

## **CHAPTER III**

### **EVALUATION METHODOLOGY FOR TERMINOLOGY MANAGEMENT TOOLS**

#### **1. INTRODUCTION**

In order to fully appreciate the role of terminology management systems in translation, we need to look deeply into their structure, functionalities and linguistic behavior. A comparative study of terminology management tools gives us an in-depth and objective knowledge which can be very helpful in deciding whether we should consider using them in a given translation project or not<sup>16</sup>. Therefore, after the theoretical introduction presented in chapter I, the presentation of the basic functionalities of terminology management systems in chapter II, we are now ready for a more specific insight into the systems in question. The present chapter will be devoted to the software evaluation methodologies in general, as well as to the evaluation methodology used in the testing procedure conducted within this thesis.

First, a basic introduction to software testing methodologies will be presented. We will discuss the differences between the black box and glass box testing techniques. Next, the readers will find a detailed classification of natural language processing (NLP) software methodologies. This will include descriptions of scenario testing, systematic testing, and feature inspection. Following this introduction, a description of the background methodology of the evaluation procedure applied in this thesis will be presented in more detail. Finally, we will put forward the feature checklist compiled for the needs of the evaluation procedure. Having assumed that all the necessary notions and explanations concerning the common features of terminology management tools were already presented in chapter II of the present thesis, the author decided not to include any additional comments in the checklist.

---

<sup>16</sup> A number of comparative analyses of CAT tools have been carried out recently, e.g. Translation Memory Systeme im Vergleich (Translation Memory Systems in Comparison) by Massion (Massion 2002), Feder (Feder 2001) and Palacz (Palacz 2003); however these analyses did not have terminology management as their central focus.

## **2. SOFTWARE TESTING METHODOLOGIES**

Software testing is an extensive branch of computer science. There is no consent among software engineers on how to test different types of software, not to mention the existence of consistent terminology in this area. The classification of test types as presented in the report written in 1995 by the Expert Advisory Group on Language Engineering Standards (EAGLES 1996) is just one of the many already existing. This proposal relates to different test types that can be conducted on NLP systems. Hence, it is relevant to our study. However, before we become familiar with this specific classification, we should first be introduced to a more general classification.

### **2.1. GLASS BOX vs. BLACK BOX TESTING**

The most general classification divides testing methodologies into black box and glass box techniques. Black box testing is an approach to software evaluation in which only input and output behavior can be observed (EAGLES 1995:32). This definition implies that black box approach is not concerned with how a given output is obtained. The evaluation is conducted on the basis of the actual output produced by the software tested. While black box testing is only interested in input and output material (in the case of NLP it focuses on the linguistic productivity of the systems tested), glass box approach takes into account the underlying architecture of the systems i.e. their internal workings (Feder 2001:69). The glass box approach to software testing requires expert knowledge of the system's architecture, which involves the source code and a number of data which are not made available to the end users. Glass box testing is usually conducted by software engineers who created the software under evaluation.

## **2.2. CLASSIFICATION OF NLP SOFTWARE EVALUATION METHODOLOGIES**

Let us now introduce the classification of testing procedures of NLP systems, as presented in the Eagles Report (EAGLES 1995).

### **2.2.1. SCENARIO TESTING**

Scenario testing entered software evaluation in 1990s (EAGLES 1995:33). This kind of test aims at using a realistic user background for the evaluation of software. It is a typical example of black box testing. In scenario testing the suitability of the product for everyday

routines is subject to evaluation. It usually involves putting the system into its intended use by performing standard tasks. Scenario testing provides good empirical information concerning the usability of the product. It also supplies information on accuracy, adaptability, operability etc. of the software.

**a. Field testing**

Field testing is a type of scenario testing in which the testing environment is the normal working place of the user. The user is usually observed by one or more evaluators. Tasks designed for the field tests can include problematic issues such as data transfer difficulties between the systems tested and other systems. Different records of these tests are used, e.g. notes on evaluation checklists, pre- post- testing interviews, thinking aloud protocols (TAP), log file recording, etc.

**b. Laboratory testing**

It is a type of scenario test in which a number of isolated users perform a given task in a test laboratory. It is advocated to apply laboratory tests to the systems that are not fully operable. The costs of laboratory tests are around four times greater than in the case of comparable field tests (EAGLES 1995:34).

**2.2.2. SYSTEMATIC TESTING**

Systematic tests examine the behavior of a system under specific conditions and with the specific results expected. They can be performed only by software engineers or user representatives. Systematic testing has a number of sub-types.

**a) Task-oriented testing**

Task-oriented tests aim at verifying whether the software under evaluation fulfils the pre-defined tasks (stated in requirements specification document or implied by third parties e.g. consumer reports). The main focus of the task-oriented tests is on functionality. The working place of the evaluator usually constitutes the testing environment. This type of test can be carried out at any stage of the product life cycle, including the process of software development. These tests are relatively inexpensive as no investments apart from the software tested and hardware are required (i.e. evaluator's technical environment).

**b) Menu-oriented testing**

In menu-oriented testing each feature of the program is tested in sequence. The evaluators may adopt either black box or glass box techniques. Since each individual function is checked, it is a very detailed evaluation. Menu-oriented testing can also take place at any

stage of the product life cycle. This technique requires a good testing staff able to develop *ad hoc* metrics and data.

