Vol-2546
urn:nbn:de:0074-2546-4

Copyright © 2019 for the individual papers by the papers' authors. Copyright © 2019 for the volume as a collection by its editors. This volume and its papers are published under the Creative Commons License Attribution 4.0 International (CC BY 4.0).

CS&SE@SW 2019
2nd Student Workshop on Computer Science & Software Engineering

Proceedings of the 2nd Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2019)

Kryvyi Rih, Ukraine, November 29, 2019.

Edited by

Arnold E. Kiv *
Serhiy O. Semerikov **
Vladimir N. Soloviev **
Andrii M. Striuk ***

* Ben-Gurion University of the Negev, Beer Sheba, Israel
** Kryvyi Rih State Pedagogical University, Kryvyi Rih, Ukraine
*** Kryvyi Rih National University, Kryvyi Rih, Ukraine

Table of Contents

Invited Talks

Computer Science

Software Engineering

2020-01-24: submitted by Serhiy O. Semerikov, metadata incl. bibliographic data published under Creative Commons CC0
2020-02-09: published on CEUR-WS.org |valid HTML5|

Some aspects of designing of the structural semantics visualization system

Yaroslav Vasylenko[0000-0002-2954-9692], Galina Shmyger[0000-0003-1578-0700]
and Dmytro Verbovetskyi[0000-0002-4716-9968]

Ternopil Volodymyr Hnatiuk National Pedagogical University,
2, Maxyma Kryvonosa Str., Ternopil, 46027, Ukraine
{yava, shmyger, verbovetskyj_dv}@fizmat.tnpu.edu.ua

Abstract. The *purpose* of this article is to investigate the principles and technologies of creating such a semantic interconnection system that would be useful and practical for use in areas such as machine translation, search engines and contextual search. According to the purpose of the research the main *tasks* are defined: 1) to study and analyze the basic principles of construction of semantic dictionary of English language WordNet; 2) to create a lexical-semantic web dictionary of IT-terms of the Ukrainian language. The *novelty* of the work is to adapt all the principles of WordNet to the Ukrainian language. The *practical meaning* of the results obtained is to create a semantic dictionary of the Ukrainian language that will allow to better analyze Ukrainian texts by searching not only the words themselves, but also words that are in one way or another related to the primary, and that will significantly increase the speed of search and analysis of information. In the created web-application (thesaurus) the basic functions of similar existing systems and the latest methods of information linguistics are implemented.

Keywords: structural semantics, lexical-semantic web-dictionary, structural semantics sort systems, thesaurus.

1 Introduction

Providing automation of the efficient work with data presented in the form of natural language texts is one of the actual tasks of computational linguistics. It is caused both by an increase in the e-information stream, and by the need for critical analysis of texts for the subject of authenticity, similarity, probability, etc. A correct understanding of a language is possible provided there is a knowledge of how words and concepts are related to one another, which is meant by one or another utterance, what the purpose has speaker saying a one or another phrase; what is said and what needs to be found in context or perceived based on previously learned information. To solve the problems of analyzing the relationship between words and concepts, to identify all the features of a language, so-called lexical and lexical-semantic databases were developed. Such systems include Princeton WordNet [7; 8; 9; 10], MindNet (Microsoft Research Project

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2546/paper17.pdf

software), FrameNet, VerbNet, HowNet, ConceptNet and more. However, for Ukrainian-language content, such developments are at an early stage [5].

The **purpose** of this article is to describe the structural and logical scheme of building a web application for lexico-semantic analysis of user query in the subject area "Informatics".

The applied meaning of structural semantics closely intersects with the problem of natural language analysis, which is that structural semantics serves as the key to defining the lexical contextual meaning of words, which is the main task of natural language analysis [1]. At this stage of human development, we have not yet fully learned by machine means to interpret natural language and to create a system that at human level is able to perceive natural language and interpret the results, to continue the dialogue.

That is why structural semantics is today a topical direction in the development of both philological and informational disciplines. The structural semantics is at the intersection of two different approaches to exploring the world, it absorbs the best of both, creating a scientific symbiosis that is the building block of the science of the future.

2 Analysis of the basic concepts of the study

2.1 Natural language thesaurus

Thesaurus is a complex component of the dictionary type, in which all the meanings of the dictionary are interconnected by semantic relations, reflecting the basic relations of concepts in the described subject area of knowledge [6]. In the past, thesaurus was mostly referred to by dictionaries, which with the utmost completeness represented the vocabulary of the language with examples of its use in texts.

The thesaurus consists of tokens relating to four parts of the language: adjective, noun, verb and adverb. The descriptions corresponding to each part of the language have a different structure.

The main relationships in thesauruses are:

Synonymy – a link between words in one language, different in sound and spelling, but having the same or very similar lexical meaning, for example, daring – brave.

Antonymy – the relation between the words of one part of the language, different in sound, has the exact opposite meaning: true – false, good – evil.

Hyperonym – a word with a broad meaning that expresses a general, generic concept, the name of a class (set) of objects (properties, features).

