



Integrated intelligent LEARNing environment for Reading and Writing

D4.4 – Content Classification Module



Document identifier	D4.4_Content_classification_module_final.docx
Date	2013-12-21
WP	WP4
Partners	EPIRUS, NTUA, DYSACT, LBUS
WP Lead Partner	NTUA
Document status	Final

Deliverable Number	D4.4
Deliverable Title	Content Classification Module
Deliverable version number	Final
Work package	WP4
Task	Task 4.4 Content classification
Nature of the deliverable	Report (R)
Dissemination level	Public (PU)
Date of Version	2013-12-21

Author(s)	Chris Litsas
Contributor(s)	Maria Mastropavlou, Dominik Lukes, Cantemir Mihiu
Reviewer(s)	Antonios Symvonis
Abstract	In this report we describe in detail the methods that are employed for text classification (with respect to the degree of appropriateness for a particular user, based on her profile). Text classification is supported for both the English and Greek languages through appropriately developed lexical analysis components.
Keywords	Text Classification; User Model; User Profile; Difficult Word; Word Score; Text Score

Document Status Sheet

Issue	Date	Comment	Author
v01	2013-12-04	Content classification methodology	Chris Litsas
v02	2013-12-11	Contribution on the resources and the libraries used by the system	Cantemir Mihiu
v03	2013-12-17	Contribution on the “generic text metrics”	Maria Mastropavlou, Dominik Lukes
v04	2013-12-19	General review and corrections on v03	Antonios Symvonis
v05	2013-12-21	Final version	Chris Litsas

Project information

Project acronym:	ILearnRW
Project full title:	Integrated Intelligent Learning Environment for Reading and Writing
Proposal/Contract no.:	318803

Project Officer: Krister Olson

Address:	L-2920 Luxembourg, Luxembourg
Phone:	+35 2430 134 332
E-mail:	kriste.olson@ec.europa.eu

Project Co-ordinator: Noel Duffy

Address:	Dolphin Computer Access Ltd. Technology House, Blackpole Estate West, Worcester, UK. WR3 8TJ
Phone:	+01 905 754 577
Fax:	+01 905 754 559
E-mail:	noel.duffy@yourdolphin.com

Table of Contents

1. INTRODUCTION.....	6
2. LINGUISTIC COMPLEXITY AND READING DIFFICULTY.....	7
2.1. WHAT IS LINGUISTIC COMPLEXITY?	7
2.2. LINGUISTIC COMPLEXITY AND TEXT COMPLEXITY.....	8
3. REVIEW OF COMMONLY USED READABILITY FORMULAS	10
3.1. READABILITY FORMULAS AVAILABLE FOR ENGLISH	10
3.1.1. <i>The Coleman-Liau Readability Formula (The Coleman-Liau Index)</i>	10
3.1.2. <i>The Automated Readability Index (ARI)</i>	10
3.1.3. <i>Fry Readability Formula (1965)</i>	11
3.1.4. <i>SMOG Readability Formula (1969)</i>	11
3.1.5. <i>FOG Index Formula (1952)</i>	11
3.1.6. <i>The Flesch Reading Ease Scale</i>	11
3.1.7. <i>Linsear Write Readability Formula</i>	12
3.2. READABILITY FORMULAS ADAPTED TO GREEK	12
4. METHODOLOGY	13
4.1. DESCRIPTION OF OUR APPROACH.....	13
4.2. USER MODELING	13
4.2.1. <i>Description</i>	13
4.2.2. <i>User Model Visualization</i>	16
4.3. CLASSIFICATION OF WORDS/TEXT	17
4.3.1. <i>Word Related Functions</i>	17
4.3.2. <i>Classification of Words</i>	17
5. TECHNIQUES	21
5.1. TOOLS WE USE	21
5.2. TEXT PROCESSING.....	22
5.2.1. <i>Text pre-processing</i>	22
5.2.2. <i>Check words against user's problems</i>	23
5.3. SERVER-SIDE COMPONENT	23
6. APPLICATIONS.....	24
6.1. ILEARNRW TEXT CLASSIFICATION TOOL	24
6.1.1. <i>User Profile</i>	24
6.1.2. <i>File Explorer</i>	25
6.1.3. <i>Word Metrics</i>	25
6.1.4. <i>Text Metrics</i>	26
7. CONCLUSIONS.....	28
REFERENCES.....	29

1. Introduction

A child that learns to read and/or write will practice with several pieces of text. However, not all text is appropriate to be used in the learning process (either for reading or for writing) of a particular child. The level of difficulty (or “degree of appropriateness”) of the text must be carefully considered. For a child without learning difficulties, the degree of appropriateness depends, among others, on the child’s age, the size of her vocabulary, the syntactical complexity of the text, etc. For children with learning difficulties, many more factors can be combined to decrease the degree of appropriateness of text and render it as unsuitable for a child. The child’s profile specified several error-types the child is likely to make. As a consequence, a text rich in words/structures that are sensitive to these error types is likely to cause more problems to the child during reading/writing. Text classification with respect to the degree of appropriateness for a particular user (based on her profile) will be widely used in order to search for appropriate content for a particular user. Text classification will be a major component of the on-line recourse bank that will be supported by iLearnRW.