### **c) Benchmark testing**

Benchmark testing examines the performance of the systems. This methodology allows for the evaluation of either the performance of individual functions, system modules or the system as a whole. “A benchmark test is a measurement of system performance which cannot be affected by variables resulting from human involvement.” (EAGLES 1995:34). Typically a benchmark test involves a checklist of quality characteristics, the benchmark, measurement technique and results.

### **2.2.3. FEATURE INSPECTION**

Feature checklists enable the evaluator to describe technical characteristics of a piece of software in a detailed manner. Checklisting is designed so that one could compare a number of systems of the same type. Feature checklists are compiled bearing in mind the individual features of the systems tested. Once the checklist is compiled, the systems under testing are inspected for the presence of the features listed. Therefore, this methodology helps indicate the differences between similar tools, and, as such, provides assistance in the selection of tools suitable for particular tasks and users from among the numerous products available on the market.

## **3. EVALUATION PROCEDURE USED IN THIS THESIS**

### **3.1. THE GOAL OF EVALUATION**

We are now ready to get acquainted with the evaluation methodology used in this thesis. In order to apply a proper evaluation procedure, we need to specify the goal of evaluation (Balkan, Meijer *et al.* 1994:3).

According to the Argos Company White Paper: ‘The overwhelming majority of translation companies in Poland are run out of the owners’ home and do not own legal software or know how to use TM tools.’ (Argos 2002). This paper further implies that the Polish translation market is in a desperate need of introducing new technologies to facilitate the process of translation and improve the quality of translation services. The same opinion is voiced by Feder throughout his unpublished doctoral thesis (Feder 2001).

In order to satisfy this increasing need and make the domestic translation market competitive worldwide, Polish translators and translator trainers should become aware of the existence and benefits of CAT tools and be able to select the ones tailored to their specific needs. Consequently, they need to be equipped with an inexpensive and transparent CAT software evaluation methodology. Despite the fact that different CAT tools have been in use for a few decades, their evaluation methodologies are still unsatisfactory and usually cannot be successfully applied in small scale projects.

The specific conditions of the Polish translation market make such an enterprise even more difficult. Scenario testing, for instance, is far too expensive to be applicable in our fragmented market where it is almost impossible to gather a representative sample of users for a field test or find a laboratory equipped with all the necessary software and hardware to conduct laboratory tests. The same is true for systematic testing which should be conducted on the fully functional program versions instead of free demo versions<sup>17</sup> available from the Internet. Moreover, systematic testing requires expert knowledge of software technology or excellent command of the tools tested. The above conditions *per se* exclude the application of this methodology in this project. As we may expect, few professional software engineers in Poland are interested in foreign terminology management tools<sup>18</sup>, and the potential users mostly have little or no experience of working with such tools whatsoever. Additionally, the tools in question are a costly investment, even for large translation agencies.

The comparative evaluation conducted in this project is supposed to deliver a comprehensive and unbiased picture of terminology management tools. To achieve it within the very limited scope of a master's thesis, the author decided to apply a feature inspection methodology. The readers will find below a comprehensive list of questions concerning various aspects of the applications tested, with metrics specified for each question. The evaluation criteria presented in the GTW Report<sup>19</sup> (Mayer *et al.* 1996) constitute the backbone of the checklist. The criteria created by the GTW are supplemented with the detailed questions put forward by Feder in the shortlist of evaluation criteria for terminology management systems (Feder 2001:342). Also the guidelines outlined in the EAGLES report (EAGLES 1995; EAGLES 1999) were taken into account in the process of checklist compilation along with recommendations of the POINTER report (POINTER 1996). Finally,

---

<sup>17</sup> Demo versions usually have some limitations e.g. the number of termbase records or translation units is limited to 100.

<sup>18</sup> There are attempts at creating Polish CAT tools, e.g. T4Office by the Poznań-based DomData company.

<sup>19</sup> Criteria for the Evaluation of Terminology Management Software, 1996. Gesellschaft für Terminologie und Wissenstransfer e.V. (Association for Terminology and Knowledge Transfer).

the author expanded the checklist by adding some new categories of questions in order to account for the new developments in the area of terminology management that were not present at the time the GTW report was being drawn up, and were treated as futuristic in Feder's thesis (Feder 2001).

Since the Polish translation market is characterized by a large number of small translation agencies – led by a single translator – it is most appropriate to create a checklist that will be able to help a freelance translator make an objective choice. The evaluation methodology applied in this project leaves the final choice to the readers and no preferences are expressed or suggested. The author should like to emphasize that every effort was made to compile an objective and comprehensive list of questions which does not put any of the tools tested in a favored position. The evaluation criteria can be applied to many tools of the same type and as such have a certain level of universality. Moreover, they can be easily expanded to encompass still new developments or criteria. Hence, they may constitute a good resource for those who need to make an informed choice of a terminology management system they plan to purchase. Furthermore, the transparent structure of the present feature checklist should not deter translators who have little or no knowledge of the tools in question. Finally, the present checklist contains certain criteria which are much of a 'wishlist' type, and can be viewed as a suggestion for CAT software developers of what possible features could be added to the new versions of terminology management tools. Also they can be treated as a background for more scientific discussion on how to evaluate these applications. In conclusion, the evaluation methodology suggested here, is an attempt not only at providing an objective evaluation procedure applicable in small-scale projects, but also at taking a stand in the on-going debate on terminology management software testing methodologies.