Hyponym – a word with a narrower meaning that names an object (property, feature) as an element of a class (set). These relationships are transitive and nonsymmetrical. A hyponym inherits all the properties of hyperonyms. It is a central relation for the description of nouns.

Meronymia / Partonymy – the relation "part – entire". Within this relationship stand out the relationship of "being an element".

In addition to these relationships, they also introduce thematic relationships that connect the concepts of one subject area.

An example of a thesaurus:

The hut is a wooden peasant’s house.

[Hyperonym]: a residential building

[Meronym]: a rural settlement

[Synonym]: a house

All relationships create a complex hierarchical network of concepts. The properties of relations in the description of different parts of the language are different. In different systems, a thesaurus can perform different functions:

- a source of specialized knowledge in a narrow or wide subject area, a way to describe and ordering the terminology of the subject area;
- search engine in information retrieval systems;
- manual document indexing tool in information retrieval systems (so-called control dictionary);
- automatic text indexing tool.

The main documents governing the thesaurus format are ISO 2788-1986 standards for describing monolingual thesauruses, and ISO 5964-1985 for multilingual ones.

ISO 2788-1986 defines a thesaurus as a set of terms that relate to each other.

The American standard ANSI / NISO Z39.19-1993 extends and refines the ISO 2788-1986 standard for monolingual thesauruses, and imposes a number of additional restrictions on the thesaurus structure.

Thesauruses remain to current date the most accepted form of description of subject domain knowledge, suitable for human perception. Examples of modern foreign thesauruses are WordNet and EuroWordNet.

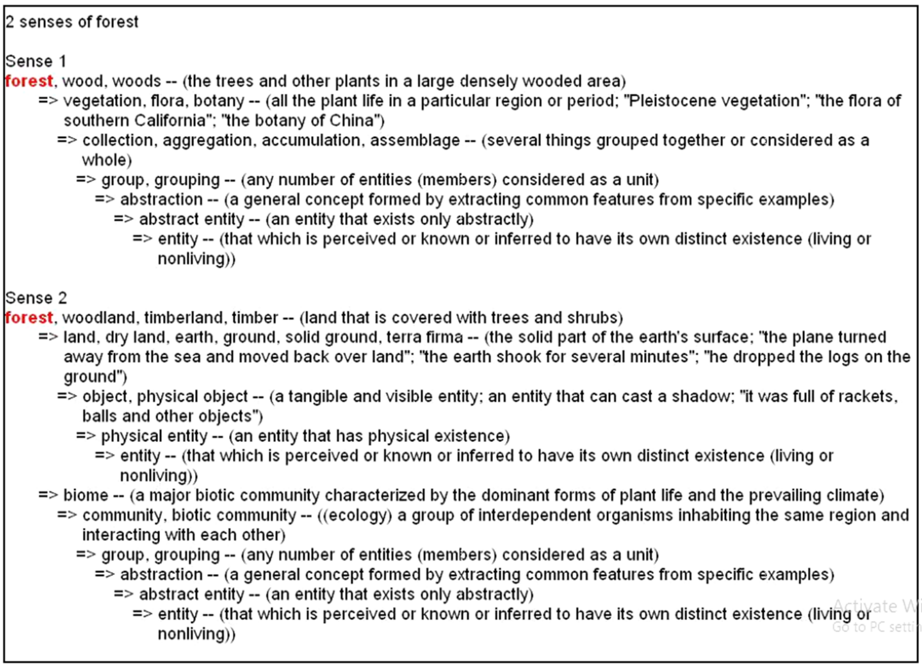
The WordNet English thesaurus emerged in 1990 and began to actively attracted in various areas of automatic word processing. WordNet covers about 100,000 different units (nearly half of which are phrases) organized in 70,000 concepts.

The development of the thesaurus was started in 1984 at Princeton University of the United States under the leadership of the famous psycholinguist George A. Miller [7; 8]. In 1995, WordNet appeared on the Internet freely and caused a surge of research on its use in various computer applications of automatic word processing. The results of using WordNet in automatic word processing turned out to be not unambiguously positive, but WordNet ushered in a new era of developing extra-large structured linguistic resources and caused the emergence of a large number of followers in different countries who create such “natives” for their languages [5]. This thesaurus has also become the basis for multifaceted discussions and research, on the basis of which principles should be built large linguistic resources, suitable for various applications in computational linguistics [4].

The main relation in WordNet is the attitude of synonymy. Synonym sets – synsets – are the basic structural elements of WordNet.

The concept of synonymy used by WordNet developers is based on the criterion that two expressions are synonymous, if replacing one of them with another in the sentence does not change the meaning of the truth of the expression.