Content classification and, consequently, profile parameterized text searching is a valuable feature for a user with a reading/writing disability. In exactly the same way we are able to sort our files based on their size or creation-time, she should be able to list them in decreasing order of suitability/appropriateness for her profile. The implementation of this component is a language dependent task and is supported for both the English and Greek languages.

The current document is organized as follows: Firstly, we describe the notions of *text complexity* and *reading difficulty* by presenting a brief review of the corresponding bibliography. In section 3, we list the most well-known readability tests, some of them have been incorporated and implemented for the purposes of the content classification module. A description of the methodology used for the development of the content classification module is presented in section 4. Descriptions of the *user model* as well as some basic definitions and metrics relevant to classification of words and texts are presented. In section 5, the basic techniques and the resources that we used in order to implement the module are listed. Section 6 presents a demo application that was implemented for the purposes of the first annual review of the project. We conclude in section 7.

2. Linguistic complexity and reading difficulty

This report aims to provide a description of the content classification module (CCM) that will be incorporated in the ILearnRW software. The design of the CCM aims to provide individualized teaching assistance to children with dyslexia by enabling a teacher or parent to classify texts with respect to the degree of appropriateness for a particular child, based on his/her profile, as well as to search for appropriate content for a particular child. The classification of texts will be made based on their readability, which is closely related to and even determined by the linguistic complexity of a text in the sense that the readability of a text increases as linguistic complexity decreases and vice versa. Therefore, linguistic complexity is a central notion when dealing with text classification.

2.1. What is linguistic complexity?

The first requirement in order to understand the notion of complexity is to have a working conceptualization of it. A central question then arises, one which involves the language-specific characteristics that are considered complex, that is, that render specific linguistic material complex or simple. The different linguistic domains are considered as complementary in nature, which means that complexity in one grammatical domain is often compensated by simplicity in another (Miestamo, 2009). Under this hypothesis, which is known as the *equi-complexity dogma* (Kusters 2003), all languages are characterized by equal levels of complexity. However, this hypothesis has been severely criticized so that linguistic complexity has been – and is still – receiving a large amount of research interest.

Defining linguistic complexity is currently one of the most hotly debated notions in linguistics. Finding a widely accepted definition of where complexity applies has been quite challenging within linguistic research, while methods of measuring levels of linguistic complexity has also been highly controversial. A number of different criteria have been proposed, such as the length of a linguistic expression (e.g. a clause or a sentence), ambiguity, etc. (Kusters 2008, McWhorter 2007).

In a first quantitative description of linguistic complexity, Blache (2011) identifies the types of constructions that are considered complex and thus difficult to process. He differentiates between *local complexity*, which refers to structural complexity, *difficulty*, which involves processing aspects and cognitive load, and *global complexity*, which refers to the language as a system rather than the complexity of a given realization. In a similar classification, Miestamo (2008) distinguishes between *global* and *local* linguistic complexity, referring to the complexity of a language or language variety and the complexity of a particular linguistic domain respectively. Of the two levels, local complexity is considered measurable and has drawn considerable attention in the literature. Local complexity therefore includes *phonological complexity* (e.g. size of phonemic inventory, incidence of marked phonemes,

phonotactic restrictions, maximum complexity of consonant clusters), *morphological complexity* (e.g. extent of allomorphy use and morphophonemic processes), *syntactic complexity* (e.g. level of clausal embedding and recursion), *semantic and lexical complexity* (e.g. extensive occurrence of homonymy and polysemy, type/token ratios), *pragmatic complexity* (e.g. degree of pragmatic inferencing) (see Szmrecsanyi & Kortmann 2012 for a review).

Various attempts to measure language complexity have led to the formulation of a number of specific notions of complexity: *absolute-quantitative complexity*, *redundancy-induced complexity* and *irregularity-induced complexity*. *Absolute-quantitative complexity* takes into account quantitative measures like the number of marked phonemes and the number of syntactic rules employed (Arends 2001). *Redundancy-induced complexity* involves linguistic elements that are active in a language but are synchronically non-transparent, that is, features that constitute remnants from earlier stages of development of a language but have no synchronic aetiology whatsoever. These may include ergativity, grammaticalized evidential marking, “dummy” verbs, syntactic asymmetries between main and dependent clauses, verb-second, and others (McWhorter 2001). *Irregularity-induced complexity* refers to the frequency of irregular, that is opaque or non-transparent, inflectional and derivational processes (McWhorter 2008).