### **3.2. HOW THE GOAL IS ACHIEVED**

Having defined the goal of the present evaluation we now need to specify the way in which it is to be achieved. Since we aim at a comprehensive and detailed study, we need to inspect both technical and linguistic features of the applications under testing.

While evaluating the responses of the systems tested to different linguistic phenomena, we adopt a black box approach. However, we need to account for the fact that terminology management systems are usually language-independent, i.e. by supporting a large number of different languages, the tools have a limited linguistic competence in each

of the languages supported. Unlike in the evaluation of translation memory modules, where such linguistic aspects as segmentation rules, syntactic analysis, etc. constitute a large testing area, in terminology management systems the testing of linguistic aspects is usually limited to retrieval accuracy, spelling error recognition, recognition of compound terms, and the like.

However, if we want to obtain a detailed picture of terminology management tools, we need to account for certain technical issues such as software and hardware requirements – which often constitute critical factors for small translation agencies run on a single PC. Thus, we enter the sphere of glass box approach to software testing. The translators obviously do not need to know the details of the programs' architecture, or the source code for that matter, in order to use the applications as intended by the manufacturer. Still, some basic knowledge of the system's underlying architecture may help use the system more effectively. Moreover, such knowledge may come in extremely useful in troubleshooting e.g. when problems occur and no technical support is immediately available.

Therefore, we must not limit ourselves to either black or glass box testing methodology, but adopt a mixed approach instead.

Bearing in mind that terminology management systems are subordinate to human linguistic activity (Feder 2001:61) and have a very limited linguistic competence of their own, we should expect the majority of evaluation criteria to be of technical nature. However, the linguistic performance of the applications scrutinized in this project is also subject to analysis.

As has already been stated, the evaluation is conducted on the basis of a feature checklist compiled from a number of sources. The information reproduced in the test report, presented in chapter IV of this thesis, comes from a number of sources as well. The terminology management systems' linguistic performance is tested on the basis of a number of practical tasks. Testing the linguistic performance of a terminology management system requires specific linguistic input. Since using a natural text for the purposes of a small scale evaluation is highly unproductive i.e. a large quantity of textual data may include very few linguistic phenomena against which we intend to test the systems; it is most sensible to apply an artificial test suite. For the purposes of the present evaluation a number of small databases – each containing at least 20 records<sup>20</sup> – was created and later applied on an artificial test suite designed according to the guidelines laid down in the report titled: *Test Suite Design* –

---

<sup>20</sup> This evaluation is not supposed to test the systems' robustness but retrieval accuracy; therefore, there is no need to build extensive databases.

*Guidelines and Methodology*. (Balkan&Meijer *et al.* 1994:3). The readers will find enclosed a test suite (Appendix II) and a glossary version of the databases used in the evaluation procedure (Appendix I).

While the linguistic aspects of the applications under evaluation were tested on the basis of practical tasks, not all the technical criteria could be empirically verified within the limited scope of this project, e.g. the support of different operating systems is impossible to test on a single PC with a single operating system installed on it. In such cases the author had to rely on the product documentation to provide required data. Also the commercial aspects of the products under evaluation could only be reported on the basis of information made available to the public. It was either obtained through direct enquiries to the distributors and manufacturers or from their official homepages. Whenever the information provided in the report sheet could not be empirically verified, this fact is indicated by an asterisk. Since the evaluation conducted within this project is meant to be objective, the author decided to limit certain criteria which involve subjective user judgment, e.g. user-friendliness of the interface.

Finally, the author should like to emphasize that the evaluation was not designed to measure the performance of the system under extreme conditions. Therefore, no tasks were designed in order to confirm the data stated in the documentation, e.g. what the maximum number of terminological entries that can be entered in a single database is, or whether the number of fields within an entry is really unlimited. Bearing in mind the small size of the databases built for the purposes of this evaluation, we should also expect the numeric values concerning the number of records returned in hitlists to be low.

### **3.3. FEATURE CHECKLIST USED IN THE EVALUATION PROCEDURE**

Following the theoretical introduction of the NLP software testing methodologies and the description of methodology applied in this thesis, we will find below the feature checklist. As has already been mentioned, it is presented in the form of questions. For the sake of convenience, metrics are specified below the questions they refer to.

#### **3.3.1. TECHNICAL DESCRIPTION**

##### **3.3.1.1. HARDWARE REQUIREMENTS**

3.3.1.1.1. What type of platform is required/ recommended?



**Metrics:** descriptions

3.3.1.1.2. What type of microprocessor is required?

**Metrics:** descriptions

3.3.1.1.3. What is the minimum RAM size required?

**Metrics:** numeric values MB/GB

3.3.1.1.4. What is the recommended size?

**Metrics:** numeric values MB/GB

3.3.1.1.5. What HD space is required?

**Metrics:** numeric values MB/GB

3.3.1.1.6. What HD space is recommended?

**Metrics:** numeric values MB/GB

3.3.1.1.7. What is the required graphics standard? i.e. what type of graphics card is required?

**Metrics:** descriptions

3.3.1.1.8. What are the required/advocated peripheral devices (e.g. printer, mouse / trackball / touch pad, monitor, CD-ROM, modem, network card etc)?