The relations between the synsets form a hierarchical structure (Fig. 1). When constructing hierarchical systems on the basis of genitive relations, it is usually assumed that the properties of the parent concepts are inherited by the child – the so-called property of inheritance. Thus, nouns are displayed as a hierarchical system with inheritance. In this case, a systematic effort should be made to find for each synset its generic concept, its hyperonym.



```
2 senses of forest

Sense 1
forest, wood, woods -- (the trees and other plants in a large densely wooded area)
    => vegetation, flora, botany -- (all the plant life in a particular region or period; "Pleistocene vegetation"; "the flora of
       southern California"; "the botany of China")
       => collection, aggregation, accumulation, assemblage -- (several things grouped together or considered as a
          whole)
          => group, grouping -- (any number of entities (members) considered as a unit)
             => abstraction -- (a general concept formed by extracting common features from specific examples)
                => abstract entity -- (an entity that exists only abstractly)
                   => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or
                      nonliving))

Sense 2
forest, woodland, timberland, timber -- (land that is covered with trees and shrubs)
    => land, dry land, earth, ground, solid ground, terra firma -- (the solid part of the earth's surface; "the plane turned
       away from the sea and moved back over land"; "the earth shook for several minutes"; "he dropped the logs on the
       ground")
       => object, physical object -- (a tangible and visible entity; an entity that can cast a shadow; "it was full of rackets,
          balls and other objects")
          => physical entity -- (an entity that has physical existence)
             => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or
                nonliving))
    => biome -- (a major biotic community characterized by the dominant forms of plant life and the prevailing climate)
       => community, biotic community -- ((ecology) a group of interdependent organisms inhabiting the same region and
          interacting with each other)
          => group, grouping -- (any number of entities (members) considered as a unit)
             => abstraction -- (a general concept formed by extracting common features from specific examples)
                => abstract entity -- (an entity that exists only abstractly)
                   => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or
                      nonliving))
```

Fig. 1. Hyperonyms for two values of the forest noun:
forest as a collection of trees and forest as an area where trees grow.

EuroWordNet multilingual thesaurus is currently being developed. Initially, in four languages (Danish, Italian, Spanish, and American English), a network of word meanings is developed that is linked to semantic relationships and allows you to find words that are similar in meaning to different languages. Unlike WordNet, which was designed to describe the lexical and conceptual system of the English language, EuroWordNet is primarily designed to solve the practical tasks of automatically processing large text arrays. The most important tasks that are supposed to be solved with this thesaurus are the following:

- providing multilingual information retrieval;
- increasing the completeness of information search;
- request formulation in natural language;
- semantic indexing of documents, etc.

Domestic scientific institutions have created more than a hundred industry thesauruses that satisfy a certain state standard for dictionaries of this type. They are called – IRT – information retrieval thesauruses.

Standard IRTs are intended primarily for manual indexing of documents, as well as for the formulation and variation of search queries. There are non-standard thesauruses that make the task rather of selective systematizing of terminology in a particular field of knowledge more relevant – this is especially relevant for new subject areas.

2.2 Thesaurus interfaces in information systems

In an information system, a thesaurus is not only an independent information resource, but also a tool for classifying or indexing resources. Thus, the user of the information system should be able to:

- view the thesaurus;
- search for resources by associated terms or concepts (resource search can be accomplished in two ways: keyword search or using a thesaurus);
- do navigation on thesaurus, that is, searching for the desired concept first in the thesaurus, and then querying resources corresponding to this concept.

When searching for keyword resources, the search engine can, by using a thesaurus, extend the search results by giving the user not only the resources that match the keywords entered, but also the resources of related terms or terms, which also denote narrower terms for the original term.

Thesaurus view interface must:

- to show all attributes of a given term or concept;
- to show what terms and concepts are associated with that term or concept;
- to show for the user visually the place of the term or concept in the thesaurus concept hierarchy.

The first 2 points will be fulfilled if show for the user for each thesaurus concept on a separate screen (page) all its attributes, all related terms (in all or in a specific language), and all related concepts. The interface must, at the same time, provide a transition to the viewing page of any of the concepts listed on this page. If the thesaurus data schema allows the term to be bound to more than one concept, then on the same page for each term the terms to which the term is still bound must be listed. If the concept has terms in other languages, fully equivalent to, but attached in the structure of the thesaurus to other concepts, links to pages of those concepts should be provided on the page.

If the thesaurus has a strictly tree-like structure, then the tree is usually presented in the following ways:

- visualization the path of the tree from the root to the current element;
- visualization the path of the tree from the root to the current element, as well as the neighbors of each ancestor of the current element;
- visualization of the whole tree completely. Usually in such cases, the user can open and close the reflection on screen the descendants of any nodes.

To provide an efficient sampling (by one request) of the necessary incisions of hierarchical structures that are fed recursive links between the nodes of these structures, the database tables are expanded with auxiliary columns and integrity conditions.

3 The Presentation of Main Results

3.1 Technological tools for implementing the structural semantics sorting system

Frontend. The Bootstrap framework was chosen as the creation tool of the frontend part of the project, which in the modern world in one way or another serves as the basis for most Internet projects. Bootstrap is the most popular HTML, CSS and JS framework for designing the look and interactivity of web pages.

Designed for anyone and any device, Bootstrap helps you make web pages look faster and easier. It is suitable for people of all levels of experience, for devices of all sizes, and for projects of any size.

