UNIVERSITY OF WOLVERHAMPTON
SCHOOL OF HUMANITIES, LANGUAGES AND SOCIAL SCIENCE

DEVELOPMENT OF ENGLISH-BULGARIAN COMPREHENSION ASSISTANT

BY

TODOR ARNAUDOV

PRESENTED IN PARTIAL FULFILMENT OF THE FOLLOWING AWARD
BA (HONS): EXCHANGE

STUDENT NO.: 0624851

PROJECT CODE: LN3000

CREDITS: 15

DATE: MAY 2007

SUPERVISOR: R. MITKOV

This work or any part of it has not been previously submitted in any form to the university or to any other institutional body for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

Acknowledgements

I would like to thank all the people from the Research Group in Computational Linguistics for their suggestions, support and friendliness through the time of the development of my course project!

I would also like to thank my colleagues Jitka and Sonia for discussing with me ideas about the features that an advanced computer dictionary should have.

PREFACE

This paper discusses the architecture and the features of the English-Bulgarian Comprehension Assistant “Smarty”.

The project aims to test interface tools and advanced dictionary look-up techniques for part-of-speech tagging, multi-word expressions recognition and word sense disambiguation, which narrow the set of suggested translations and make comprehension of text in English faster and easier for non-expert readers.

Since there is no evidence for the existence of equivalent or comparable developments of comprehension assistant tools for English to Bulgarian or for any other language to Bulgarian, “Smarty” also aims to be the most sophisticated Bulgarian application for bilingual dictionary look-up.

The paper is divided in five chapters. The first one introduces the term "comprehension assitant" and discusses the need of intelligent dictionaries, supported by natural language processing technology. The second chapter presents previous developments and achievements in the field. The third one is devoted to the system we have developed and suggests a number of future improvements. The fourth chapter evaluates the results. Chapter five concludes the paper.

Table of Contents

Table of Contents

Table of Contents

1. Introduction to Comprehension Assistants

The most widely used software tools, created to help comprehension, still are conventional computer dictionaries, which ironically work the simplest possible way.

This chapter emphasizes the inadequacy of this situation to the state-of-the-art of Natural Language Processing technologies and gives suggestions about how a NLP-supported dictionary should look like.

1.1. Conventional Computer Dictionaries

Globalization and explosion of the Internet, almost exclusively in English at the time, made computer "To English" and "From English" dictionaries a must-have utility in late 90-ies and early 2000s in many non-English-speaking countries, including Bulgaria. Although the quantity and variety of information, available in other languages grew immensely, yet English is, and is supposed to be the ultimate way to spread messages around the world to as many people as possible.

Since acceptable open-domain machine translation systems do not exist and what is available can not satisfy the need of proper and accurate translation, the masters of mass quasi-translation market became a lot of off-line and on-line multilingual computer dictionaries.

However, while actually being much more useful and handy than the classic paper dictionaries, almost all available computer dictionaries provide roughly the same limited set of functionality and suffer from the same simplicity. The user just enters a word and receives an article from scanned or typed paper dictionary.

Few programming enhancements - predictive typing and automatic translation of a word, copied by the user to the clipboard of the operating system - quickly became a sort of standard, but yet these merely programming tricks do not save the tedious job of browsing for the appropriate sense, when the queried word has many of them.

1.2. Conventional English-Bulgarian Computer Dictionaries

The most popular English-Bulgarian dictionaries - **SADictionary** and **AEnglish** - differ very little in their interfaces and share almost the same functionality and lexical database.

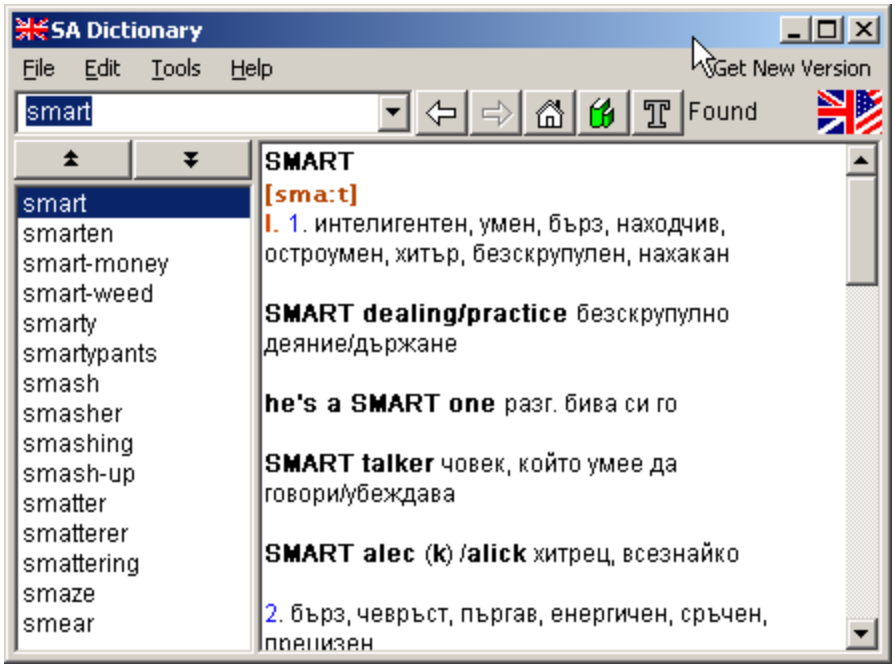


Fig.1 SA Dictionary

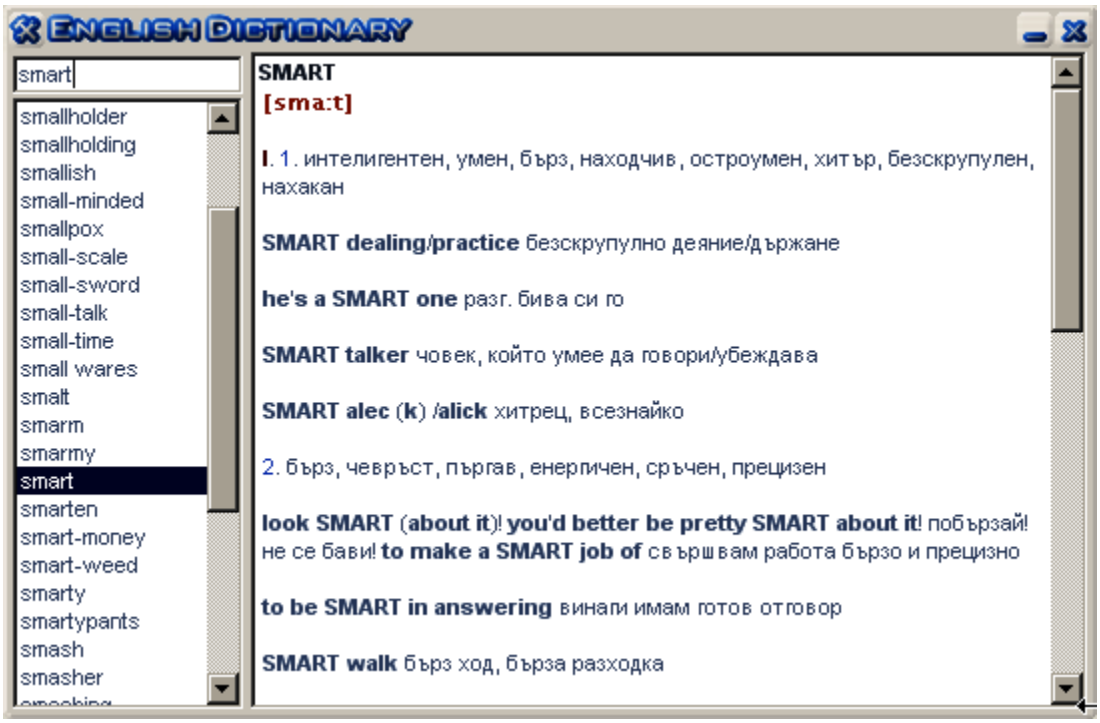


Fig.2 AEnglish Dictionary

Even though they both have examples for multi-word expressions (MWE) in their dictionary entries and they do display them neatly in bold font, these dictionaries do not search for matches, so if more than one word is entered, in the best case the user will get an article for the word which begins like the first word from the input.

Using SA Dicitionary, the only way for recognition of MWE is scrolling the article and search for the expression by a naked-eye, assisted solely with bold font and capitalized keywords. Alphabetical order is unavailable.

AEnglish has "Find" dialog-box for the entries, but it would capture only exact matches.

Obviously, even basic NLP-processing is missing.

1.3. Limits of Conventional Dictionaries

While it has to be admitted, that conventional bilingual computer dictionaries work fine in a lot of occasions - dealing with unambiguous words with single or very few senses, studying new words - the usefulness of this simple technology falls down and down with the grow of the list of senses.

Many English words are extremely ambiguous, and even they do at two levels: they could play different roles as part-of-speech – this is **PoS-ambiguilty** - and in each role they play, they may have a lot of sub-meanings – this is **lexical polysemy,** – or they may have one or several **homographs –** words which share the same string of letters, but have completely different meanings.

For example, **WordReference.com** suggests about **85 translations** of the word "run" in French - as a verb and as a noun.

If a dictionary with the same database has the information that "run" is, e.g. a **noun** and not a verb, that would reduce the list of translations to about **25** (the numbers are similar for English to Bulgarian, referring to SA Dictionary). Although still not a precise suggestion, the progress is very good, and it is important to note that this is not a difficult problem to solve, since today's part-of-speech tagging is a robust technology with very high precision, reaching over 95-97% for some languages.[7, Ch. 11]

If the word is used with a preposition or next to particular collocating word, the dictionary may try to recognize corresponding compound form or a multi-word expression. In case of success, this step could reduce the senses to **one or very few.**

Going further, current NLP technology allows Word Sense Disambiguation (WSD), based on analysis of the context of the given word, for solving lexical polysemy. State-of-the art precision of WSD is up to 75% [7, Ch. 13].