**Metrics:** list

### **3.3.1.2. SOFTWARE REQUIREMENTS**

3.3.1.2.1. What operating systems does the tool support?

**Metrics:** list

3.3.1.2.2. Is the tool network/multi-user enabled?

**Metrics:** yes/no

3.3.1.2.3. Is the tool equipped with a mechanism enabling multi-tasking/ quasi-multitasking?

**Metrics:** yes/no

3.3.1.2.4. Is there any other software required to run the advanced functions of the tool?

**Metrics:** yes/no; list

### **3.3.2. COMPATIBILITY**

3.3.2.1. Are different versions of the same tool fully compatible?

**Metrics:** yes / no; descriptions

3.3.2.2. If not, how can the files created in the older versions be used by the new versions?

**Metrics:** descriptions

3.3.2.3. Can a user exchange not only data but also profiles, filters and settings?

**Metrics:** yes / no

3.3.2.4. How can a tool be extended into a new version? By means of upgrades? Purchasing a new version?

**Metrics:** descriptions

### **3.3.3. USER INTERFACE**

#### **3.3.3.1. INSTALLATION PROCEDURE**

3.3.3.1.1. What is the installation routine?

**Metrics:** description

#### **3.3.3.2. TYPE OF USER INTERFACE**

3.3.3.2.1. What is the type of user interface?

**Metrics:** description

3.3.3.2.2. How many primitive actions need to be performed in order to open the interface?

**Metrics:** numeric value; description

#### **3.3.3.3. INTERFACE LANGUAGES**

3.3.3.3.1. What dialog languages are available?

**Metrics:** list

3.3.3.3.2. When is the dialog language selected?

**Metrics:** description

3.3.3.3.3. How does one switch these languages?

**Metrics:** description

### **3.3.3.4. PRODUCT DOCUMENTATION, TRAINING AND USER HELP**

3.3.3.4.1. What forms of help are available to the users (e.g. manual/ online help/ tutorial/ technical support on-site/ wizards)?

**Metrics:** list

3.3.3.4.2. In what languages are these forms of help available?

**Metrics:** list

3.3.3.4.3. Is proper documentation supplied alongside the product (e.g. user manual, demos, workbooks, tutorials, sample files / DBs, online help, wizards, etc.)?

**Metrics:** yes / no, list

3.3.3.4.4. Are the materials in question supplied in a language that is understood by the user?

**Metrics:** yes / no

3.3.3.4.5. Does the documentation also cover troubleshooting?

**Metrics:** yes / no

3.3.3.4.6. Is the information on the internal workings of a program made available?

**Metrics:** yes / no

3.3.3.4.7. Are there any other forms of obtaining technical support and consultancy (e.g. user groups, mailing lists, newsletters, etc.)?

**Metrics:** yes / no; descriptions

### **3.3.3.5. USER INTERFACE ELEMENTS**

3.3.3.5.1. Is the communication implemented by use of...

a. typed commands?

**Metrics:** yes / no

b. function keys?

**Metrics:** yes / no

c. traditional menus?

**Metrics:** yes / no

d. pull down and pop-up menus?

**Metrics:** yes / no

e. dialog boxes?

**Metrics:** yes / no

f. icons?

**Metrics:** yes / no

g. clickable buttons?

**Metrics:** yes / no

3.3.3.5.1.8. Does the interface design require the use of a mouse/trackball?

**Metrics:** yes / no

3.3.3.5.1.9. Are any keyboard shortcuts (hotkeys) available?

**Metrics:** yes / no

3.3.3.5.1.10. Is it possible to manipulate taskbars, menus, toolbars, buttons, etc. (hide, move, resize, docked vs. floating bars)?

**Metrics:** yes / no

### **3.3.3.6. ON SCREEN DISPLAY**

3.3.3.6.1. Is the on screen display user-definable?

**Metrics:** yes / no; descriptions

3.3.3.6.2. Is the information displayed in a WYSIWYG manner?

**Metrics:** yes / no; descriptions

3.3.3.6.3. Are there any default display layouts that suit the needs of various users including special groups of users (e.g. translators, terminologists, writers, editors, etc.)?