2.2. Linguistic complexity and text complexity

The complexity or the degree of challenge of a particular text is the result of combinations and interactions of a variety of factors. These may include linguistic complexity factors, topic familiarity, word difficulty, sentence length, concreteness of ideas and concepts and others. In a description of text complexity, Lipson and Wixson (2003) define a number of factors that affect the readability of a text, which include the number of syllables in the words and the number of words in the sentences, while other linguistic characteristics, such as vocabulary and sentence structure, text organization and the amount of background knowledge that is required of readers are also often taken into account when determining the appropriateness of a text for a particular reader (Chall, Bissess, Conard, & Harris-Sharples 1996). In a more detailed account of the linguistic factors that affect text readability, Hess and Bigham (2004) define these factors as the following: word difficulty and sentence structure, text structure, discourse style (e.g. satire or humor), genre, background knowledge, degree of familiarity with text topic, level of reasoning required, organization and layout of text and text length.

During the 19th century, research on literary analysis and linguistics started developing qualitative tools aiming to analyze text complexity, focusing on the description of text features that impact on their readability. This research interest led to the formulation of readability formulas, which were first developed in the 1920's and constituted the outcome of the need to match reading materials with specific readers, as well as the need to select appropriate teaching materials for the classroom. Readability formulas were based on the

assumption that reading difficulty is determined by specific text features, which can be entered into an equation that will produce a numerical estimate of readability for a particular text. Each level of readability would then be mapped onto a specific educational level or age group, enabling thus the selection of appropriate reading material for an individual reader of a particular age or educational background.

In an extensive review of readability research, Klare (1984) described the four most commonly used readability formulas: the Flesch Reading Ease Index (Flesch, 1948), the Fry Index (Fry, 1968), the Dale-Chall Readability Formula (Chall & Dale, 1995), and the Flesch-Kincaid Grade Level (GL) Score (Kincaid, Fishburne, Rogers, & Chissom, 1975). These formulas test readability by employing two independent variables: syntactic and semantic complexity. Syntactic complexity is measured in terms of sentence length, while semantic difficulty is differently measured in the four approaches: three of them (Flesch, Flesch-Kincaid and Fry) take into account word length measured in number of syllables, while the Dale-Chall measure assesses semantic difficulty in terms of mean word frequency.

In 1988 Stenner et al. introduced the *Lexile Framework for Reading*, an alternative measure of text readability that is now very widely used in elementary and middle schools in the U.S. The Lexile Framework also makes use of two linguistic variables in assessing text difficulty: syntactic complexity in terms of sentence length, and semantic complexity in terms of word frequency, which is established through occurrence counts in a large corpus of texts that constitute representative reading materials for students from kindergarten through college. However, this method overlooks a number of significant linguistic factors that impact on syntactic complexity, such as embedding, recursion, locality, as well as factors that affect semantic complexity, such as polysemy, concreteness etc.

The readability formulas described so far all overlook important variables that determine the linguistic complexity of a text. These include discourse characteristics, density of information, inferential requirements, rhetorical structure, text genre, complexity of ideas etc. Additionally, reader-related variables are also overlooked, such as motivation, cultural background and general world knowledge. For these reasons, readability formulas have often been criticized and considered “overly simplistic” (Sawyer 1991) with regards to the complexity of what is being assessed, especially due to the fact that critical variables are not taken into consideration.

3. Review of commonly used Readability Formulas

Readability tests are designed to predict whether a particular text is appropriate for a particular reader, although they cannot measure the reader's comprehension abilities directly. Additionally, text features like the complexity of the ideas, cohesion and coherence cannot be evaluated. Today, a considerable number of readability measures are available, most of which measure characters per word, words per sentence, sentence and paragraph statistics. The most widely used readability formulas that are available for English are described in the following section, while those that have been adjusted to Greek are described in 3.2.

3.1. Readability formulas available for English

3.1.1. The Coleman-Liau Readability Formula (The Coleman-Liau Index)

The Coleman–Liau Readability Formula is designed to approximate the usability of a text. It provides word statistics based on numbers of characters rather than numbers of syllables. Its rationale is that instead of using syllable/word and sentence length indices, computerized assessments understand characters more easily and accurately than counting syllables and sentence length. The formula used by the Coleman-Liau index is given in (1) below:

$$(1) \text{CLI} = 0.0588L - 0.296S - 15.8$$

where L is the average number of letters per 100 words and S is the average number of sentences per 100 words.

3.1.2. The Automated Readability Index (ARI)

The Automated Readability Index (ARI) has been designed to assess the comprehensibility of a text. ARI results from ratios that represent word difficulty (number of letters per word) and sentence difficulty (number of words per sentence). The mathematical formula used by ARI is given in (2):

$$(2) 4.71 \left(\frac{\text{characters}}{\text{words}} \right) + 0.5 \left(\frac{\text{words}}{\text{sentences}} \right) - 21.43$$

where *characters* is the number of letters, numbers, and punctuation marks, *words* is the number of spaces and *sentences* is the number of sentences.