1.4. Comprehension Assistants

"**Comprehension assitant**" and "**Intelligent dictionary**" are synonyms of:

Advanced multilingual computer dictionary, which is not merely an electronic copy of a paper dictionary, but is enhanced with variety of natural language processing techniques for narrowing down the set of suggested translations.

While standard dictionaries are queried with **single words**, unrelatied to any others, comprehension assistants process ***words in context***.

While standard dictionaries always display all the information, linked to a given **word,** comprehension assistants aim to give the ***most relevant information for particular usage of a lemma.***

2. Previous Work

The usefulness of comprehension assitants is undoubted and proved by systems initiated more than 10 years by Xerox LocoLex.

Unfortunately this type of tools still lacks popularity among regular users, due to its high-cost or unavailability.

2.1. LocoLex

Rank Xerox has developed the first comprehension assistant at PARC, XERC between March 1994 and March 1996 within the framework of the COMPASS European project. *"LocoLex uses part-of-speech tagging and idiom recognition to provide users with the translation relevant to the word in its particular context, faciliating and accelerating the process of comprehension"*. [8]

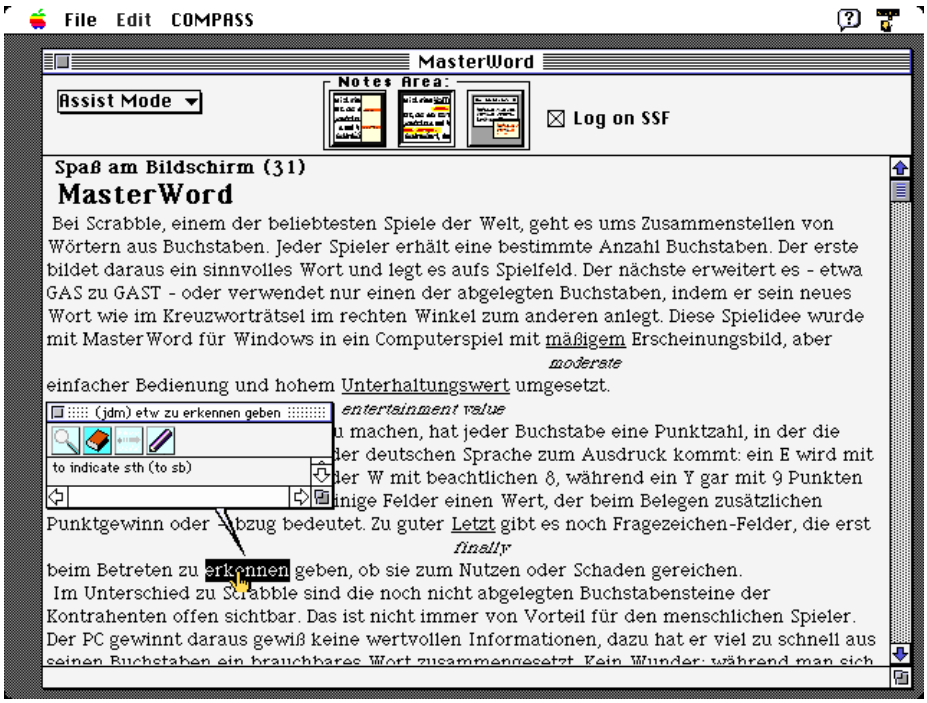


Fig 3. LocoLex

2.1.1. LocoLex Features

- User points a word from a text in a window of LocoLex.
- The context around the word is splitted into words.
- Each word from the sentence is normalized to a standard form for unified dictionary look-up.
- Words are morphologically analysed and possible parts-of-speech cases are generated.
- The most likely syntactic usage of the pointed word is disambiguated, basing on surrounding words.
- The relevant entries are found, including possible homographs and compound forms.
- Eliminate irrelevant sections, using morphological analysis and disambiguation phases.
- Identify special or idiomatic usage.
- Display to the user only the most appropriate translation, based on the part of speech and surrounding context. [7, Ch. 38]

These features became the minimum standard for an application to be accepted as “comprehension assistant”.

2.1.2. Evaluation

H. Feldweg and E. Breidt reported "overwhelmingly positive" results even at the first test phase.[8] A second test phase, with an improved version of the prototype of Locolex had been supposed to be conducted in the beginning of 1996 [8]. However, the final paper and evaluation of the first Locolex was not found.

2.1.3. LocoLex with Word Sense Disambiguation

A rule-driven semantic word sense disambiguation was developed as an extension of Locolex in late 90-ies, initially only monolingual - for English, - even though the system extracts its disambiguation rules from the same Oxford-Hachette English-French and French-English dictionary, used in LocoLex. [8]

Bilingual Word sense disambiguation extension of LocoLex was created, as well - unsupervised rule-based semantic tagger, which works on all input words. Semantic disambiguation rules are directly extracted from dictionary examples and their sense numberings. [7, Ch. 38].

2.2. Morphologic MobiDic, MetaMorpho and MobiCat

2.2.1. Goals

These applications emphasize the complexity of nowadays graphical user iterface applications, which are loaded with too many windows and menus. They aim to provide minimalistic user-friendly interface for their comprehension assistants, allowing their systems to be easily and transparently integrated in third-party applications.[12]

2.2.2. MobiDic and MobiMouse

Morphologic's ideas about interface are illustrated in MobiDic, MobiMouse and MobiMouse Plus series. Their software is capable to detect and translate any word on the screen, including menu commands, and the result is displayed in the corner of the screen - user can always look if he or she feels uncertain, yet without being distracted if he or she does not need comprehension assistance at a given moment.

2.2.3. MobiCat with MetaMorpho

MetaMorpho is a powerful sentence-level English-Hungarian machine translation system, based on Pattern Based Machine Translation - a hybrid method, combining probabilistic Example Based Machine Translation and a deterministic Rule Based Machine Translation. [12]

MobiCat is a commercially available application of MetaMorpho.

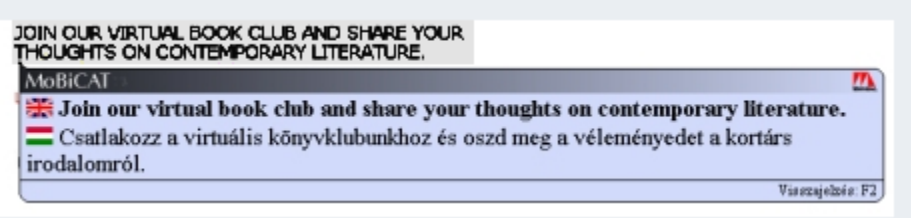


Fig 4. MobiCat

3. Smarty - English-Bulgarian Comprehension Assistant

The architecture of "Smarty" and its design are discussed in this chapter, as well as possible improvements, left for future work.

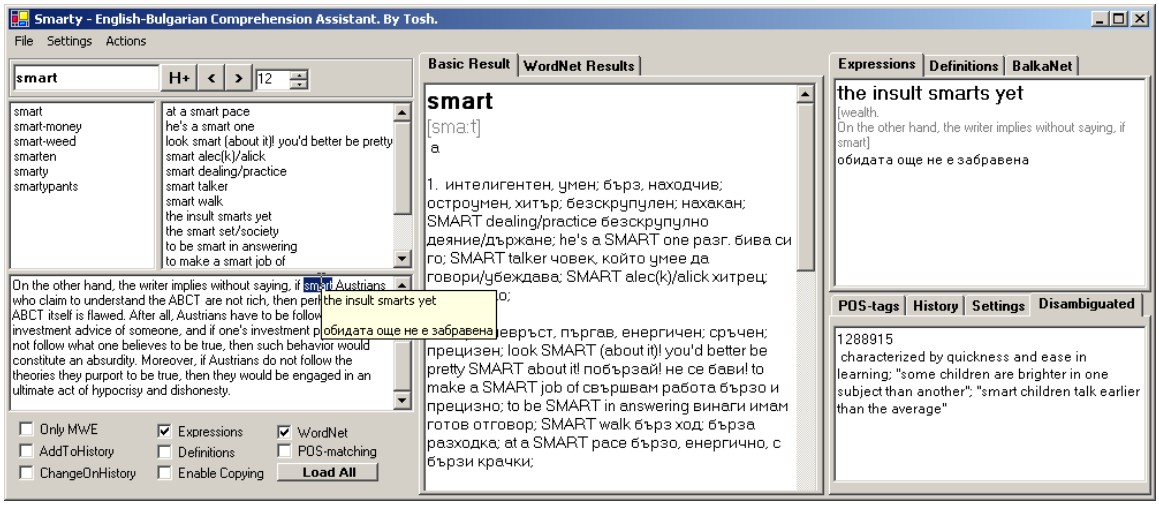


Fig 5. Smarty in action

3.1. Linguistic Resources

"Smarty" is based on three linguistic resources:

1. English-Bulgarian Standard Dictionary.
2. Princeton University WordNet 2.0.
3. BalkaNet – WordNet for Bulgarian.

3.1.1. English-Bulgarian Standard Dictionary

Initially this was a text-file, named "en.txt", distributed freely on the Internet as a base for development of conventional dictionaries. It consists of about 52000 dictionary entries, each of them containing a word; a transcription; and an entry with part of speech information, list of senses and, for some of the words, examples of multi-word expressions or usage.

Main drawback of the database is the "condensation" of information in it, which complicates the parser for expanding dictionary entries to a convenient form.