**Metrics:** yes / no; descriptions

3.3.3.6.4. Are the settings selected visible to the user?

**Metrics:** yes / no; descriptions

### **3.3.4. TERMINOLOGICAL ASPECTS**

#### **3.3.4.1. DATA MANAGEMENT**

3.3.4.1.1. What are the languages supported by the tool?

**Metrics:** list

3.3.4.1.2. Are all the languages available both as the source and target languages?

**Metrics:** yes / no; list differences

3.3.4.1.3. Are language varieties supported by the tool?

**Metrics:** yes / no; list

3.3.4.1.4. Are bi-directional and DBCS languages supported both as SL and TL?

**Metrics:** yes / no; list differences

3.3.4.1.5. What is the underlying data model of the database (flat, relational, object-oriented, semantic network)?

**Metrics:** descriptions

3.3.4.1.6. What types of data can be inserted into the entry? (textual, graphic, multimedia, etc.)

**Metrics:** list

3.3.4.1.7. What file types are supported by the tool?

**Metrics:** list

3.3.4.1.8. What is the maximum number of data collections/databases?

**Metrics:** numeric values

3.3.4.1.9. Can more than one database be used at a time?

**Metrics:** yes / no

3.3.4.1.10. What is the maximum number of data collections/databases that can be consulted at a time?

**Metrics:** numeric values

3.3.4.1.11. Is it possible to define the lookup order?

**Metrics:** yes / no

3.3.4.1.12. What is the maximum number of languages per databank?

**Metrics:** numeric values

3.3.4.1.13. Can a mono-/bilingual subset be extracted from a multi-lingual database?

**Metrics:** yes / no

3.3.4.1.14. Does the tool perform sorting according to the language?

**Metrics:** yes / no

3.3.4.1.15. Can the directions of the database be changed?

**Metrics:** yes/no

3.3.4.1.16. How many steps are required to do it?

**Metrics:** numeric values; description

3.3.4.1.17. Are the following project management functions supported by a tool:

- a. statistical data (DB size, no. of units in a DB, word count, reusability, no. of translated / untranslated words, no. of reused segments / terms);

**Metrics:** yes / no; list

- b. quality assurance (project status, terminological consistency, spelling, proper application of resources);

**Metrics:** yes / no; list

- c. data security (passwords, access rights, locking, read-only vs. write access, functionality blocking, protected screen areas, data compression and encryption, max. no. of users, etc.);

**Metrics:** yes / no; list

- d. file and folder structure (automatic vs. manual, naming and re-naming, long filename support, backup files, inclusive export / import of all associated files, etc.);

**Metrics:** yes / no; list

- e. messages (consistency check, other)?

**Metrics:** yes / no; list

3.3.4.1.18. Are these functions built into the tool or does the tool rely on external software to provide these options (e.g. WP)?

**Metrics:** yes / no; descriptions

3.3.4.1.19. Can these features be suppressed?

**Metrics:** yes / no; descriptions

### **3.3.4.2. ENTRY MODEL AND STRUCTURE**

3.3.4.2.1. Is the entry structure free/quasi-free/fixed?

**Metrics:** descriptions

3.3.4.2.2. Is it possible to add fields (in the case of quasi-free record structure) or are there any freely definable fields available (fixed and quasi-free structures)?

**Metrics:** yes / no; descriptions

3.3.4.2.3. What are the field names and field naming conventions?

**Metrics:** descriptions

3.3.4.2.4. What data categories are set in a given tool if it has a fixed or quasi-free record structure?

**Metrics:** list

3.3.4.2.5. What data categories are required by a user (in the case of a free record structure)?

**Metrics:** user specifications

3.3.4.2.6. Are there any standard fields offered (free record structure)?

**Metrics:** yes / no; list

3.3.4.2.7. What is the maximum field length?

**Metrics:** numeric values

3.3.4.2.8. Are there any standard record templates? How many?

**Metrics:** yes / no; numeric values

3.3.4.2.9. Can they be modified and then saved as different templates?

**Metrics:** yes / no; descriptions

3.3.4.2.10. Are there any standard field attributes?

**Metrics:** yes / no; list

3.3.4.2.11. Are certain data categories filled in automatically?

**Metrics:** yes / no; list

3.3.4.2.12. Are there any fields for which entry is mandatory?

**Metrics:** yes / no; list

3.3.4.2.13. Is the total number of fields limited?

**Metrics:** yes / no; numeric values

3.3.4.2.14. What is the minimum no. of fields that have to be filled in to create a valid record? **Metrics:** numeric values

3.3.4.2.15. How many primitive actions need to be performed in order to create the simplest entry i.e. containing only TL and SL equivalents?

**Metrics:** numeric values

3.3.4.2.16. Are there any special words / word classes / characters that cannot constitute a valid DB entry?

**Metrics:** yes / no; list

3.3.4.2.17. Is it possible to change field definition (e.g. name, length, position in the record)?

**Metrics:** yes / no; descriptions

3.3.4.2.18. Can data be grouped within an entry?

**Metrics:** yes / no

3.3.4.2.19. Is categorization achieved automatically/manually?

**Metrics:** descriptions

3.3.4.2.20. Is intentional repetition of (some) data categories possible?

**Metrics:** yes / no; descriptions

3.3.4.2.21. Is it possible to specify / restrict the type of data to be entered into a given field (e.g. alphanumeric vs. numeric)?

**Metrics:** yes / no; descriptions

3.3.4.2.22. To what fields do these limitations apply?

**Metrics:** list

3.3.4.2.23. Is it possible to create cross-references among records?

**Metrics:** yes / no

3.3.4.2.24. Are these created automatically or manually?

**Metrics:** descriptions

3.3.4.2.25. Are the cross-references created via special fields or from within any field?

**Metrics:** descriptions

3.3.4.2.26. Is it possible to create links to external resources?

**Metrics:** yes / no

3.3.4.2.27. Is the DB / record structure mono-, bi- or multilingual?

**Metrics:** descriptions

3.3.4.2.28. Is the DB/record structure term or concept oriented?



**Metrics:** descriptions

3.3.4.2.29. What is the maximum no. of languages a record can hold?

**Metrics:** numeric values

3.3.4.2.30. Is it possible to customize the display to show only two, three, etc. languages of the total no. of languages covered by the database?