Bootstrap comes with pure CSS, but its core code uses the two most popular CSS preprocessors Less and Sass. You can quickly get started with CSS ready, or prefer to building of the styles from core.

Bootstrap was chosen for this project because of the ease, speed and capability of more extensive and easier customization than CMS systems. Also, the choice was made with the expectation that the code of the system, written according to Bootstrap standards can be easily transferred to any other system without significant difficulties, which gives undoubted advantages in the perspective of the project development.

Backend. Since the software part of the project is its core, it was decided not to implement it in the languages of web programming (PHP, JS), but in the full-fledged OOP language C#, using ASP.NET technology [2; 3].

ASP.NET is a technology for creating web applications and web services from Microsoft. It is part of the Microsoft.NET platform.

Since the thesaurus project is essentially a large-scale work on the database, editing it, adding new values and relationships between them, considerable attention was paid to the choice of the database management system (DBMS). The choice was made on MySQL.

Today, MySQL is one of the most well-known, reliable and fastest of the whole existing DBMS family. The principle of operation of MySQL is similar to the principle of operation of any DBMS that uses SQL (Structured Query Language) as a command language to create / delete databases, tables, to replenish tables by data, to perform data sampling.

MySQL, like any other DBMS, is a server program that resides in the computer's memory and maintains a TCP port. The client connects to the DBMS from this port and sends the SQL queries. In turn, the server interprets them by performing the necessary actions and sends the results of the request back to the client. This is how the database server communicates with the client programs.

Because the project is implemented on C# and on ASP.NET technology, choosing a programming environment was not a problem. Because C# is a programming language created by Microsoft and is a product of its own, it was decided to opt for another Microsoft product – Visual Studio (which is perhaps the only full-fledged C# development tool).

3.2 Basic structural elements of the program

Web application Word Topology (WT) consists of such structural elements as database (dictionary, synsets, relationship between synsets), server part (backend), web interface (Frontend).

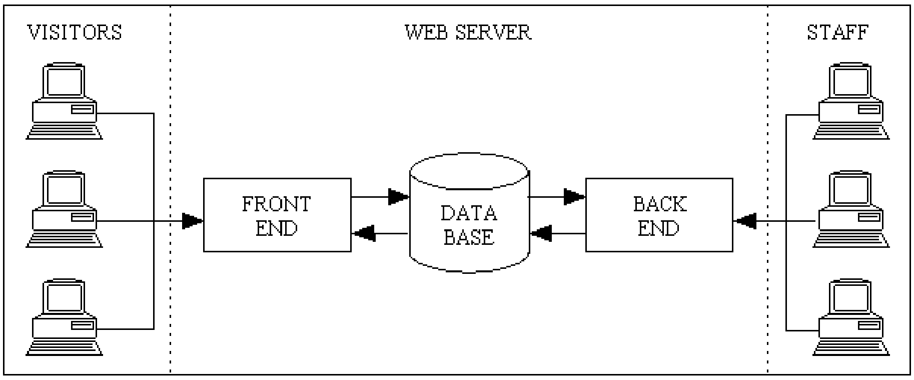


Fig. 2. Scheme of web application work with database, server part and user interface

Database is the place where all the data used by the web application are stored and systemized. As a thesaurus is, in essence, a giant database, as much as possible attention was paid to the DB architecture. Of course, WT works with much smaller amounts of information than, for example, WordNet or other common thesauruses, but the simplicity and ergonomics of the database architecture play an important role even in such projects. Thanks to a well-designed database, it is possible to reduce the server's response time and make the web-application not only a training platform, but also a completely practical system that can be used by users from any corner of the globe.