For example, the bound between an example of multi-word expression or usage and its translation is only the character set used – latin versus cyrillic. Probably, due to optical scanning, there are mistaken letters, which look the same in cyrillic and latin – e.g. **e, n, o**. This must be corrected by heuristics; otherwise the expressions are incorrectly extracted. In other cases, some Cyrillic letters "З" are placed instead of number "3".

There are English words in brackets, inserted between the words in the definition in Bulgarian, which is another complication.

Also, entries for words with many senses, like “run” or “go”, have references from one sense to another sense of the same word, by expressions like “run for 1)= run1 17.;”

3.1.2. WordNet 2.0

Princeton University WordNet is a large lexical database, consisting of synonym sets of words - “synsets” – structured by part-of-speech and numerous types of semantic relations. The richness of its structural information makes it highly accepted as a tool for various NLP and Compuational Linguistics tasks, such like word sense disambiguation, machine translation, automatic generation of multiple-choice tests and many other fields. [7, 11]

Although current version is 3.0, in “Smarty” we use a version from 2003, due to its alignment with the third important resource for our research – BalkaNet (see below). WordNet 2.0 consists of over 115000 synsets, and the total count of all unique lexemes of noun, verbs, adjectives and adverbs is 144309 with 203145 word-sense pairs in total, which yet makes it a very rich resource. [10]

Our system relies on WordNet in three directions:

1. Glosses with various senses of words in English are used to back-up the basic dictionary in case the word does not have an entry there. The gloss is not a translation, but may help dictionary user to grasp the meaning, because it is supposed that he or she has some knowledge of English.

2. Glosses in Bulgarian from BalkaNet are presented when applicable, by mapping to WordNet.

3. Word-Sense Disambiguation - for narrowing the offered list of senses to one-single - is performed, using the glosses in WordNet.

3.1.2.1. Structure of WordNet 2.0

The core of the database is organized in four index and four data files for **nouns**; **adjectives** and **adjective satellites**; **adverbs** and **verbs**.

Index files list the synsets to which a given lexeme belongs.

Data files contain the **lemmas – individual words or collocations –** of the synsets; lists of semantic and lexical relations between different synsets - hyperonymy, hyponymy, synonymy, antonymy, holonimy, meronimy etc. – and short gloss and/or examples of usage for the lemmas.

The lemmas of the synsets and their corresponding glosses are our resource for performing word sense disambiguation (see below).

3.1.3. BalkaNet

"The Balkan WordNet aims at the development of a multilingual lexical database comprising of individual WordNets for the Balkan languages. The most ambitious feature of the BalkaNet is its attempt to represent semantic relations between words in each Balkan language and link them together in order to develop an on line multilingual semantic network." [2]

BalkaNet was developed for Bulgarian, Czech, Greek, Romanian, Serbian, Turkish and was alligned to WordNet 2.0, i.e. the synsets of any pair of these lexical semantic databases could be linked directly.

	Bulgarian	Czech	Greek	Romanian	Turkish	Serbian
Synsets	21441	28456	18461	19839	14626	8059
Nouns	14174	21009	14426	13345	11059	5919
Verbs	4169	5155	3402	4808	2725	1803
Adjectives	3088	2128	617	852	802	324
Adverbs	9	164	16	834	40	13
Literals	44956	43918	24366	33690	20310	13295
Literal/Synset	2.1	1.54	1.33	1.7	1.39	1.65
BC1s	1218	1218	1218	1218	1220	1219
BC2s	3471	3471	3462	3471	3479	3469
BC3s	3827	3827	3825	3827	3794	1369
Domain Specific Synsets	2065	304	238	286	300	305
Balkan Specific Synsets	220	257	309	151	103	117
Language Specific Synsets	116	257	52	545	204	206
Language Internal Relations	28599	25683	24368	25885	19834	12787

Table 1. Statistics for the final version of BalkaNet. [2]

"Smarty" uses smaller version of Bulgarian part of BalkaNet, consisting of about 15000 synsets.

3.2. Architecture of Smarty

The System can be divided in 11 major components:

1. Interface.
2. Wrapper of SharpNLP PoS-tagging library.
3. Additional Tokenizer and Token-With-Context Extractor.
4. Basic bilingual dictionary database.
5. Predictive Typing Processor.
6. Standard Dictionary Look-up.
7. Dictionary Entries Analyzer.
8. Normalizer (Lemmatizer and Forms Generator).
9. Multi-Word Expression Matcher.
10. WordNet 2.0 and BalkaNet Wrapper and Mapper.
11. Word Sense Disambiguator.

3.2.1. Interface

Although the limited time that we had for development forced us to emphasize much more on natural language processing aspects than on the graphical user interface (GUI) and some of the windows were left in unpolished stage, we were obliged to spend some time for research and development of GUI.

3.2.1.1. Main Input and Output in Standard Mode

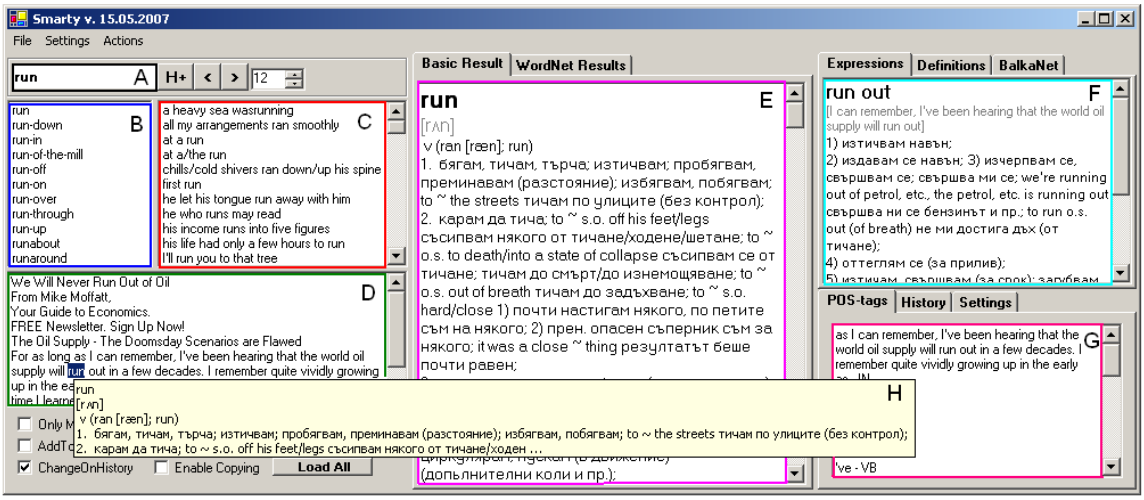


Fig 6. Input and output boxes in standard mode

A. Query Textbox
B. Suggested (Predicted) queries
C. Alphabetically sorted known multi-word expressions
D. Free text input
E. Basic dictionary output
F. Suggested multi-word expression
G. Auxiliary PoS-tagged output
H. Optional definition/expression tool tip