3.1.3. Fry Readability Formula (1965)

The Fry readability formula is often used for regulatory purposes, such as to ensure that publications have a level of readability that is accessible to a wider portion of the population. The tool plots a number of measures on a graph, presenting the mean number of sentences per one hundred words on the y-axis and the mean number of syllables per one hundred words on the x-axis. The intersection of the average number of sentences and the average number of syllables determines the reading level of the content.

3.1.4. SMOG Readability Formula (1969)

SMOG is a widely used readability measure, commonly used for checking health messages. The mathematical formula used is given in (3):

$$(3) \text{ grade} = 1.0430 \sqrt{\text{number of polysyllables} \times \frac{30}{\text{number of sentences}}} + 3.1291$$

3.1.5. FOG Index Formula (1952)

The Gunning Fog Index Readability Formula is considered a considerably accurate readability measure. The rationale behind the specific measures it employs is that short sentences written in Plain English achieve a better score than long sentences written in complicated language. The ideal score for readability with the Fog index is 7 or 8. Anything above 12 is too hard for most people to read. The formula used is given in (4):

$$(4) 0.4 \left[\left(\frac{\text{words}}{\text{sentences}} \right) + 100 \left(\frac{\text{complex words}}{\text{words}} \right) \right]$$

3.1.6. The Flesch Reading Ease Scale

The Flesch Reading Ease Scale is the most commonly used readability formula. It produces readability scores on a scale from 0 (very difficult to read) to 100 (very easy to read). The mathematical formula used is given in (5):

$$(5) 206.835 - 1.015 \left(\frac{\text{words}}{\text{sentences}} \right) - 84.6 \left(\frac{\text{syllables}}{\text{words}} \right)$$

The scores produced by the formula in (5) are mapped on seven scale levels that enable their interpretation (Table 1).

Score	Notes
90-100	Very Easy (Easily understood by an average 11-year old student)
80-90	Easy
70-80	Fairly Easy
60-70	Normal (Easily understood by 13 to 15 year old students)
50-60	Fairly Difficult
30-50	Difficult
0-30	Very Difficult (best understood by college graduates)

Table 1. Readability levels used by the Flesch Reading Ease Scale

3.1.7. Linsear Write Readability Formula

Linsear Write has been specifically designed to calculate the United States grade level of a text sample, measuring mean length of sentences and the number of multi-syllable words (i.e. words with more than 3 syllables).

3.2. Readability formulas adapted to Greek

The four most common readability indicators [Flesch Reading Ease (Flesch,1948), Flesch Grade level (Flesch & Kincaid, 1976), SMOG (Gunning, 1952) and Flesch Fog Index (McLaughlin, 1969)] have been used in the context of creation and construction of the software GRVAL 1.1.

GRVAL 1.1 is primarily an automated process of inference as to the degree of readability of Modern Greek texts. It enables the evaluation of the degree of difficulty of examinations texts with the use of a very simple tool, available online at http://www.greek-language.gr/greekLang/modern_greek/foreign/tools/readability/index.html.

4. Methodology

4.1. Description of our approach

Our main goal is to construct text classification algorithms in order to be able to sort different texts based on the difficulty for a particular user. We start by creating a simple algorithm of classifying the smaller parts that the text includes, the words. After that, and based on our results we generalize this concept in a way which gives us the ability to classify a text. Central to our approach is the notion of the *User Model* for users with dyslexia has been developed as part of the iLearnRW project.

4.2. User Modeling

Not all children with dyslexia demonstrate the same set of difficulties. As a consequence, not all children make the same reading errors and, in addition, even if they make the same type of reading errors the severity may be different. The same applies to spelling errors (dysorthographia). The user model includes, among other things, the error types the user is likely to make and their severity and the learner's age, as well as information related to the learning history and progress during the usage of the system.

User Modelling is based on the following, simple, idea: by having information about a specific individual a given computer system can make decisions which are best suited to that individual. Any user model consists of three components; the data being stored about attributes of a user, the algorithms which process this data to affect change on the computational environment and the method by which the data is obtained and updated.

The content classification module makes use of the *User Model* since, by tracking the specific individual difficulties a given child has, it can provide her appropriate texts for study. Ideally, this is what teachers would like to do in their classrooms. However, the time necessary to interpret the *User Model* of each child in a class, and subsequently produce an individual teaching plan appropriate for that child's specific difficulties and skills for each lesson, is beyond the time resources of nearly all teachers. However, this is something well within the abilities of a computer using a *User Modelling* component.

4.2.1. Description

Next, we describe the basic aspects of the *User Model* notion and the reasons that make this component central to the content classification module.

The user model is intended to provide data to other components through holding information about a given student's linguistic abilities and weaknesses. The full description of the User Model can be found in deliverable 4.1 (User Modelling) of the iLearnRW project. For the purposes of this document, we present a high level description of the linguistic difficulties as

they are organized into lists of similarities (the data contained in these lists also called *profile entries*). We also explain the notions of *severity*, *working index* and *tricky words*.