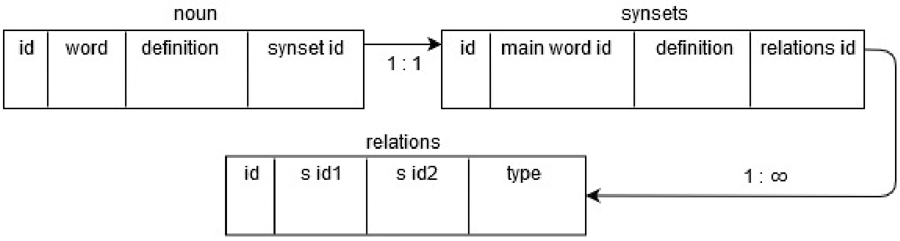
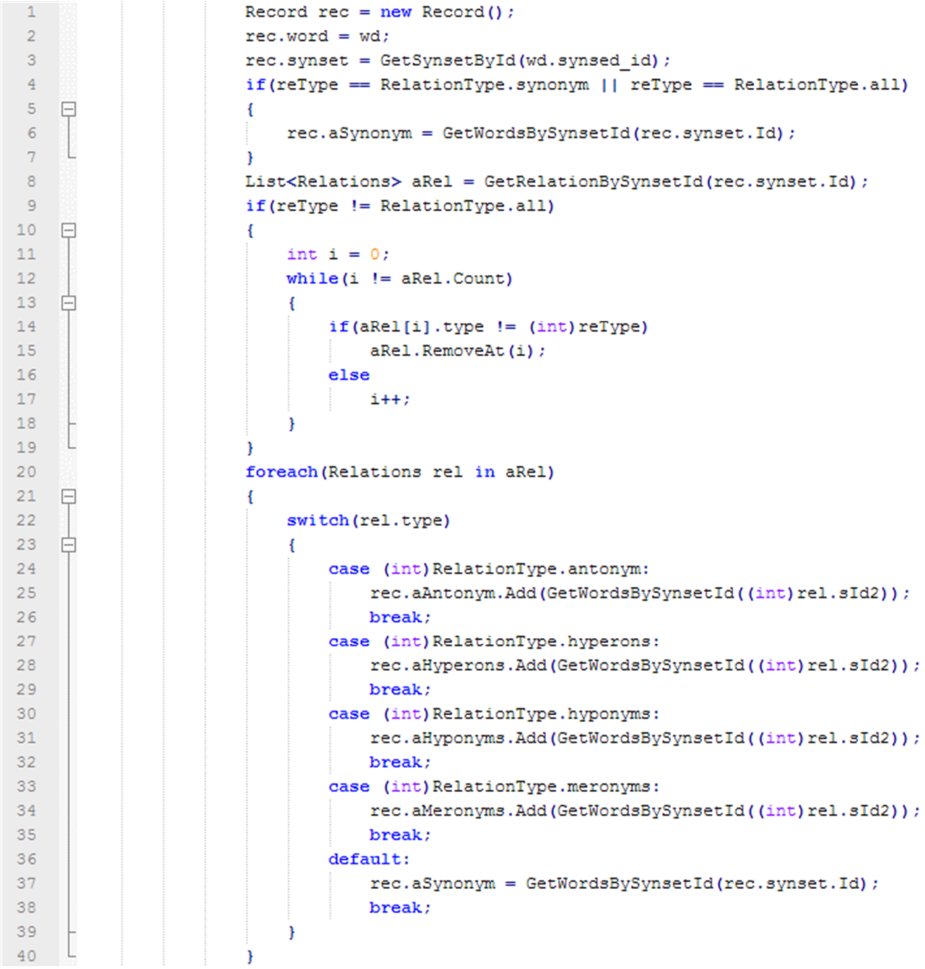


Fig. 3. Organization of the database for the WT web application

Backend is the part of a web application that is responsible for encapsulated user actions and information processing processes. In the WT project, backend is a set of frontend interaction functions, database access, query result formatting, and return of those results back to the request source. Functionality of the backend part is implemented in C# programming language. Here is an example of code from the server side (Fig. 4).

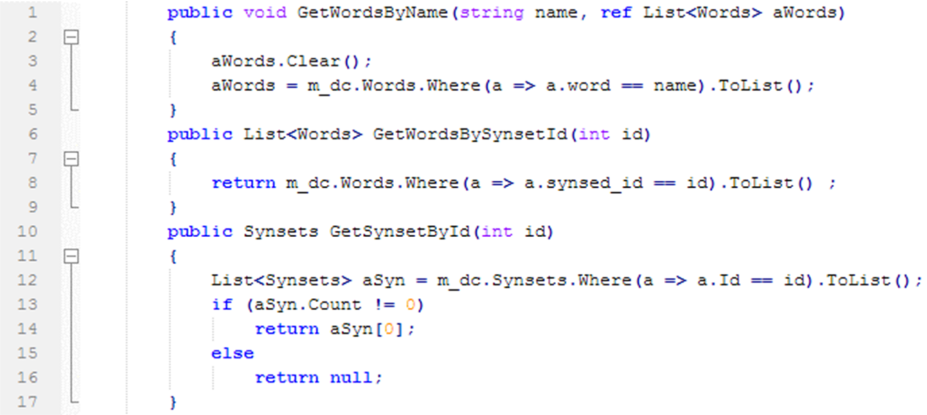


```
Record rec = new Record();
rec.word = wd;
rec.synset = GetSynsetById(wd.synsed_id);
if(reType == RelationType.synonym || reType == RelationType.all)
{
    rec.aSynonym = GetWordsBySynsetId(rec.synset.Id);
}
List<Relations> aRel = GetRelationBySynsetId(rec.synset.Id);
if(reType != RelationType.all)
{
    int i = 0;
    while(i != aRel.Count)
    {
        if(aRel[i].type != (int)reType)
            aRel.RemoveAt(i);
        else
            i++;
    }
}
foreach(Relations rel in aRel)
{
    switch(rel.type)
    {
        case (int)RelationType.antonym:
            rec.aAntonym.Add(GetWordsBySynsetId((int)rel.sId2));
            break;
        case (int)RelationType.hyperons:
            rec.aHyperons.Add(GetWordsBySynsetId((int)rel.sId2));
            break;
        case (int)RelationType.hyponyms:
            rec.aHyponyms.Add(GetWordsBySynsetId((int)rel.sId2));
            break;
        case (int)RelationType.meronyms:
            rec.aMeronyms.Add(GetWordsBySynsetId((int)rel.sId2));
            break;
        default:
            rec.aSynonym = GetWordsBySynsetId(rec.synset.Id);
            break;
    }
}
```

Fig. 4. Creation and populating an instance of the Record class – an intermediate link between the backend and the frontend

Creating an instance of the Record class is the output product of the backend system. The instance attributes store all the information about the result of the database query. From this fragment it is easy to see that such information contains: the word sought, its definition, synsets in which the word resides, the relation of the synsets data to others (hyponymy, hyperonymia, meronymia, antonymy, etc.).