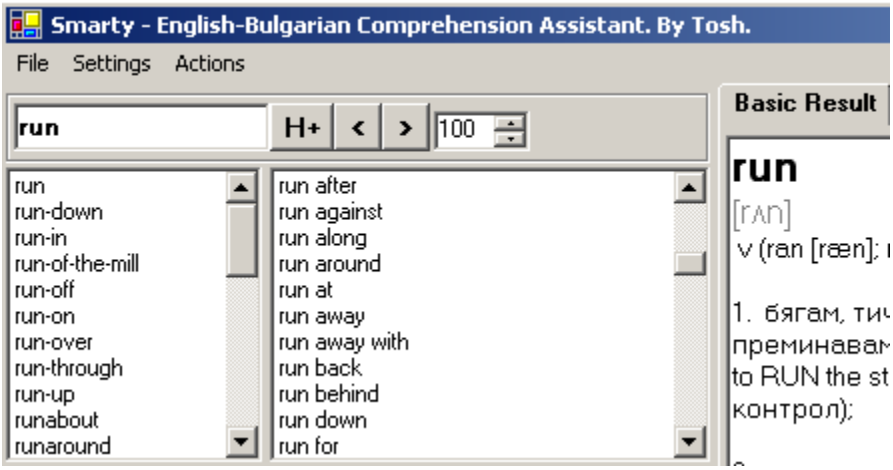


Fig 6. Predicted queries and the list with multi-word expressions

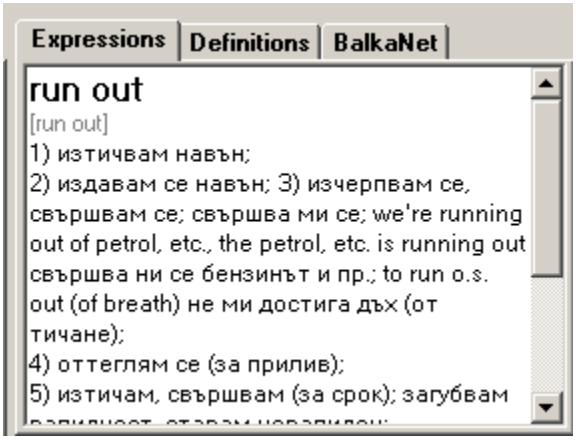


Fig 7. Multi-word expression translation

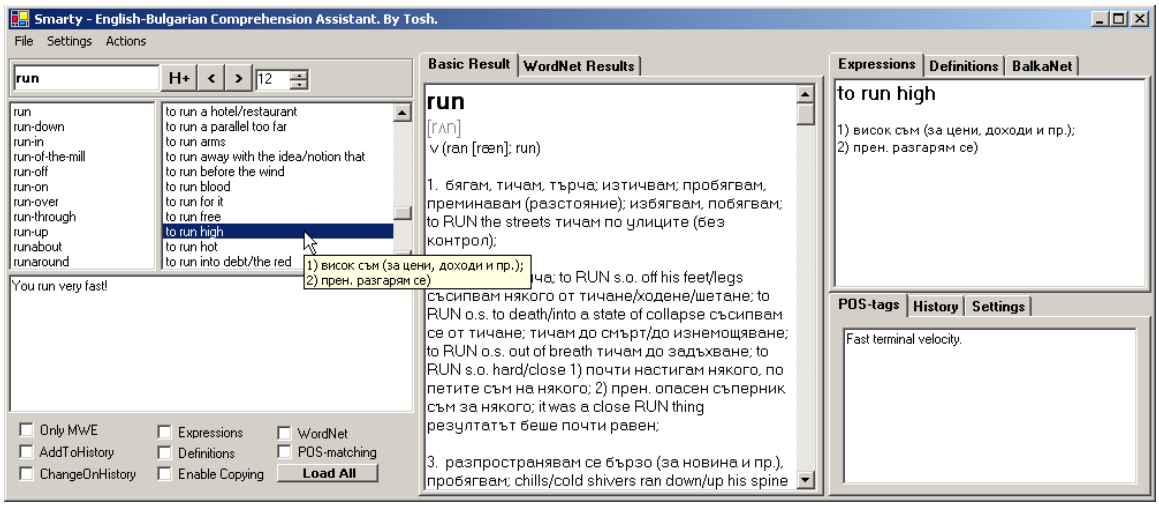


Fig 8. Multi-word expression tooltip

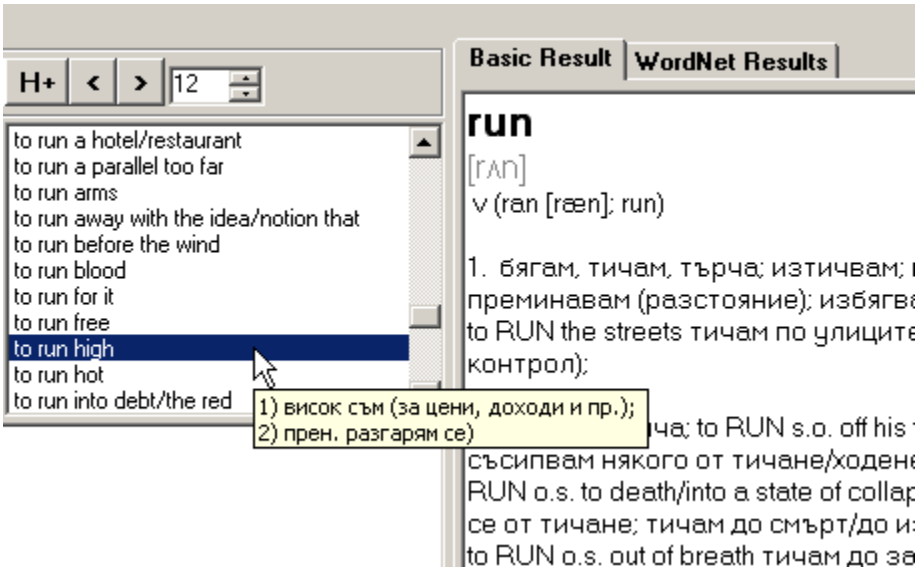


Fig 9. Multi-word expression tooltip – closer view

3.2.1.2. WordNet/BalkaNet Mode with Word Sense Disambiguation

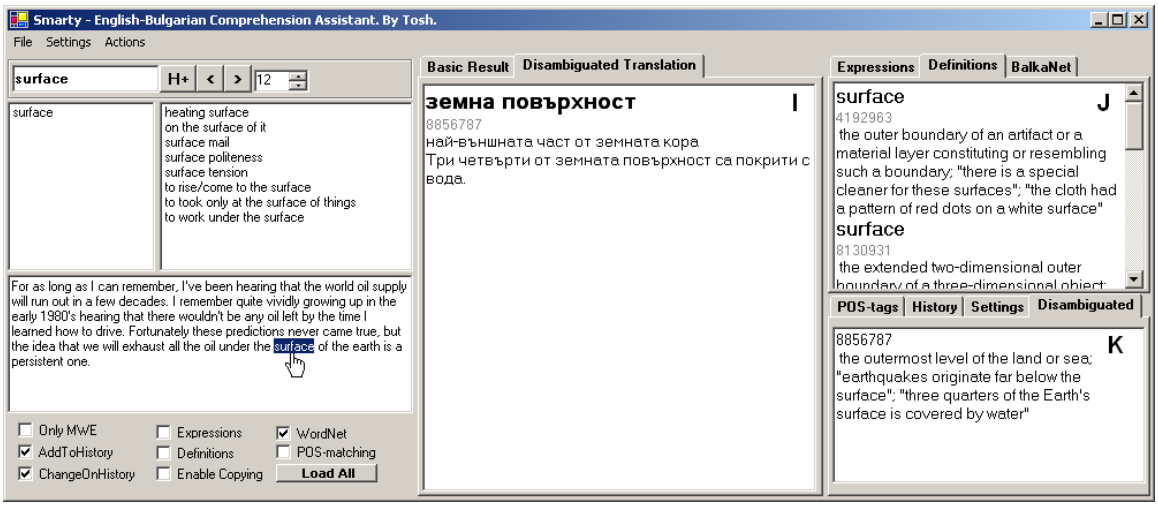


Fig 10a. Output windows boxes in Word Sense Disambiguation mode

I. Disambiguated translation

J. Full list of possible translations from BalkaNet

K. Disambiguated sense in English from WordNet.

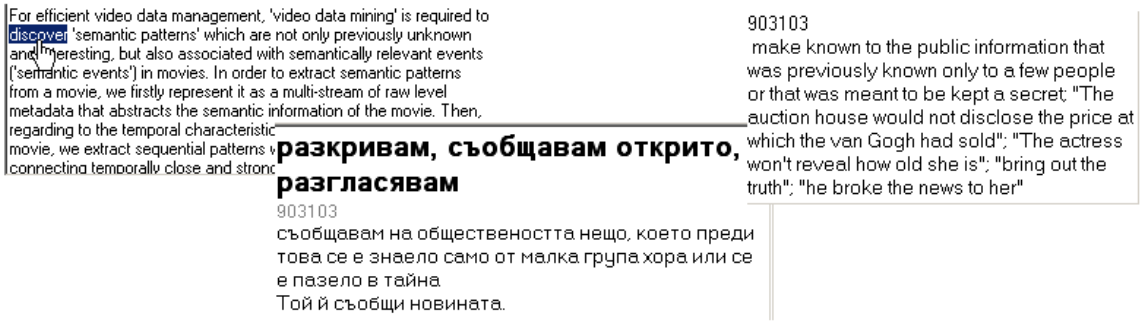


Fig 10b. Word Sense Disambiguation windows

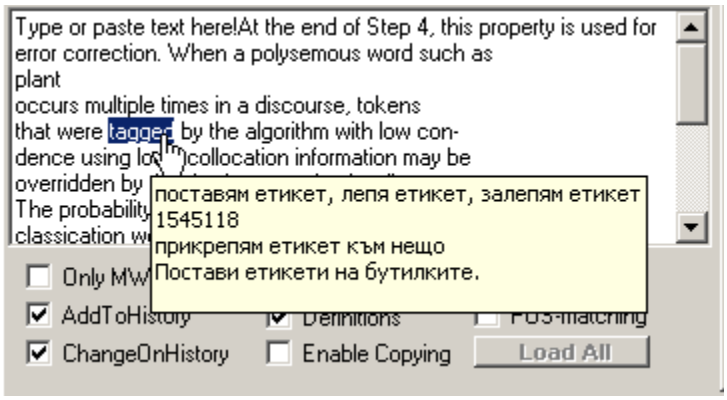


Fig 10c – Disambiguated translation may appear also as a tooltip, on right mouse click