User Profile Entries

By the term *User Profile Entries* we refer to the set of problems that a child may have. We have grouped these problems based on the linguistic difficulty that they cover. In addition, the experts sorted these problems based on their difficulty in ascending order.

The *User Model* can be considered to be a two-dimensional array incorporates the following information:

- 1) Each cell (also referred as *profile entry*) of the array includes a description of a single problem
- 2) The i^{th} row contains only problems of a specific linguistic difficulty
- 3) The problems in the i^{th} row are placed starting from the easiest (leftmost) to the most difficult (rightmost)

The profile entries are language related. Next, we briefly present the linguistic areas that are covered by the iLearnRW project for the English and the Greek *User Models*. Full details appear in D4.1.

English Profile Entries:

- 1) Syllable division: the difficulty some children have in dividing longer words into smaller chunks (i.e. syllables) which are more manageable.
- 2) Vowel sounds: refers to the challenge that occurs due to the fact that, in English, there are many vowel sounds which share the same letters (e.g. “i” in did vs. “i” in ivy).
- 3) Suffixing
- 4) Prefixing
- 5) Grapheme/phoneme correspondence: similar to vowel sounds but with consonants (e.g. the phoneme /sh/ appears as “sh” in shop and “s” in sure).
- 6) Letter patterns: the difficulty some letter patterns have (e.g. “mb” in bomb).
- 7) Letter names: children need to learn the names of the letters in the alphabet.
- 8) Irregular/sight words: those words which do not follow any of the patterns within English (e.g. sword).
- 9) Confusing letter shapes: some graphemes are visually similar (e.g. “b” and “d”) which can be challenging for children with dyslexia.

Greek Profile Entries:

- 1) Syllable division
- 2) Phonemes (Consonants): some words may be confused with others due to a sound similarity among them. This category contains only problems caused by sound similarity of consonant letters.
- 3) Phonemes (Vowels): same as above, but in this category we consider only problems caused by sound similarity of vowel letters.
- 4) Suffixing (Derivational)
- 5) Suffixing (Inflectional / Grammatical)
- 6) Prefixing
- 7) Grapheme/phoneme correspondence
- 8) Grammar / Function words

Severity Level

In D4.1 we defined the *severity level* to be an integer that associates each specific case of difficulty as to whether it always occurs (3), sometimes occurs (2) or never occurs (1).

However, in the implementation, we decided to utilize a scale that the severity scale that runs from 0-3. We use the extra level to describe the case “never occurs”.

Working Index

To fully describe a user’s profile we associate each language area with a *working index*. For example, the Syllable Division category includes 20 entries for the Greek Profile (see the first row in Figure 1). Each entry corresponds to a specific instance of the difficulty and it has been positioned in the row in order of learning complexity. This means that if a student is currently working on the 6th, then she has already worked on entries 1-5 (or has acquired to them a satisfactory degree) and is currently working on the specific instance illustrated in 6. This information is important to the Text Classification algorithms and is referred in Figure 1, the *working index* is denoted by the black boxed squares.

Tricky-Words List

The system also allows a user to “store a personalized word bank”. This is a list of words that is created by each user. Other components of the system (most notably the reader) that allow her to identify and store words that she “struggles” with. Within the iLearnRW project the word bank has been termed as the *tricky words list*.