Also noteworthy is the implementation of search methods for records in the database by the entered name and the search for words on the basis of the synset ID (Fig. 5). Implementing two approaches for word search is a necessary step, since the main task of creating a WT thesaurus was to create such a graph-oriented interaction system (which is a thesaurus + relation) so that the user can move freely between the nodes of the graph without any artificial restrictions and with maximum convenience.



```
public void GetWordsByName(string name, ref List<Words> aWords)
{
    aWords.Clear();
    aWords = m_dc.Words.Where(a => a.word == name).ToList();
}
public List<Words> GetWordsBySynsetId(int id)
{
    return m_dc.Words.Where(a => a.synsed_id == id).ToList() ;
}
public Synsets GetSynsetById(int id)
{
    List<Synsets> aSyn = m_dc.Synsets.Where(a => a.Id == id).ToList();
    if (aSyn.Count != 0)
        return aSyn[0];
    else
        return null;
}
```

Fig. 5. The code snippet that is responsible for 2 different methods of finding the desired word in the database

The frontend is written in HTML hypertext markup language and CSS cascading style sheets. The whole system is designed with bootstrap framework and ASP.NET technology, which allows to connect functionally frontend and backend.

The whole interface of the program is implemented (for ease of use) by a structure called accordion. The essence of structure is the submission of information in the form of collapsing lists. The implementation of this element in the software part of the web application is shown in Fig 6.

3.3 Functionality of the program

The main purpose of the program is to create a natural language thesaurus that, taking into account the mistakes of previous similar developments, could serve as a more efficient and accessible capacitive system of human vocabulary, which can be easily used in such areas as automatic translation of texts, systems of parser scanning of documents, systems contextual autocomplete / contextual search in search engines.

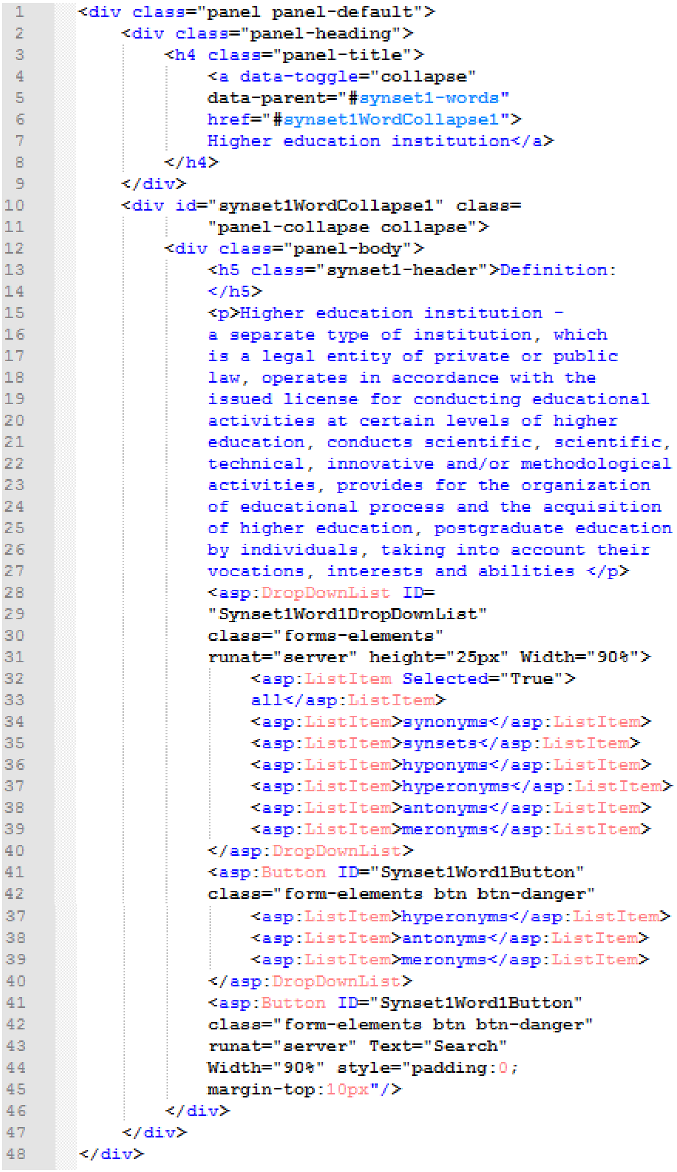
The main purpose of WT has defined the entire functionality of the application.

At this stage of development, the project does not contain all the planned functions, their development requires a deeper analysis of the context of the topic and increased knowledge in parallel with the increase of the project development team.