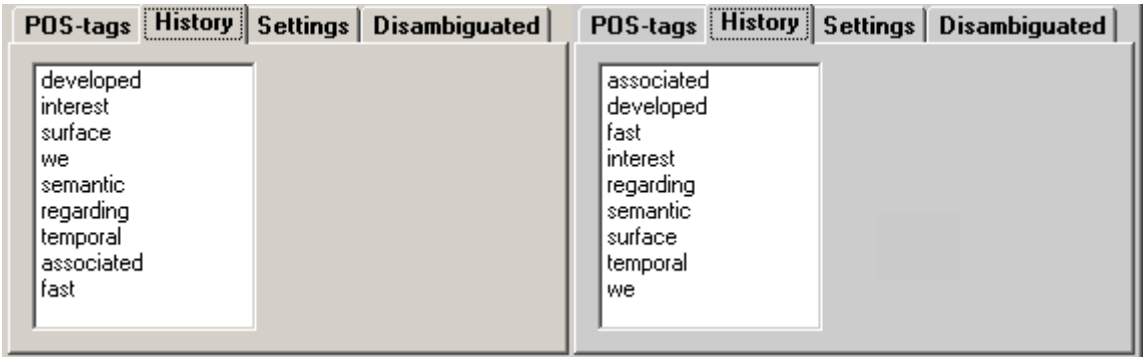


Fig 11. History of queries could be sorted by time and alphabetically

L. History of queries

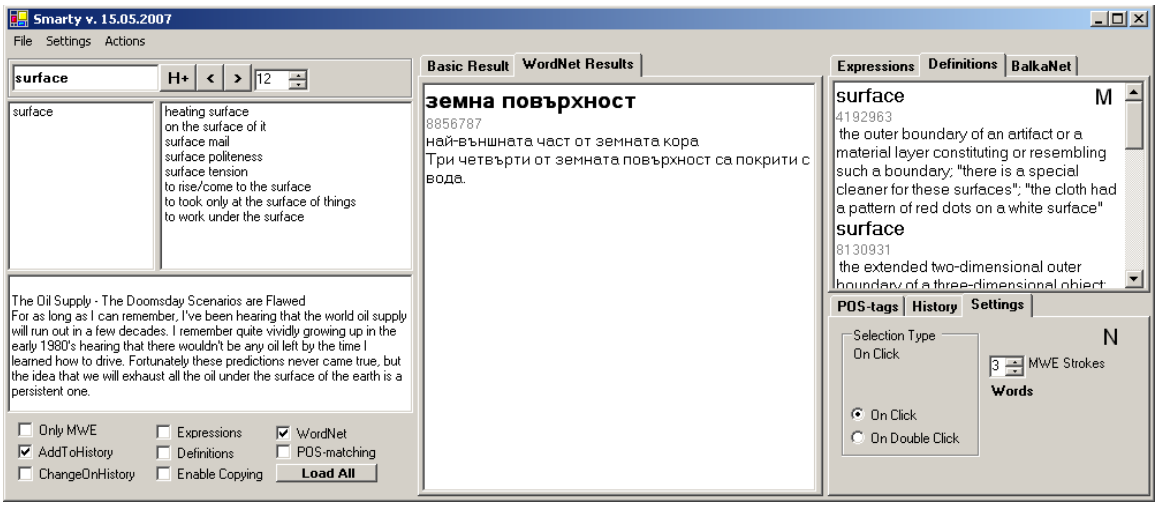


Fig 12.

M. Full list of senses from WordNet

N. Auxiliary options panel

3.2.1.3. Other Controls and Options

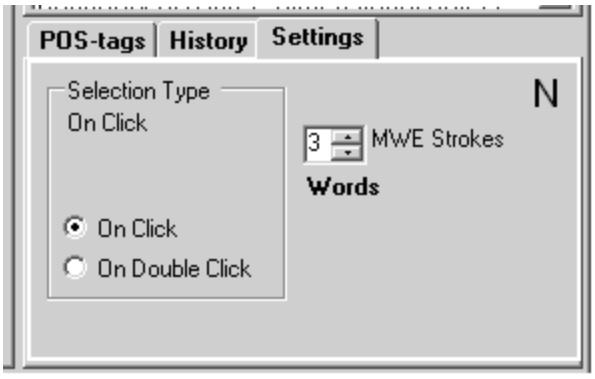


Fig 13. Options

N. Auxiliary options panel

O. Main options

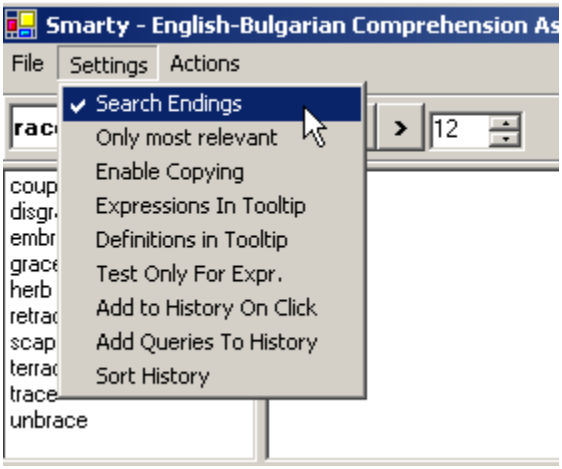


Fig 14. Suffix search

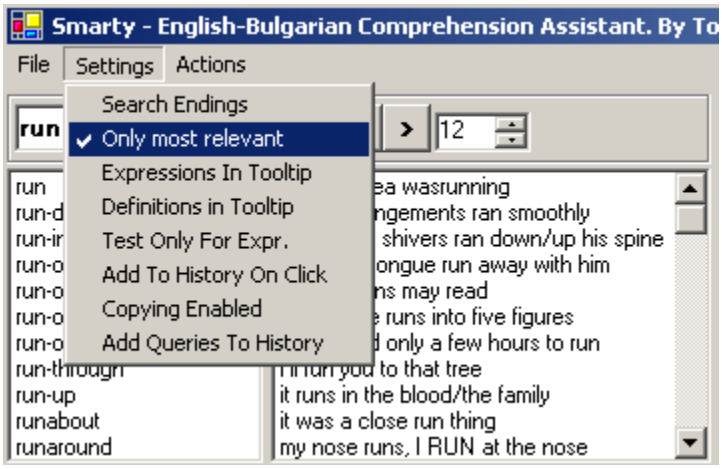


Fig 15. Display only the entries with matching PoS-tag

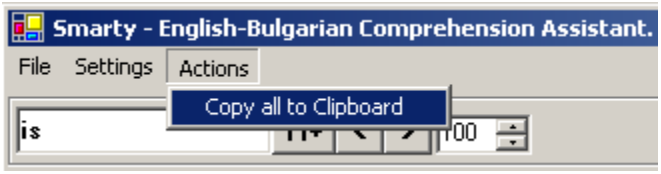


Fig 16. ***Copy All to Clipboard*** *- all text in all windows of the dictionary is copied*

Fig 17. Query Panel

Q. Query panel

- Query Texbox

- Add to history list

- Move a step back/ahead in history list

- Maximum number of predicted query suggestions

3.2.2. Interface in Details

3.2.2.1. Basic Input

The basic mode of operation looks similar to the operation of conventional dictionaries – that is when the user types or pastes a word or a short expression in the query textbox.

In case of typing, the program tries to answer user's query as fast as possible. On each keystroke (or after text is pasted):

1. A search through the words database is performed and a list of words **beginning like the query** is shown.
2. Suggestions about possible **continuations** (default mode) or possible **beginnings** (suffix mode – menu P) are displayed.

3. Without additional user actions**, appropriate dictionary entries, matching the query, are immediately displayed**. If the entered query does not match an entry from the list with suggestions, the program "makes a guess" and displays the one which is most similar.
4. Once a few keystrokes – the number is specified in N – or when Enter is pressed, **the query is matched to the list of multi-word expressions**. **This is useful for checking compound forms like "run away", "go down" etc**. Suggested expression – either the correct one or just an example - is displayed in window F.

   Multi-word expressions matches could be based **on the lexemes** or on the **part-of-speeches** – that is specified in N and O by clicking on "**Words**" or by checking "**PoS-matching**" check-box.

5. If the user presses Enter, the PoS of the word is determined (however, it is ambiguous without context) and the dictionary word is **lemmatized**; yet the dictionary takes into account the original input, also.

   In case of existence of entries for both lemmatized and original form, the dictionary puts on the top of the basic output the entry for the original form, even in cases when the PoS-tag doesn't match, presuming that this entry could be the closest to the query, since the PoS-tagger is not 100% accurate.

   The lemmatized and PoS-tagged word is also checked in WordNet database. If the word exists in a synonym set, a list of all relevant definitions is displayed in M; after that, the program aligns the results with BalkaNet and if corresponding synsets are found, all possible definitions in Bulgarian are also shown in windows J and I.

Even in this basic mode, the intelligent dictionary shows up as more powerful tool than its standard English-Bulgarian dictionary rivals, due to the *Multi-Word Expressions matching*, *WordNet and BalkaNet look-up and*

suffix-searching.

These functions will be discussed below.

3.2.2.2. Free Text Input

The real advantages of "Smarty" over simple dictionaries emerge as soon as the user paste or write complete text in the Free Text Input window. Now the context around the word could be effectively used for narrowing the number of possible translations.

In this mode, the sequence of actions is as follows:

1. User clicks on a word in any window with text, including windows with results from previous queries with either the left or the right mouse button.
2. Context and the word are tokenized and PoS-tagged.
3. The word is lemmatized.
4. The most relevenat entry from the standard dictionary is decided like in point 5. for the basic input.
5. Query textbox is filled with the word and suggestions for other forms of the word are generated.

If the left mouse button is pressed and tooltip-options **"Expressions"** or **"Definitions"** are checked, a tooltip with the top lines of the definition or with suggested expression are shown near the mouse pointer.

If the right mouse button is pressed, then a word-sense disambiguated translation appear as a tooltip, after passing points 7. and 8.

If **"Only MWE"** is checked, the program does not translate clicked words, but only tries to match multi-word expressions.

"On click" or **"On double-click"** in the auxiliary options panel N select the way the dictionary is queried from the free text input.

“Enable copying” disable translation on click and allows user to select and copy text from any of the output or input windows.

“Add to history” – add clicked words to history list.

“Change on history” – when a word from the history list is clicked, the query textbox is filled with the pointed word and forms and additional queries are predicted.

6. Multi-word expression matching is performed and an expression is suggested.
7. Relevant definitions from WordNet and BalkaNet are extracted and displayed.
8. Word-sense disambiguation is performed on the context. A disambiguated gloss in English is shown and **a disambiguated translation is suggested in Disambiguated Translations window I or in a tooltip.**

3.2.2.3. Tooltips

Tooltips with the top part of a dictionary entry, or with suggested multi-word expression, pop-up after clicking the left mouse button. The right mouse button activates a tooltip, containing disambiguated translation in Bulgarian.

A tooltip window with translation of a multi-word expression pops-up after a click on the list with expressions, and the content of that tooltip window could be easily copied to the clipboard by pressing F3.

3.2.3. SharpNLP PoS-tagger

Part-of-speech tagging, or PoS-tagging is one of the crucial NLP-techniques, used in “Smarty”.

Pre-processing tokenization and then PoS-tagging are done every time a dictionary is queried with context or with single word.

Smarty uses GNU SharpNLP PoS-tagger. Details about its precision are not available, but it is considerably robust for a test system.