4.2.2. User Model Visualization

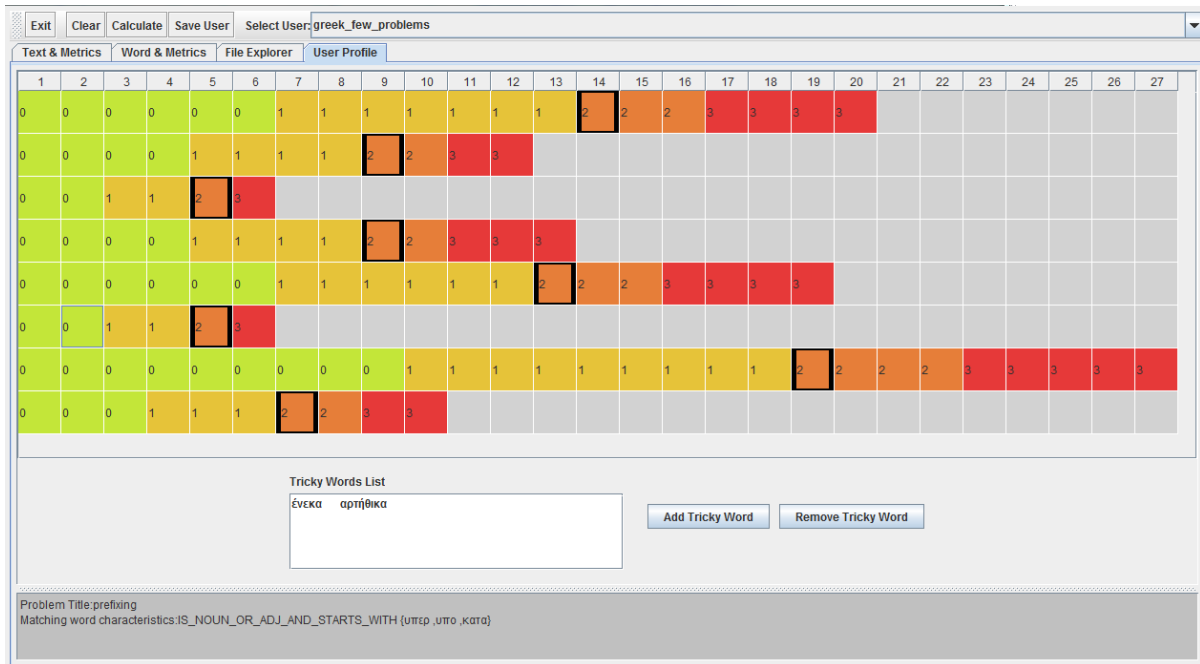


Figure 1. User Profile Viewer

In Figure 1, the Greek profile is presented as it appeared on the iLearnRW demo application which was created for the purposes of the first annual review of the project (we choose to show the Greek profile since it has a more compact view compared to the English profile). Each colored line of the matrix represents a linguistic difficulty while the individual cells display special problems that belong to this linguistic difficulty. The numbers inside each cell describe the severity of the problem for the specific user. In addition, the black bordered cells demonstrate the user's working indices (there is only one in each row). In the white box below the "User Model table" contains the list of tricky words that are associated with the user. Finally, the information frame at the bottom of the screen displays the description of the problem that is described/contained by the clicked cell (the light colored cell in the 6th row, 2nd column).

In terms of content classification, the User Model needs to provide no information beyond the data we hold about a child's difficulties. Based on User Profile, and as a first approach, a text can be classified on the basis of the number of words within it containing a difficulty (such as the "-ing suffix") and, consequently, rate how difficult it is to read.

Observe that Figure 1 shows the profile of a user with few problems since this table has a high percentage of low severities (0 or 1) and also the working indices are set to be close to the end of each linguistic difficulty.

4.3. Classification of words/text

In order to describe the *Text Classification Component* we first need when a word or a text is considered to be difficult. To achieve that, we utilize some metrics which take into account the User Profile.

4.3.1. Word Related Functions

We first start by defining three functions that describe profile and word properties in a more mathematical sense. For simplicity and without loss of generality we assume that the profile is a two dimensional table. So, each entry of the profile is referred by two indices (i,j).

When a word has a structure that falls into the description of the (i, j) profile entry we say that the word *matches* problem (i, j).

We continue now to define three mathematical functions that will be used to derive the score of a word:

- 1) Let $hit(int\ i, int\ j, Word\ w) \rightarrow \{0,1\}$ be an indicator function that returns 1 if word *w* *matches* profile entry (i,j), 0 otherwise. We assume that indexes i, j always refer to a valid profile entry. Note that if a word *matches* the same profile entry for more than one reasons, then it is counted only **one time**, for example the word "probability" matches the 8th problem of the Letter Word Patterns (contains both "ob" and ab" patterns - see deliverable 4.1) more than once.
- 2) Let $severity(Profile\ p, int\ i, int\ j) \rightarrow \{0,1,2,3\}$ be the severity of profile *p* that corresponds to profile entry (i,j).
- 3) Let $workingIndex(Profile\ p, int\ i) \rightarrow \{0,1, \dots, n\}$, where *n* is the length of the *i*-th profile row, be the working index of profile *p* that corresponds to the *i*-th linguistic difficulty.

4.3.2. Classification of Words

We are now able to firstly define the *word score* based on which we then characterize a word as *difficult word*, or *very difficult word*.

Word Score

Word Score, denoted by *Wscore*, is a metric that is defined to be the sum of all the severities of the user's profile entries matched by the word. A more formal definition follows in the next two lines:

$$Wscore(Word\ w, Profile\ p) \rightarrow N$$
$$Wscore(w, p) = \sum_{i,j} (hits(i, j, w) * severity(p, i, j))$$

As it is easy to see, the bigger the score is the more difficult the word is since in this case either the word matches more problems or it matches problems with higher severities.

Difficult Word

In this section we provide three alternative definitions of the *difficult word*. All of them use natural assumption that a word is more difficult for the user if it *matches* many problems or problems with higher severities or problems that are beyond her working index.

Let w be a single word, p be a valid user profile and $Wscore(w,p)$ be the score of the word for the specific user. Let $severity(p, i, j)$ and $hit(i,j,w)$ be the functions described in §4.3.1. The following list contains alternative definitions for the notion of difficult word:

Definition 1: w is considered to be **difficult** for profile p if there is at least on pair of indices i, j such that $hit(i,j,w)=1$ and $severity(p, i, j)>1$. This means that a word is considered to be difficult if it matches at least one user's problem with severity >1

Definition 2: w is considered to be **difficult** for profile p if $Wscore(w,p)>1$. This means that a word is difficult when it matches with at least two user's problems of severity 1 or with at least one problem of severity bigger than 1.

