or fact. The statement is the relation between different objects. In practice, the semantic tree describes the syntactic behavior of individual words, ie. what words can be attached to them so that the correct language constructions are obtained.

At this point in the S_FrameNet submodule, we build on the concept tree built in part 4.3. and from edges expressing relations we get a set of control models for concepts and verbs related to text queries in Bulgarian. The S_FrameNet submodule will serve to represent the minimum of structured information that defines the types of phenomena, events, situations or processes. We will step on the concept introduced by [4] - frame, as a means of presenting knowledge. Frame models are used to represent relatively independent, logically distinct units of knowledge that can be reused in a variety of situations (including data extraction and aggregation) [5].

Each frame has a unique name and structure consisting of ordered elements called slots [6]. In the frame, each slot has a unique name and is described with a wide range of different features, including dictionary interpretation, classification information about its syntactic and semantic functions, etc. Slots are described with one or more attributes that represent knowledge describing the element in question. Attributes, or so-called facets, represent an expanded knowledge of slot values and allow it to be controlled [7]. The value of a slot has a certain type of data. It can be elementary or composed. The contents of each slot can be text or numeric value, function, formula, relation to another slot, predicate dependency or other frame. The frame structure is dynamic, ie it allows new slots to be added at any time [8]. The purpose of classifying data into frame structures is to maintain and store data, to define relationships between them, and to facilitate access to them.

As elements of the frame system in the S_FrameNet submodule, different concepts are involved with their connections, described by facts and rules in the relevant domain. Structurally, it represents system-bound frame structures with different levels of abstractness.

The purpose of classifying data into frames is to maintain and store the data, to define the links between them, and to facilitate access to them. In this way, overall commitment, hierarchy, and subordination of actions contribute to the complete description of the logical model of the particular domain. Clarification of the basic principles for classifying units and the creation of uniform and consistent categories is a necessary condition for the applicability of classification (Stoyanova, 2012b). This structured knowledge base is reduced to a classification task, where individual frames contain answers to the most frequently asked questions for the domain under consideration.
4.5. **Forming a set of question patterns – подмодул S_InfoQA**

The proposed methodology for forming question patterns is illustrated with examples. Each example shows a question pattern and provides the necessary validations and sufficient conditions for the implementation of each specific pattern.

**Example 1:** Show / Display all semester fees.

![Figure 7. Schema showing the requested TableN instance](image)

**Pattern 1.** [Which are | Show | Display | Find] [all | every] X, where X is the only TableN class member and no attributes are searched.

The condition for applying this database query pattern to it is that the term is found to be a single TableN class representative. As a result, all records for that particular table are returned.

\[
X - \text{PaymentFee} - \text{a specific instance of the TableN class}
\]

Therefore, the pattern is undoubtedly applied because no specific attributes were found, and there is no other object in the TableN class. There is only one specific table and the result is all available records in it. The corresponding SQL query for this particular pattern is:

\[
\text{SELECT * FROM X}
\]

**Example 2:** Show the type and value of semester fees.

The example shows that in addition to the TableN class representative, there are already ColumnN class representatives, which are specific fields from a specific table.

![Figure 8. Schema showing the requested attributes of the TableN and ColumnN classes](image)

**Pattern 2.** [Show | Display | Find] A1, A2,..., An [and | ,] A2 [from | for | in | on] X, where X is the sole representative of the TableN class and A1, A2,..., An are the unknown attributes representing the ColumnN class.

The validations and checks that are required in this case are: Is there another TableN class member in the user string and all the concepts found are eligible to be columns on the only table. As a result, only the columns corresponding to the requested attributes from the table specified in the search string are returned.

\[
X - \text{PaymentFee} - \text{a specific instance of the TableN class}
\]

\[
A1 - \text{PayType}; A2 - \text{Value} - \text{are representatives of the ColumnN class and have the property: A = isColumnOf (X)}
\]

\[
\text{PayType} = \text{isColumnOf (PaymentFee)}; \text{Value} = \text{isColumnOf (PaymentFee)}
\]

Therefore, the pattern applies when, after the analysis and refinement of the properties of the attributes (concepts) involved in the query, if all attribute names are from the sole table, i.e. the isColumnOf (TableN) property, where TableN is a representative of the same class. Then the corresponding SQL query on the pattern is:

\[
\text{SELECT A1, A2,..., An FROM X}
\]
Example 3: Show the value of the fee for the second semester.
In the example, the search terms are identified as representing the TableN and ColumnN classes, but there is already an attribute value, i.e. a specific numeric (text) value for a column.