This is a list of the most important tags that SharpNLP produces:

NN, NNS – Noun in singular, Noun in plural.
VB - Verb in present tense.
VBZ - Verb in present in 3-rd person.
VBG – Ing-form of a verb.
VBD – Verb in past tense.
VBN – Verb participle.
JJ – Adjective
RB – Adverb
IN - Preposition
DT - Determiner

A subtle post-processing of the output is done, in cases of strings of attributive nouns. SharpNLP tags them as mere nouns, while “Smarty” turns the attributive nouns to adjectives.

Also, in the process of lemmatization, words that ends in “-ing” and are tagged as nouns (NN), are regarded as verbs.

3.2.4. Token-With-Context Extractor

This component tokenizes the word that is pointed with the mouse in Free Text Input window or any other window with text, and extracts the context around it, using tokenizer. The tokenizer extracts whole words only, thus avoids sending randomly cut off parts of words to the PoS-tagger.

3.2.5. Predictive Typing Processor

Predictive typing processor is based on a tree structure, internally called "LettersNet", which describes a deterministic finite-state automaton, representing all the words from the standard dictionary database.

Positions in the words (first letter, second letter, etc.) are states, and the letters are transitions between states.

Each node can be continued with none, with zero or with all of the 26 letters of the English alphabet and with some special characters – space, hyphen and apostrophe. Last positions of the words are marked as final states of the automaton.

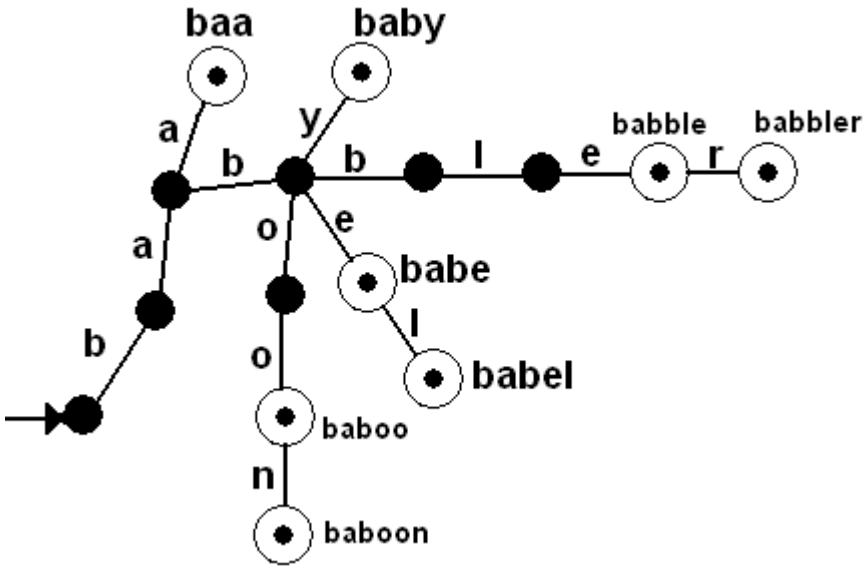


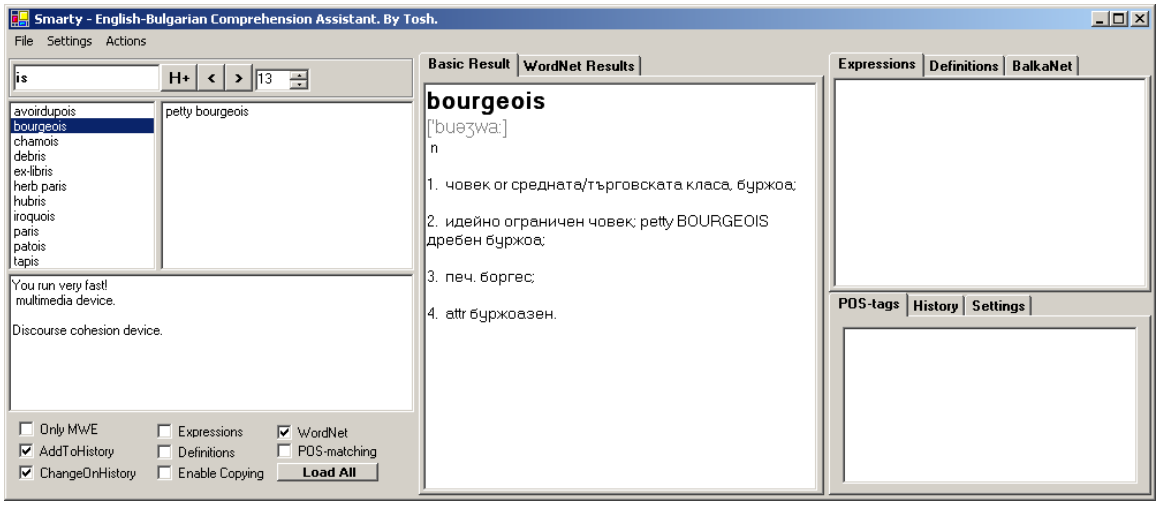
Fig 18. "LettersNet" finite automaton

This structure provides linear complexity algorithm for search and almost "free" string sort, but at the cost of memory overhead.

Smarty uses two "LettersNet" structures – one for default mode, built from first to last letter of the words; and one reversed - i.e. the last letters of the words are fed first – which is used in suffix-mode.

3.2.5.1. Suffix Mode

This mode is useful for exploration of words from Latin, French etc. origin, and for writing poetry - rhyme search.



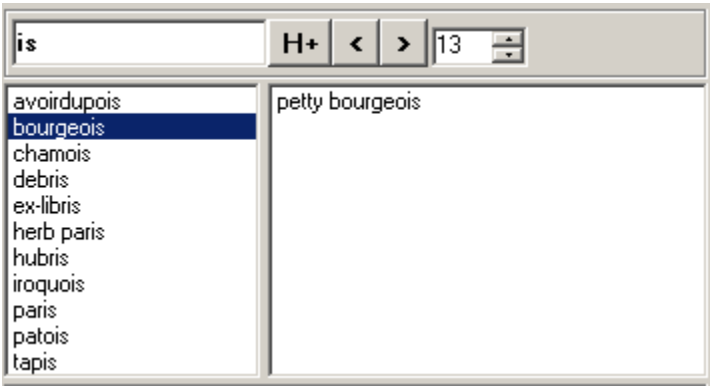


Fig 19. "Smarty" in suffix mode

We don't know about any other English-Bulgarian dictionary,

capable of suffix-search.

3.2.6. Dictionary Entries Analyzer

This is the component which parses dictionary entries and extracts multi-word expressions. It does analysis on-the-fly, when a word is queried.

Due to the lack of a clear format definition and because the database is a scan of a paper dictionary, supposed to be read by humans, we had to build up rules which explore the entries like humans do. “Condensation” of the data made some troubles (see also 3.1.1).

The result is a complicated scanner, which has to take into account open and closed brackets, numbering and abbreviations. The scanner works together with the Normalizer (see below), which supplies the Analyzer with different forms of the words, in order to let it capture variations.

3.2.7. Normalizer

The role of the Normalizer is to find the lemma and, in some cases, to generate forms of a given PoS-tagged word .

Lemmatization part of the Normalizer is used for every query, and is activated either by typing or by clicking in any input window. Lemmatization is used also by Multi-Word Expression Matcher and by Word-Sense Disambiguator.

Forms Generator helps Dictionary Entries Analyzer to detect multiword expressions, because it does not lemmatize the whole context but does scan it.

In Basic mode, Forms Generator can expand a verb in present tense, to verb in past tense and participle.

3.2.7.1. Lemmatizer

Lemmatization of nouns and verbs in “Smarty” is based on simple morphological rules, a list with irregular verbs and irregular plural nouns, and the standard dictionary database.

For example, the rules for verb lemmatization are given below, where *Verb* is the input string.

0. If the Verb is any of (be, is, am, are, was, were)
   Then Return “be”
  Else
  If the Verb is “has”
  Then return “have”
  Else
  If the Verb is in any of (lay, lain, lied)
  Then Return “lie”
1. If the Verb is found in the list of irregulars – return the Infinitive.
Else
2. If the Verb ends with “ed”
   3. If in the Dictionary exists a form which is same like the Verb without the last “d”.
    Then Return that form. //scribed - scribe
    Else
    4. If the Verb ends with double consonant before “ed” and it is not “l” //rolled - roll
      Then Return the Verb, shorten with 3 characters //rib – ribbed
    Else
    Return the Verb, shorten with 2 characters.
 Else
 5. If the Verb ends with “ing”
    6. If the Dictionary has a form, which is like the Verb without the last “ing”, + “e”.

Then Return the form from the Dictionary. //taking – take

Else

7. If the Verb ends with double consonant before “ing” and it is not “l” //calling - call

Then Return the Verb, shorten with four characters. //running – run

Else

8. Then Return the Verb, shorten with three characters. //flying - fly

Else

9. If the Verb ends with “ss”

Then Return the Verb

Else

10. If the Verb ends with “ies”

Then Return the Verb, shorten with three characters and add “y” //flies - fly

Else

11. If the Verb ends with “es”

If the Dictionary has a form which is like the Verb without the last “s”

Then Return the form from the Dictionary. //takes – take

12. If the Verb ends with “s”

Then Return the Verb, shorten with one character.

Else

Return the Verb.

3.2.7.2. Forms Generator

It uses similar algorithms, but in reverse direction in order to construct a list with forms. However, this sub-component was not fully developed, because only a generator of present-past-participle forms of verbs was needed.

3.2.8. Multi-Word Expressions Matcher

After Dictionary Entries Analyzer propagates a list of expressions, detected in the relevant active dictionary entry, Multi-Word Expressions Matcher is supposed to recognize an expression in the context of the pointed word.