At this stage the following functions are implemented:

- search word from database;

- search for a word from a database relative to a word synset;
- a convenient way to move between words within the synset and in close (on graph) territories;
- search for all relationships of the synset on all levels of the hierarchy;
- output full information regarding the synset.



```
<div class="panel panel-default">
    <div class="panel-heading">
        <h4 class="panel-title">
            <a data-toggle="collapse"
            data-parent="#synset1-words"
            href="#synset1WordCollapse1">
            Higher education institution</a>
        </h4>
    </div>
    <div id="synset1WordCollapse1" class=
            "panel-collapse collapse">
        <div class="panel-body">
            <h5 class="synset1-header">Definition:
            </h5>
            <p>Higher education institution -
            a separate type of institution, which
            is a legal entity of private or public
            law, operates in accordance with the
            issued license for conducting educational
            activities at certain levels of higher
            education, conducts scientific, scientific,
            technical, innovative and/or methodological
            activities, provides for the organization
            of educational process and the acquisition
            of higher education, postgraduate education
            by individuals, taking into account their
            vocations, interests and abilities </p>
            <asp:DropDownList ID=
            "Synset1Word1DropDownList"
            class="forms-elements"
            runat="server" height="25px" Width="90%">
                <asp:ListItem Selected="True">
                all</asp:ListItem>
                <asp:ListItem>synonyms</asp:ListItem>
                <asp:ListItem>synsets</asp:ListItem>
                <asp:ListItem>hyponyms</asp:ListItem>
                <asp:ListItem>hyperonyms</asp:ListItem>
                <asp:ListItem>antonyms</asp:ListItem>
                <asp:ListItem>meronyms</asp:ListItem>
            </asp:DropDownList>
            <asp:Button ID="Synset1Word1Button"
            class="form-elements btn btn-danger"
                <asp:ListItem>hyperonyms</asp:ListItem>
                <asp:ListItem>antonyms</asp:ListItem>
                <asp:ListItem>meronyms</asp:ListItem>
            </asp:DropDownList>
            <asp:Button ID="Synset1Word1Button"
            class="form-elements btn btn-danger"
            runat="server" Text="Search"
            Width="90%" style="padding:0;
            margin-top:10px"/>
        </div>
    </div>
</div>
```

Fig. 6. The code snippet responsible for displaying the word in synset in accordion form

3.4 Program interface

Much effort has been put into the interface of the program, as in the modern Internet space, users pay attention not only to the functionality and usefulness of the resources used, but also to their appearance. The UI / UX rules of clarity, simplicity, convenience and aesthetic appearance were taken into account when designing the web application interface.

Structurally, the web application interface is divided into two parts - a greeting page and a thesaurus page, which in turn consists of a navigation menu of search, a output field for information about synsets and an output area for information about interconnections between synsets.

3.5 Development of a business layer of structural semantics sorting system

The business layer of this system is implemented using the classes and interfaces listed in Table 1.

Table 1. Classes and interfaces implemented in the system

Name	Attributes	Fields	Methods
class Default	–	o _ws o form o relationDropDown o synsetHolder o txtWord	o Buton1Click() o DisplaySynset() o Page_Load() o RenderControlToHtml()
class Main_dbEntities	o Relations o RelationType o Words	–	o Main_dbEntities() o OnModelCreation()
class WorkSpace	-	o m_ds	o WorkSpace() o AddWord() o DeleteWord() o GetAllWords() o GetSynset() o Init() o SearchWords() o UpdateWord()
class Synset	o ID o Words	o _id o _words	o Synset()
interface Word	o description o Id o name o synset_id	–	–
interface RelationType	o description o Id o name	–	–
interface Relation	o Id o relationType_id o word1_id o word2_id	–	–

Figure 7 presents the classes and interfaces of the developed system.

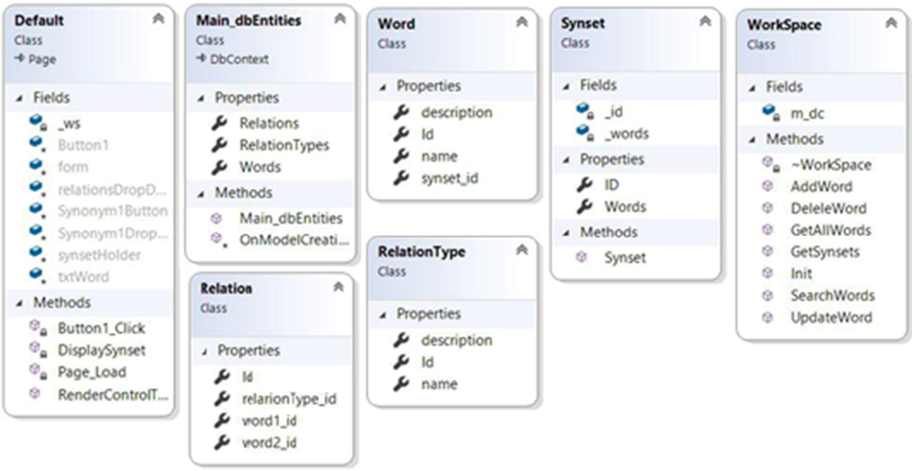


Fig. 7. System classes and interfaces

The search is performed by the word the meaning of which you want to output and by the type of connection that combines the words.