**Metrics:** yes / no; descriptions

3.3.4.2.31. Is it possible to define constant values for certain fields to be applied uniformly throughout the entire database?

**Metrics:** yes / no; descriptions

3.3.4.2.32. Is there a limit to the record size (no. of fields, their lengths, size of record in KB, no. of pages, etc.)?

**Metrics:** yes / no; descriptions

3.3.4.2.33. Are there any different record types?

**Metrics:** yes / no; descriptions

3.3.4.2.34. What is the total number of records per database / dictionary?

**Metrics:** numeric values

3.3.4.2.35. Does the tool support the following administrative data categories:

- a. project name;
- b. subset name;
- c. language pair and direction;
- d. language variant;
- e. translator / terminologist;
- f. project manager / system administrator;
- g. creation date;
- h. change / update date;
- i. match level;
- j. match source;
- k. translation status;
- l. subject domain;
- m. client;
- n. associated resources;
- o. copyright information;
- p. usage counter;

- q. DB usability / validity restrictions;
- r. other remarks?

**Metrics:** list

### **3.3.5. RETRIEVAL OF INFORMATION**

#### **3.3.5.1. ACCESS TO INFORMATION**

3.3.5.1.1. Which of the search options are offered by the tool?

- a. Exact match?

**Metrics:** yes / no

- b. Partial match?

**Metrics:** yes / no

- c. Truncation (left/right/center)?

**Metrics:** yes / no

- d. Wild card?

**Metrics:** yes / no

- e. Free text?

**Metrics:** yes / no

- f. Fuzzy search?

**Metrics:** yes / no

- g. Via record / translation unit number?

**Metrics:** yes / no

- h. KWIC?

**Metrics:** yes / no

- i. Boolean operator

**Metrics:** yes / no

- j. Relational operator?

**Metrics:** yes / no

- k. Morphological?

**Metrics:** yes / no

- l. By synonym; cross-reference/ internal/external link?

**Metrics:** yes / no

- m. Proximity?

**Metrics:** yes / no

n. Meaning?

**Metrics:** yes / no

o. Subject area?

**Metrics:** yes / no

p. Global?

**Metrics:** yes / no

q. Restricted (filtered)?

**Metrics:** yes / no

r. By segments containing a term or phrase?

**Metrics:** yes / no

s. Capital vs. small letter?

**Metrics:** yes / no

t. Punctuation and spacing variation?

**Metrics:** yes / no

u. By mark-up / formatting features?

**Metrics:** yes / no

v. Search history?

**Metrics:** yes / no

w. Search log?

**Metrics:** yes / no

x. Browsing (alphabetical, chronological, conceptual, etc.)?

**Metrics:** yes / no

y. Access via any data category?

**Metrics:** yes / no

z. Query language (e.g. SQL)?

**Metrics:** yes / no

3.3.5.1.2. Can search criteria be combined by Boolean or relational operators?

**Metrics:** yes / no

3.3.5.1.3. Is global search and replace possible?

**Metrics:** yes / no

3.3.5.1.4. Do the search and replace options work equally well for both languages?

**Metrics:** yes / no; descriptions

### **3.3.5.2. SYSTEM'S RESPONSES**

3.3.5.2.1. What are the tool's responses if search criteria are not met?

a. hitlist of near matches

**Metrics:** yes / no

b. 'not found'

**Metrics:** yes / no

c. logging term not found

**Metrics:** yes / no

d. history of search

**Metrics:** yes / no

3.3.5.2.2. If the hitlist contains fuzzy matches, is this fact indicated in any way?

**Metrics:** descriptions

3.3.5.2.3. Is the tool able to recognize a misspelled term?

**Metrics:** yes / no

3.3.5.2.4. How does a tool respond to a compound term when not found in the database?

**Metrics:** description

3.3.5.2.5. Does the tool return the base form for inflected forms?

**Metrics:** yes/no

3.3.5.2.6. Does the tool recognize spelling variants (e.g. color vs. colour)?

**Metrics:** yes/no

3.3.5.2.7. Does the tool recognize differences in compound spelling (hyphenated vs. non-hyphenated variants)?

**Metrics:** yes / no

3.3.5.2.8. Does the tool recognize the part of speech of a term?

**Metrics:** yes / no

### **3.3.5.3. SECURITY OF INFORMATION**

3.3.5.3.1. Can access rights to the database be defined?

**Metrics:** yes / no

### **3.3.6. INPUT OF INFORMATION**

#### **3.3.6.1. EDITING**

3.3.6.1.1. Is it possible to format the characters?

**Metrics:** yes / no

3.3.6.1.2. Is it possible to format paragraphs?

**Metrics:** yes / no

3.3.6.1.3. Is it possible to edit entries through...

a. Copy?

**Metrics:** yes / no

b. Paste?

**Metrics:** yes / no

c. Drag and drop?

**Metrics:** yes / no

d. Search and replace?

**Metrics:** yes / no

e. Delete?

**Metrics:** yes / no

f. Redo?

**Metrics:** yes / no

g. Undo?

**Metrics:** yes / no

h. Insert?

**Metrics:** yes / no

i. Changing the layout?