Even if the Matcher is unable to recognize one particular expression with certainty – which is often the case – the Matcher suggests one which looks most similar to the context anyway. In the worst case, the example may just enrich user's knowledge.

Basic characteristics of the Matcher are:

- Two matching algorithms are selectable: **Words-Matching** and **PoS-Matching**.
- Multi-word expressions are stored in a format, regarding the "back" and "forward" content of the expression, where the main word is in the "centre". E.g.:

> *"he let his tongue run away with him"* is stored as:
> **Back**: he let his tongue
> **Center:** run
> **Forward:** away with him

- Before performing matching, the tokens from the context are lemmatized in order to capture variations.
- The context that is to be matched is also stored not just as simple text, but with "back" and "forward" subsections for better accuracy. Appropriately sized sub windows of words from the context - back and ahead - are selected for each particular multi-word expression that is matched.
- Both algorithms rely not only on the absolute number of matched tokens, but also take into account the ratio of matched tokens to number of tokens, and the length of the matched expressions.

- Fully matched expressions have higher priority than partially matched, but fully matched longer expressions have higher priority than fully-matched shorter ones.
- In this version “Smarty” always guesses only one expression, supposed to be most similar to the context.

3.2.9. WordNet and BalkaNet Wrapper

This component index, query and link together WordNet and BalkaNet lexical databases.

One of the scenarios of using the Wrapper is in standard mode, when no context is given. Then the Wrapper is queried with a “bare” token.

The Wrapper PoS-tags the token. If the token with the tagged part-of-speech exists, the Wrapper returns a list, consisting of a synonym set and glosses/examples-of-usage for each member of the synset.

Glosses are displayed in window M with their synset offset for reference.

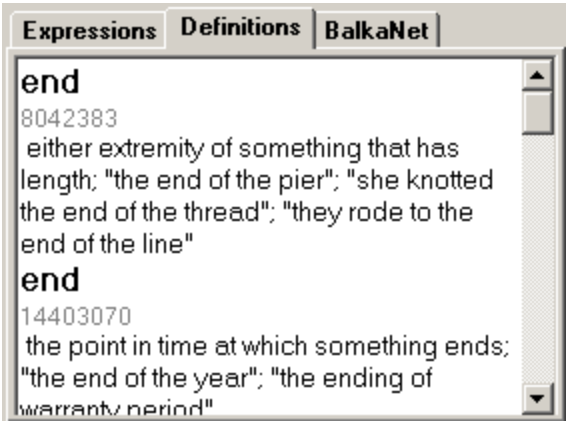


Fig 20. WordNet definitions

Then BalkaNet is checked for corresponding synsets; if found, they are displayed, also:

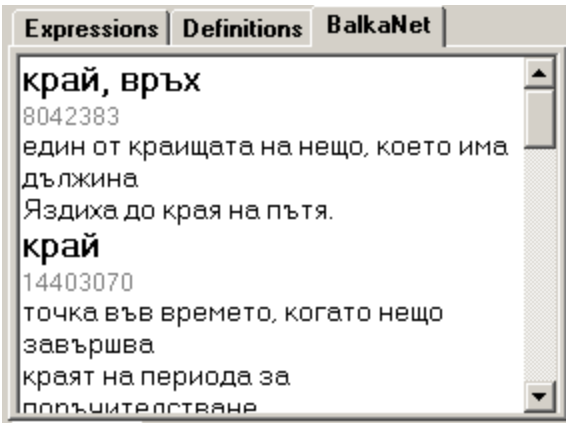


Fig 21. BalkaNet definitions

The second scenario is when the Wrapper is queried with a word, pointed in a context. That is the moment when the final and most advanced component of the system is activated**;** the module which undoubtedly distinguish "Smarty" from all the rest English-Bulgarian dictionaries and left the basic version of Xerox Locolex slightly behind **– Word-Sense Disambiguator**.

3.2.10. Word-Sense Disambiguator

Ultimate goal of a comprehension assistant is to provide only the most relevant translation of a word in a given context. To accomplish that, the first comprehension assistant – LocoLex – applies PoS-tagging and Multi-word expression recognition. "Smarty" do it as well, but we also attempt to extend these methods.

3.2.10.1. Lexical Ambiguity

"Lexical ambiguity is a fundamental characteristic of language: words can have more than one distinct meaning. The 121 most frequent English nouns, which account for about one in five word occurrences in real text, have on average 7.8 meanings each".[6] General verbs like "run", "take" and "go" have even more than 40 senses each in WordNet 2.0.

3.2.10.2. Word-Sense Disambiguation

The process of recognition of the right meaning of a homograph or a polysemy word, used in particular context, is called **word-sense disambiguation (WSD).**

There are three main approaches for word-sense-disambiguation: [7, Ch.13]

1. Dictionary-based – introduced by Lesk in 1986. The method of Lesk and its modifications are based on the observation that the words in dictionary definitions could be used as "semantic tags" which represent connections between concepts.
2. Connectionists – they are based on researches in psycholinguistics.
3. Statistical and machine learning – the rules for WSD are learned from semantic tagged examples and aligned bilingual corpora, which help different senses to emerge.

3.2.10.3. Word-Sense Disambiguator in Smarty

Due to the limited time and resources we had for our research, only a basic variation of the ideas of Lesk was implemented.

We use WordNet and BalkaNet as aligned semantic tagged dictionaries with glosses.

The process of WSD in "Smarty":

1. A word in a text is pointed.
2. The word is tokenized and PoS-tagged together with the context.
3. The word and the context are sent to Word-Sense Disambigator.
4. Synsets from WordNet are found and glosses are extracted.
5. Context is lemmatized.

6. Lemmatized Context is cleaned from stop-words, considered to be confusing for the WSD: prepositions; “the”, “a”, “to”; pronouns, adverbs, conjunctions, numbers.
7. A cycle is started for each one of the synonym sets extracted.
8. Each gloss is tokenized and part-of-speech tagged.
9. The gloss is lemmatized.
10. Lemmatized gloss is cleaned from irrelevant and stop-words. (See 6)
11. The Context and the gloss are matched and number of word-matches is counted.
12. Until there are more glosses – go to 8. Else:
13. The gloss with higher number of matches is suggested, if there is one.
14. The synonym set of the suggested sense is matched to the synonym sets of BalkaNet.
15. If BalkaNet contains the disambiguated sense, then disambiguated translation in Bulgarian is displayed:

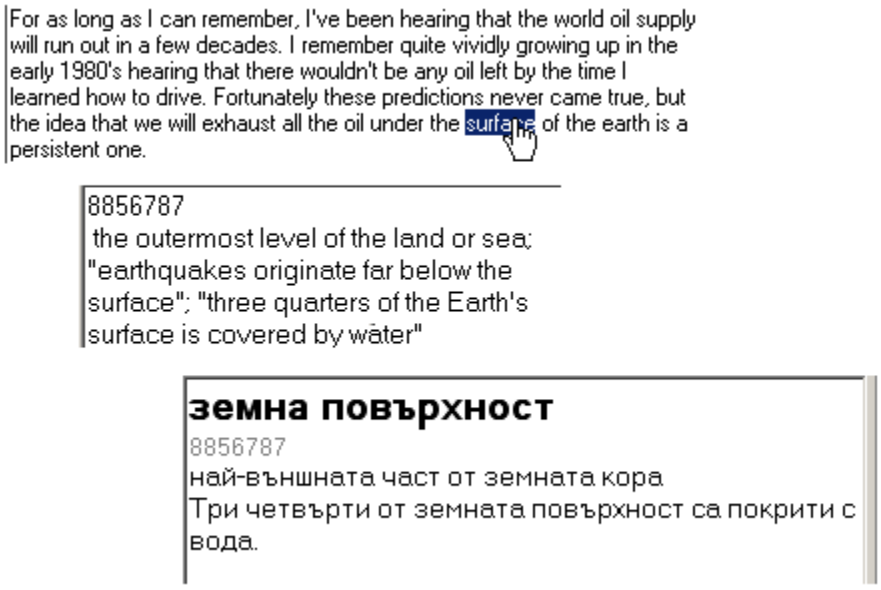


Fig 22. Word sense disambiguation windows

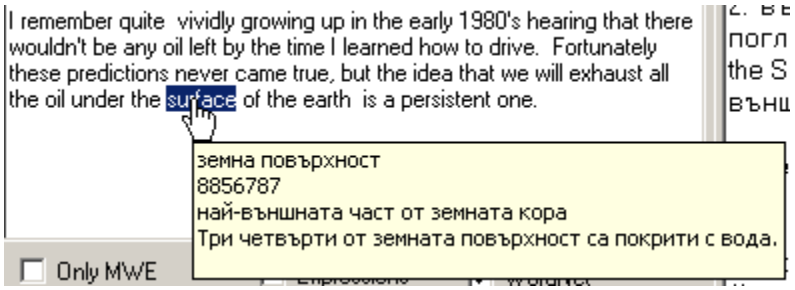


Fig 23. Disambiguated translation may appear as a tool tip, also.

Otherwise, if the synset exists in BalkaNet, but the precise sense is missing, then all senses are displayed with a question mark.