![Figure 9. Schema showing the specific attributes of the TableN and ColumnN classes that meet a certain condition](image)

Pattern 3. [Show | Select | Display | Find | Who is | Which are] A1,A2,...,An [from | for | in | on] X, [where| when] Ak="value", where X is the sole representative of the TableN class. A1, A2, A3,..., An are the requested attributes, representing the class ColumnN, and 'value' is a specific value for a particular attribute (Ak) in the same table.

For this template, validation methods include: is there another TableN class member in the user string; do all the concepts qualify to be columns of the table you are looking for. As a result, only columns that match the requested attributes from the specified table and that meet the specified condition are returned.

\[ X \rightarrow \text{PaymentFee} - \text{a specific instance of the TableN class} \]
\[ A1 \rightarrow \text{PayType}; A2 \rightarrow \text{Semester} - \text{are representatives of the ColumnN class for which:} \]
\[ A = \text{isColumnOf} (X) \]
\[ A1 \rightarrow \text{PayType} = \text{isColumnOf} (\text{PaymentFee}); A2 \rightarrow \text{Semester} = \text{isColumnOf} (\text{PaymentFee}) \]
\[ \text{Condition} \rightarrow \text{Semester} = "2" \]

Therefore, the pattern applies when, after applying the analysis, it is established that there is only one representative of the TableN class and one or more representatives of the ColumnN class, in addition to the requirement for one of the representatives of the ColumnN class.

Then the corresponding SQL query template is: SELECT A1, A2,..., An FROM X WHERE Ak = "value".

5. Conclusion
This paper presents a new study in the field of computational linguistics - the comprehension and interpretation of texts in natural Bulgarian language by computers. For the purposes of the study, a detailed examination of the results in the field of English language processing was made. Unfortunately, there are no other studies in this field for Bulgarian language so far. Developments for modeling structure, syntax, and semantics in other languages were explored. The subject of the analysis were systems based on pattern matching, keywords, frames, production rules, menus, and question-answer systems. A process model and a model of system with a natural language interface to a database for processing user queries of Bulgarian text is proposed, in the field of higher education.

From the results of the study (a process model and a model of system with a natural language interface to a database) and their general nature, we can conclude that ideas can be transferred to other domain areas. The results of the development may serve as a basis for further research into some promising areas, such as:
- improving the presented models, methods and strategies for accumulating metadata for language units;
- multiplying results for information infrastructures based on different technologies from other domain areas;
- creating tools and applications to describe other types of objects, e.g. work processes and phenomena from activities in other fields of knowledge.
References

[1] Mony M., Rao J., Potey M., An Overview of NLIDB Approaches and Implementation for Airline Reservation System, International Journal of Computer Applications (0975 – 8887) Volume 107 – No 5, December 2014

[2] Faber P., San Martin A., Conceptual modeling in specialized knowledge resources, International Journal “Information Technologies and Knowledge”, Vol. 4, Number 2, 2010

[3] González J., Florencia R., Fraire H., Pazos R., Cruz-Reyes L., Gómez C., Semantic representations for knowledge modelling of a Natural Language Interface to Databases using ontologies, International J. of Combinatorial Optim. Problems and Informatics vol. 6 no. 2 Jiutepec may./ago. 2015

[4] Minsky M, A Framework for Representing Knowledge, MIT, Cambridge, 1974
Ruppenhofer J, Ellsworth M, Petruck M R L, Johnson C R, Scheffszyk J, FrameNet II: extended theory and practice, Technical Report, 2005

[5] Gaftandzhieva S, Doneva R and Totkov G, Frame Models in Programming Training, XI National Conference "Education and Research in the Information Society", 2018

[6] Zhekova M., Totkov G., A Frame Model for Representing Semantic Roles and Processes for Building a Natural Language Interface, ARIO 2019 – Plovdiv, May 30-31, 2019
Fellbaum C, WordNet - An Electronic Lexical Database, MIT Press, 1998

[7] Gaftandzhieva S, Doneva R and Totkov G, Frame representations in e-learning applications and new developments, International Journal on Information Technologies & Security, No 2 (vol. 10), 2018
Zhekova M. Review and Study of Existing Artificial Intelligence Systems, Proceedings of the 21st National Scientific Conference for Students and PhD students "The World is a Open Book", Plovdiv, May 16-17, 2019

[8] Zhekova M., Totkov G., M., Conceptual frame model for the presentation of the concepts and rules in natural language interface for database, 8th International scientific conference “TechSys 2019”, Technical university of Sofia, Plovdiv branch 16-18 may 2019
Stoyanova Iv, Automatic recognition and tagging of composite lexical units in the Bulgarian language, 2012 Sofia