**Metrics:** yes / no

3.3.6.1.4. Can the existing data be modified as well?

**Metrics:** yes / no

3.3.6.1.5. Does the tool enable the user to perform editing tasks using search and replace options?

**Metrics:** yes / no; descriptions

### **3.3.6.2. TERMINOLOGY EXTRACTION**

3.3.6.2.1. Does the tool support the function of terminology extraction?

**Metrics:** yes / no

3.3.6.2.2. If not, does the manufacturer offer another tool/module which does?

**Metrics:** yes / no

3.3.6.2.3. What are the languages available for terminology extraction?

**Metrics:** list

3.3.6.2.4. Does the tool extract single terms/compound terms/phraseology?

**Metrics:** descriptions

3.3.6.2.5. What formats are supported for term extraction? RTF/SGML?

**Metrics:** list

3.3.6.2.6. Is it possible to extract terminology from a bilingual/multilingual corpus?

**Metrics:** yes / no; descriptions

3.3.6.2.7. If so, does the tool perform alignment?

**Metrics:** yes / no

### **3.3.6.3. VALIDATION/CONTROL**

3.3.6.3.1. Is it possible to define the rules for data import?

**Metrics:** yes / no

3.3.6.3.2. Does the tool offer control of data input?

**Metrics:** yes / no; descriptions

3.3.6.3.3. Does the tool perform spellchecking?

**Metrics:** yes / no

3.3.6.3.4. Does the tool alert about duplicate entries during import/manual input/ automatic input of terminological data?

**Metrics:** yes / no

3.3.6.3.5. Does the tool signal the omission of obligatory data categories?

**Metrics:** yes / no

### **3.3.7. EXCHANGE OF INFORMATION**

#### **3.3.7.1. PRINTING**

3.3.7.1.1. Does the tool support printing directly?

**Metrics:** yes / no

3.3.7.1.2. Is there a list of printers supported by the tool?

**Metrics:** yes / no; list

3.3.7.1.3. Is it possible to select only certain data for printing?

**Metrics:** yes / no

3.3.7.1.4. Is it possible to define the view of data for printing?

**Metrics:** yes / no

### **3.3.7.2. IMPORT/EXPORT**

3.3.7.2.1. Is import/export of data possible?

**Metrics:** yes / no

3.3.7.2.2. Is it possible to define the selection criteria for export/import?

**Metrics:** yes / no

3.3.7.2.3. Is it possible to define the views for export/import?

**Metrics:** yes / no

3.3.7.2.4. Does the tool support any of the major exchange standards?

**Metrics:** yes / no; list

3.3.7.2.5 Are there any other exchange formats supported by a given tool? E.g. does the tool support native formats of other tools of the same type?

**Metrics:** yes / no; list

### **3.3.8. INTERACTION WITH OTHER APPLICATIONS**

#### **3.3.8.1. INTERACTION WITH WORD PROCESSORS**

3.3.8.1.1. Can a termbase be accessed from a word processor?

**Metrics:** yes / no

3.3.8.1.2. Is the word processor window visible when accessing the database?

**Metrics:** yes / no

3.3.8.1.3. Is it possible to copy from the database into the WP?

**Metrics:** yes / no

3.3.8.1.4. Is it possible to copy from the WP into the database?

**Metrics:** yes / no

3.3.8.1.5. Is the copying direct/through a buffer?

**Metrics:** descriptions

3.3.8.1.6. Does the tool recognize the terms automatically?

**Metrics:** yes / no



3.3.8.1.7. Does the tool replace terms automatically?

**Metrics:** yes / no; descriptions

3.3.8.1.8. Can new entries be added?

**Metrics:** yes / no

3.3.8.1.9. Are there any minimal / rapid entry options available?

**Metrics:** yes / no

3.3.8.1.10. Can the existing entries be modified?

**Metrics:** yes / no

3.3.8.1.11. When combined with a WP, is the terminology lookup automatic, manual or both?

**Metrics:** descriptions

3.3.8.1.12. In the case of manual terminology lookup, how does one access the TMS?

**Metrics:** descriptions

3.3.8.1.13. Is the terminology transfer automatic, manual or both?

**Metrics:** descriptions

3.3.8.1.14. If manual, how is it effected?

**Metrics:** descriptions

3.3.8.1.15. Is it possible to see the whole record or only an abbreviated form?

**Metrics:** descriptions

3.3.8.1.16. How does one access the full display of a record?

**Metrics:** descriptions

3.3.8.1.17. a. Is it possible to analyze an SL text to extract found, unfound and forbidden terms?

**Metrics:** yes / no

3.3.8.1.17. b. Is this analysis performed against one dictionary / set of dictionaries / all dictionaries?

**Metrics:** descriptions

3.3.8.1.18. Can a user define that?

**Metrics:** yes / no

3.3.8.1.19. If more than one database is used, are the search results displayed in several windows?

**Metrics:** yes / no

3.3.8.1.20. Is it possible to save / mark / insert the whole segment containing a given term?

**Metrics:** yes / no; descriptions

3.3.8.1.21. Can the same database be open in several windows?

**Metrics:** yes / no

3.3.8.1.22. Is it possible to create a log file recording all unsuccessful terminological queries for subsequent addition to a dictionary?