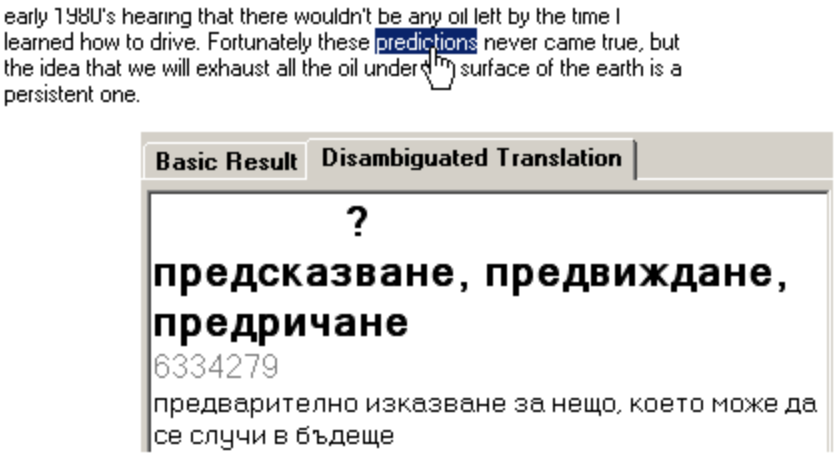


Fig 24. Uncertain disambiguated translation

In case the Disambiguator is unable to decide which one is the most appropriate sense, all senses from BalkaNet are displayed with a question mark, also.

3.3. Future Work

We may suggest a number of thinkable improvements:

- More sophisticated multi-word expressions matching with improved scoring. In case of uncertainty, a list with suggestions to be presented to the user, instead of one-single suggestion.
- More sophisticated implementation of Lesk ideas for Word Sense Disambiguation, and exploiting the semantic relations in WordNet and BalkaNet, which yet are not utilized.

Bulgarian Academy of Science is developing an extension of Bulgarian BalkaNet, called BulNet [3], which aims to increase the number of synsets with 15000, up to over 36000. This database will improve the performance of "Smarty" even without changes in Word Sense Disambiguation algorithm.

- Access to corpora and searching for passages and sentences in English, similar to the given in the input, via various matching techniques, semantic-distance evaluation and word sense disambiguation at paragraph-level.

Variations of HyperLex or/and PageRank graph algorithms, based on [1], might be tested. Although this would not be a translation, multiple examples may help reader to figure out patterns, thus - the meaning.

- Enriching multi-word expressions database.
- Development of Bulgarian-English direction in the system, intended for translation assistance.
- Web functions – Internet search for information or passages, which are relevant or similar to the context. Querying on-line dictionaries and combining results from many sources.

Several little tools are considered:

- Spell-checker – to correct typing mistakes when querying words in basic mode.
- Phonetic search – to display words that have similar transcriptions, not similar spelling.

4. Evaluation

4.1. Evaluation Method

Formal guideliness for comprehension assistant evaluation are not available, that is why our evaluation will not be based upon a formal statistical comparisons.

"Smarty" is compared to other dictionaries and comprehension assistants, and advantagous aspects of "Smarty" are discussed, as well as its weak sides.

4.2. Interface Evaluation

4.2.1. Advantages

In comparison with the other English-Bulgarian dictionaries, "Smarty" has more sophisticated and powerful user interface.

Translations on-click, multi-word expressions recognition and the popping-up tooltips tooltips charmed the few users who have seen or tried "Smarty", during development process. These features result in significant speed-up in reading texts, which contain high number of unfamiliar words, because if using a dictionary with conventional interface, the user is supposed to switch manually between the window with the text and the window of the dictionary for every single word.

History list appears to be another useful element, when we come across a series of new words and we do need quick access to their translations, especially when working with texts on paper. Once entered, words could be quickly looked-up again without retyping or copying from do-it-yourself list from a text-editor.

4.2.1. Weaknesses

Interface might be more flexible – some windows to be hidden or made bigger or smaller; fonts faces and sizes should be changeable.

History should contain part-of-speech and sense number/multi-word expression information, and the context where the word was found. Saving on disk should be available.

An option for tooltips to appear without clicking, but immediately on pointing, also might be useful.

Standard dictionary entries output should be formatted better.

4.2.2. Smarty vs LocoLex

LocoLex laid the foundations of comprehension assistant architecture and user interface, concentrating only on most appropriate translation, presented in minimalistic window. However, besides that, an option for browsing dictionary entries the standard way enriches the system, because some dictionary users actually like to read all the meanings and to explore dictionaries. The same goes for the Suffix-search option, because it is intended to assist *browsing* for words with common endings.

4.3. Multi-Word Expressions Matching

As could be expected, better results are achieved in words-matching mode than in PoS-matching mode.

Shortest form of multi-word expressions, e.g. phrasal verbs like “run in”, “run out”, “take away” are almost always recognized correctly.

Longer expressions sometimes collide and the matcher returns multi-word expression with similar wording to the context, which is not the right one.

However, in the most cases expressions are correctly recognized, and the results are adequate, even thought the algorithm should be improved in future versions.

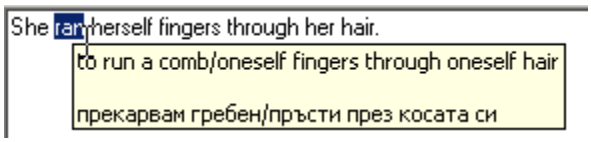


Fig 25. Multi-word expressions matching

4.4. Word Sense Disambiguation

The method we implemented is basic (see 3.2.10.3), yet it achieves successful results, when appropriate words from correspondent WodNet gloss are found in the context of the pointed word.

Unfortunately, the version of BalkaNet database that we use for providing disambiguated translations in Bulgarian consists of only 15000 synsets – compared to 115000 in WordNet 2.0, - and usually each of the synsets from BalkaNet has several times less senses than the corresponding synset in WordNet.

This is the reason why often correctly disambiguated English words could not be translated to Bulgarian at all, or if they can, the correct sense can not be found in the corresponding BalkaNet synset.

Nevertheless, the right gloss in English appears to be useful, because the user is supposed to have background in English.

4.4.1. Examples of correctly disambiguated and correctly translated senses

- *John works in a **bank**! He must have a lot of money…*

банка
7909067
финансова институция, която разполага с капитал и осъществява различни финансови операции
В тази банка направих ипотека на къщата.

bank - a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home"

- *What instrument do you play, Paul?*
- *I play the **bass**.*

бас - музикален инструмент от група, който има най-ниско звучене

2330754
bass - the member with the lowest range of a family of musical instruments

- *You are **fired**!*

уволнявам, отстранявам от работа
2330754
връчвам предизвестие за прекратяване на трудовия договор

2330754
fire - terminate the employment of; "The boss fired his secretary today"; "The company terminated 25% of its workers"

4.4.2. Example of correctly disambiguated word in English, without corresponding translation in BalkaNet:

*Mary walked to the **bank** and jumped in the river.*

8639924

Bank - sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"

?

банка
7909067
финансова институция, която разполага с капитал и осъществява различни финансови операции
В тази банка направих ипотека на къщата.

рид, насип
8639793
продълговато хълмисто възвишение, образувано най-често от ерозирането на външните земни пластове или на морското дъно
голям земен насип

5. Conclusion

This paper has presented the framework of the next generation intelligent multilingual computer dictionaries, called **comprehension assistants,** and the architecture of the newly developed English-Bulgarian comprehension assistant **"Smarty".**

"Smarty" successfully applies the basic framework, which could be briefly described as: **Point a word in context, Part-of-speech tag** and **Multi-word expression recognize.** In addition, it extends the basic framework with **bilingual Word sense disambiguation.**

Although the success of this most advanced feature was limited due to insufficiency of linguistic resources, "Smarty" proved that the method is applicable and provides useful results.

Using BulNet - a new WordNet for Bulgarian with more synonym sets - as well as development of more sophisticated Word Sense Disambiguation algorithms, which utilize the set of semantic relations in WordNet, would improve results further.

Suggesting example sentences, selected from corpora by applying semantic similarity measures and word sense disambiguation, would provide even better comprehension assistance.

6. References

1. Agirre, E; Martinez, D.; Lacalle, L and Soroa, Aitor (1998). **Two graph-based algorithms for state-of-the-art WSD.**

2. **BalkaNet - Design and Development of a Multilingual Balkan WordNet**. http://www.ceid.upatras.gr/Balkanet

3. **BulNet - Bulgarian WordNet**. http://dcl.bas.bg/BulNet/general_en.html

4. Brun, Caroline (2000): **A Client/Server Architecture for Word Sense Disambiguation**.

5. Dini, Tomasso, Segond (1999*): "GINGER II: an example-driven word sense disambiguator"*, **Computers and the Humanities**, 34 (1--2) (2000), pp 121--126

6. Edited by Agirre, Eneko and Edmonds, Philip (2006). **Word Sense Disambiguation**, **Introduction.**

7. Edited by Mitkov, Ruslan (2003), **The Oxford Handbook of Computational Linguistics**;
- Grefenstette, Gregory and Segond Frederique. **Chapter 38 – Multilingual On-Line Natural Language Processing.**
- Stevenson, Mark and Wilks, Yorick. **Chapter 13 – Word-Sense Disambiguation.**.
- Voutilainen, Atro. Chapter 11 – **Part-of-Speech tagging.**

8. Feldweg, Helmut and Breidt, Elisabeth (1996). *"COMPASS An Intelligent Dictionary System for Reading Text in a Foreign Language"***.** In Kiefer, F. and G. Kiss, editors, **Papers in Computational Lexicography**, COMPLEX '96, Budapest, pages 53-62.

9. Ide, Nancy & Véronis, Jean (1998), *"Word sense disambiguation: The state of the art"*. **Computational Linguistics**, 24(1):1–40.

10. **Princeton Univeristy WordNet 2.0 Database documentation** - wnstats.7WN.html.

11. **Princeton University WordNet Website**: http://wordnet.princeton.edu/

12. Prószéky, Gábor (2002), *"Comprehension Assistance Meets Machine Translation"*. In: Toma3 Erjavec; Jerneja Gros (eds) **Language Technologies**, 1–5. Institut Jo3ef Stefan, Ljubljana, Slovenia.

13. **SharpNLP – open source natural language processing tools**. http://www.codeplex.com/sharpnlp