There are 6 types of connections, namely:

- USE
- Used For
- Broader Term
- Broader Term Generic
- Broader Term Partitive
- Related Term

To implement them, a RelationType class was created in the application code.

The program also encounters an All link, which means only that you need to search for words across all links.

The server processes the request and returns a list of elements of the WordSearchResult class. The objects of this class will be created for each meaning of the word searched and will include:

- the word itself
- a synset that includes the specific meaning of the word
- a words list of all types of connections that include the search word, along with a words list for each such connection.

The main logic for working with the database is in the WorkSpace class:

```
public class WorkSpace
{
   public void Init();
   public List<Word> GetAllWords()
   public List<Words> GetWordsBySynsetId(int id);
   public Synsets GetSynsetById(int id);
   public List<Synset> GetSynsets(string sWord)
   public WordSearchRezult SearchWords(string sWord, int? nRelationTypeID = 0)
   public List<string> ParseRelations(string relations);
}
```

4 Conclusions

In this study, most representatives of large thesauruses were analyzed, their source code and algorithms were investigated. Based on the collected data, we created a system of structural-semantic interrelations of words of the Ukrainian language. During the development of the WT web application we took into account the negative aspects of most similar systems and created a combination of the most successful solutions in this field.

During the completion of practical part of the task, an optimal database architecture of dictionaries and other structural units was created, most of the most common thesaurus functions were written, and a user-friendly and intuitive UI was designed that allows to use of thesaurus functionality by ordinary users, not just specialists.

The practical meaning of the developed vocabulary is to improve the search quality in Ukrainian texts. This is directly related to the fact that the search will be conducted not only by a specific word, but also by synonyms, or words that are in one way or another related to the original one.

Also, the initial function of thesauruses of this type cannot be underestimated - finding information in thematic dictionaries is many times more effective than simply browsing the Internet, due to the output of an extremely large number of thematically related information.

The scientific meaning of the dictionary, like most thesaurus dictionaries, provides for the possibility of comparing various aspects of natural languages with one another.

In the the study, the goals and tasks were fulfilled, namely:

- the main principles of WordNet construction and the main types of connections between the synsets were analyzed;
- methods of synsets construction were implemented and optimal database was developed on their basis;
- C# language features were used, namely LINQ (Language Integrated Query) queries, to work with the database efficiently;
- WordTopology web application was developed.

References

1. Anisimov, A., Marchenko, O., Nykonenko A.: Alhorytmichna model asotsiatyvno-semantychnoho kontekstnoho analizu pryrodnomovnykh tekstiv (Algorithmic model of associative-semantic context analysis of natural language texts). Problemy Prohramuvannia, 2-3, 379–384 (2008)
2. Chadwick, J., Snyder, T., Panda, H.: Programming ASP.NET MVC 4: Developing Real-World Web Applications With ASP.NET MVC. O'Reilly Media, Sebastopol (2012)
3. Freeman, A.: Pro ASP.NET MVC 5. Apress, New York (2013). doi:10.1007/978-1-4302-6530-6
4. Kedrova, H., Potemkin, S.: Struktura tezaurusa WordNet i semanticheskaya metrika na lingvisticheskoy baze dannyih (WordNet Thesaurus Structure and Linguistic Metric on Semantic Database). Interface.ru. http://www.interface.ru/home.asp?artId=36137 (2008). Accessed 6 September 2019
5. Kulchytskyi, I., Romaniuk, A., Khariv, Kh.: Rozroblennia WORDNET-podibnoho slovnyka ukrainskoi movy (Development of a WORDNET-like dictionary of the Ukrainian language). Visnyk Natsionalnoho universytetu "Lvivska politekhnika" 673, 306–318. http://ena.lp.edu.ua:8080/bitstream/ntb/6774/1/34.pdf (2010). Accessed 6 September 2019
6. Lukashevych, N.: Tezaurusyi v zadachah informatsionnogo poiska (Thesauri in information retrieval problems). MSU, Moscow. https://istina.msu.ru/publications/book/1283494/ (2010). Accessed 6 September 2019
7. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, G., Miller, K.: Introduction to WordNet: An On-line Lexical Database. http://wordnetcode.princeton.edu/5papers.pdf (1993). Accessed 6 September 2019
8. Miller, G.A., Hristea, F.: WordNet Nouns: Classes and Instances. Computational linguistics **32**(1), 1–3 (2006). doi;10.1162/coli.2006.32.1.1
9. Miller, G.A.: Nouns in WordNet. In: Fellbaum, C. (ed.) WordNet – An Electronic Lexical Database, pp.23-47. The MIT Press, Cambridge (1998)
10. WordNet | A Lexical Database for English official site. The Trustees of Princeton University. https://wordnet.princeton.edu (2019). Accessed 6 September 2019