**Metrics:** yes / no

### **3.3.8.2. INTERACTION WITH TRANSLATION MEMORY**

3.3.8.2.1. Can the termbase be accessed from a translation memory module?

**Metrics:** yes / no

3.3.8.2.2. Does the tool recognize the terms automatically?

**Metrics:** yes / no

3.3.8.2.3. Does the tool replace terms automatically?

**Metrics:** yes / no

3.3.8.2.4. Can new entries be added while working in a TM mode?

**Metrics:** yes / no

3.3.8.2.5. How is this effected?

**Metrics:** descriptions

3.3.8.2.6. Can the existing entries be modified?

**Metrics:** yes / no

3.3.8.2.7. Can a term be added directly from a TM window?

**Metrics:** yes / no

3.3.8.2.8. Are there any minimal / rapid entry options available?

**Metrics:** yes / no; descriptions

3.3.8.2.9. Is it possible to insert a list of terms or just one term at a time?

**Metrics:** descriptions

3.3.8.2.10. Is it possible to analyze an SL text to extract found, unfound and forbidden terms?

**Metrics:** yes / no

3.3.8.2.11. Is this analysis performed against one dictionary / set of dictionaries / all dictionaries?

**Metrics:** descriptions

3.3.8.2.12. Can the user define that?

**Metrics:** yes / no

3.3.8.2.13. If more than one database is used, are the search results displayed in several windows?

**Metrics:** yes / no

3.3.8.2.14. Is it possible to save / mark / insert the whole segment containing a given term?

**Metrics:** yes / no; descriptions

3.3.8.2.15. Can the same database be opened in several windows?

**Metrics:** yes / no

3.3.8.2.16. Is it possible to create a log file recording all unsuccessful terminological queries for subsequent addition to a dictionary?

**Metrics:** yes / no; descriptions

### **3.3.8.3. INTERACTION WITH OTHER TOOLS**

3.3.8.3.1. Can the tool be combined with a MT system?

**Metrics:** yes / no; descriptions

3.3.8.3.2. Can the tool be combined with term extraction tools?

**Metrics:** yes / no; descriptions

3.3.8.3.3. Can the tool be combined with alignment tools?

**Metrics:** yes / no; descriptions

3.3.8.3.4. Can the tool be combined with concordancers?

**Metrics:** yes / no; descriptions

3.3.8.3.5. Can the tool be combined with word frequency programs?

**Metrics:** yes / no; descriptions

3.3.8.3.6 Can the tool be combined with speech/voice recognition software?

**Metrics:** yes / no; descriptions

### **3.3.9. FONTS AND CHARACTER SETS**

3.3.9.1. What fonts and character sets are available?

**Metrics:** list

3.3.9.2. Does a tool support all special characters and fonts needed by a given user?

**Metrics:** yes / no; user-defined specifications

3.3.9.3. Can these special characters and fonts be transferred between various application windows?

**Metrics:** yes / no; descriptions

3.3.9.4. Are these special character sets and fonts supported for other tool functionalities (e.g. segmentation, alignment, sorting, etc.)?

**Metrics:** yes / no; descriptions

3.3.9.5. What standard encoding systems are supported by the tool?

**Metrics:** list

### **3.3.10. MAINTENANCE OPERATIONS**

3.3.10.1. Is it necessary to save the database after each update?

**Metrics:** yes / no

3.3.10.2. Is it necessary to update the dictionary/database index after each update?

**Metrics:** yes / no

3.3.10.3. Is it possible to compress the files using the tool?

**Metrics:** yes / no

3.3.10.4. Is it possible to recover/repair a corrupted database?

**Metrics:** yes / no

3.3.10.5. Are backup files generated automatically?

**Metrics:** yes / no

### **3.11. COMMERCIAL ASPECTS**

3.3.11.1. Who is the manufacturer of the tool?

**Metrics:** list

3.3.11.2. Who is the distributor of the tool?

**Metrics:** list

3.3.11.3. What is the price of a single-user license?

**Metrics:** specify

3.3.11.4. What forms of updating the software are available?

**Metrics:** descriptions

3.3.11.5. Is the tool directly available on the domestic (Polish) market?

**Metrics:** yes / no

3.3.11.6. Are technical support services offered directly on the domestic (Polish) market?

**Metrics:** yes / no

3.3.11.7. What is the number of registered users of the tool?

**Metrics:** numeric value

3.3.11.8. What is the date of the first release?

**Metrics:** date

3.3.11.9. What is the date of the last update?

**Metrics:** date

3.3.11.10. Are there any renowned users of the tool?

**Metrics:** list

## **4. CONCLUSION**

In this chapter the readers were given the opportunity to get acquainted with the basic classification of NLP software evaluation methodologies. Next, there was a detailed presentation of the evaluation methodology applied in this project. The readers could follow the feature checklist compiled on the basis of some of the most significant pieces of writing

concerning CAT software evaluation in general and the evaluation of terminology management systems in particular. The testing methodology described here is supposed to provide enough information to facilitate a well-informed and unbiased choice of a terminology management system suited for the individual needs of translators, especially translators working on the Polish market. The present chapter coupled with the introductory chapters I and II provides us with all the necessary background knowledge to analyze the test results presented in chapter IV.