Definition 3: w is considered to be **difficult** for profile p if there is at least on pair of indices i, j such that $hit(i,j,w)=1$ and $workingIndex(p, i)\leq j$. That is, a word that matches at least one user's problem that is beyond his/her current working indices

Very Difficult Word

After giving several characterizations for the notion of a difficult word we now provide a definition for words that are considered to be *very difficult*. By having such a characterization for each word we can take advantage of it by treating these words in a special manner when they met inside a text.

Definition: w is considered to be **very difficult** for profile p if $Wscore(w,p)\geq 6$.

Note that for a word to be assigned of a score greater of equal to 6 it has to either match at least two problems of the greatest possible severity (i.e. 3) or at least 3 problems. For a word that matches only problems of severity 1, it has to match at least 6 problems in order to be classified as very difficult.

Based on the presented quantification on word metrics, we are now able to define the *text score* (denoted by $Tscore$) which, informally, is a positive number that describes the difficulty of the text. After having such a metric we can rank texts by sorting them according to their text scores.

Text Score

Let T be a text, i.e., a collection of words (in this definition, the order is not important)

- 1) Let $appearances(Text\ T, Word\ w) \rightarrow N$ be a function returning the number of appearances of word w in text T .
- 2) Define $Tscore(Word\ w, Profile\ p) \rightarrow R$ to be the iLearnRW-score for text T with respect to profile p , as follows:

$$Tscore(T, p) = \sum_{w \in T} \left(\frac{appearances(T, w) + 1}{2} * Wscore(w, p) \right)$$

The above formula that we use to calculate the text score captures, in a high level description, the magnitude of the user severities on problems areas that are relevant to the words in the text. That is, the more problematic words the text has (with respect to the user), the bigger the $Tscore$ is.

The term $\frac{appearances(T, w) + 1}{2}$ is derived as follows: we suppose that when a word is repeated in the text then its weight (i.e. its difficulty) reduces at each repetition. That is, if a reader sees a word multiple times inside a text then the word starts to become more familiar to her.

More precisely the i -th appearance of a word contributes $\frac{appearances(T, w) + 1 - i}{appearances(T, w)}$ of the word's $Wscore$. Doing the sum for all i , we get the total weight of a word to be:

$$\frac{appearances(T, w)}{appearances(T, w)} + \frac{appearances(T, w) - 1}{appearances(T, w)} + \dots + \frac{appearances(T, w) + 1 - appearances(T, w)}{appearances(T, w)}$$

The latest sum is equal to the following:

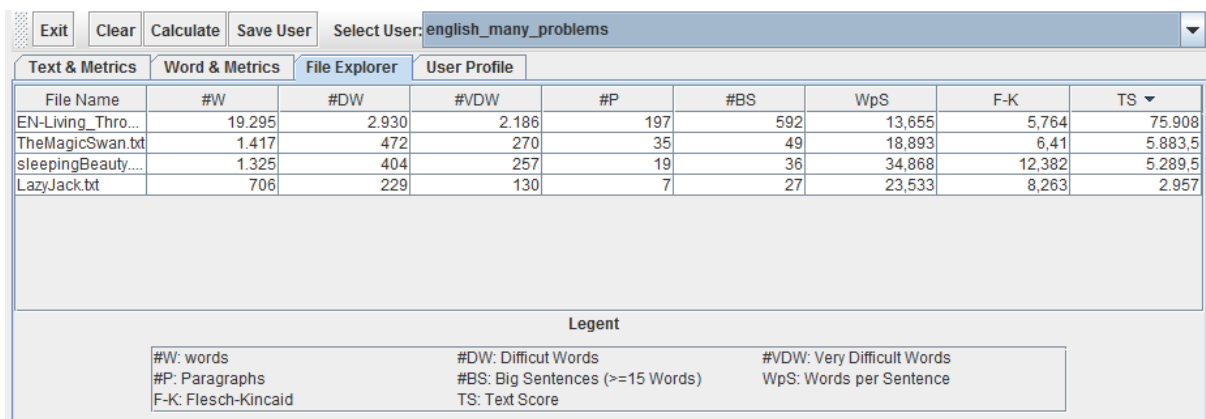
$$\frac{appearances(T, w) + (appearances(T, w)) + \dots + 1}{appearances(T, w)} = \frac{appearances(T, w) + 1}{2}$$

The metric $Tscore$ presented above is just a first approach towards text classification which takes into account the profile of an individual user. Its accuracy in ranking texts remains to be proven and it will be certainly refined. Given the volume of work available in the literature for users without dyslexia, it is fair to assume that several PhD dissertation maybe written in this topic. To address the problem of potential shortcomings of the $Tscore$ metric, we decided to make available to the user of the Text Classification component (i.e teachers, experts, parents and children) additional frequent used metrics which they can take into account when they evaluate the suitability of the text. These metrics are:

- 1) Total number of paragraphs, sentences, words and syllables
- 2) Total number of difficult words, very difficult words and polysyllabic words
- 3) Total number of big sentences (sentences that contain at least 15 words)

- 4) Average number of words per sentence and syllables per word
- 5) Generic readability tests:
 - Flesch
 - Flesch - Kincaid
 - SMOG
 - Gunning FOG
 - Automated
 - Coleman - Liau
 - Dale - Chall

In Figure 2 we can see the file manager of the demo iLearnRW application. This component makes use of the content classification module. So, we can see the different metrics that presented on the screen. The user is allowed to sort the text by different text metrics after clicking on the header of its column. The legend (at the bottom of the figure) provides explanation on the header of each column.



File Name	#W	#DW	#VDW	#P	#BS	WpS	F-K	TS
EN-Living_Thro...	19,295	2,930	2,186	197	592	13,655	5,764	75,908
TheMagicSwan.txt	1,417	472	270	35	49	18,893	6,41	5,883,5
sleepingBeauty....	1,325	404	257	19	36	34,868	12,382	5,289,5
LazyJack.txt	706	229	130	7	27	23,533	8,263	2,957

Legend		
#W: words	#DW: Difficut Words	#VDW: Very Difficult Words
#P: Paragraphs	#BS: Big Sentences (>=15 Words)	WpS: Words per Sentence
F-K: Flesch-Kincaid	TS: Text Score	

Figure 2. File Explorer

5. Techniques

In this section we present the main ideas and the tools that we used in order to implement the text classification component for the iLearnRW system. The rough idea behind the implementation of the component is the following:

- 1) Split text into sentences.
- 2) Split sentences into words
- 3) For each word:
 - a. Identify the problems the word matches in the profile matrix
 - b. Identify the user severity for each one of the matched problems with the word
 - c. Calculate the word score
- 4) Calculate the text score

Note that after the first 2 steps the iLearnRW calculates also all the generic text metrics. Next we describe the above steps thoroughly. But firstly we present the set of tools that we use in order to correctly interpret the words of the text.

5.1. Tools We Use

Text analysis tools are the tools we use for accessing and manipulating digital texts. Our system also depends on pre-constructed lexical data archives and other linguistic-software along with linguistic tools developed especially for the iLearnRW project.

Available Tools

The content classification module continually tries to determine the part of speech class a word belongs. This is achieved by using a dictionary for the Greek and English languages. For both of the languages the Hunspell dictionary is used (Németh, 2011). For the Greek language though, we created a dictionary that also contains the ‘part of speech’ information for each word as an extra module. Regarding the English language we use the PhoTransEdit application (PhoTransEdit) to convert words in to phonetic transcriptions.

Tools Developed Especially for the iLearnRW Project

For the purposes of the iLearnRW system the following linguistic analysis tools/modules were developed:

- 1) For each profile entry a function that takes as input a word and checks if the word’s structure matches to the description of the profile entry.
- 2) A function that takes as input a word and returns the list of profile entries that it matches.

- 3) A module that takes as input a text and outputs a list for each profile entry containing the words of the text that matched this problem.
- 4) Greek syllabification module (perform syllabification on a given word).
- 5) Greek phonetics module (create the phonetics transcript for a given word).
- 6) Extension of the Greek dictionary to contain the 'part of speech' information of each word.
- 7) Greek list of sound similarity word pairs (a list that contains words that when pronouncing sound similar to other Greek words).
- 8) A list of 5.000 most frequent English words (the size is actually closer to 15.000 since we added the derived forms of each word). Also a tool for generated derived forms from a list of 5,000 most frequent lemmas.
- 9) English list of pronunciations and syllabifications for the generated forms of the list of 15.000 words.
- 10) English typology of the difficulties identified in the profile
- 11) A procedure for identifying phoneme grapheme matches for English text.
- 12) Look up procedures for selected profile difficulties in text (a module which calculates the list of the user's problems that a word matches).
- 13) And a merging program to create the English dictionary file.

The Greek syllabification module is based on 9 rules of the Greek language that can be found in (Manolis Triantaphyllidis, 1991). Additionally, our implementation for the module of the Greek phonetics was based on an algorithm we produced after taking into account the rules for conversion of pure text to phonetics that we found in (Ager).

5.2. Text Processing

In this section we present the operations that take place during the process of the text classification algorithm. First is the text pre-processing operation which is a routine that gives us the ability to create all possible linguistic information for the structure of each word of the text. Then, for each word, we check it against the user's problems in order to get the severities of the user identified difficulties in the profile.

5.2.1. Text pre-processing

The text pre-processing is the task of converting a raw text file, essentially a sequence of words, into a well-defined sequence of linguistically-meaningful units.

As mentioned above the first process that takes place when a text (or a set of texts) is inserted to the iLearnRW system is to split it into smaller linguistic components. At the lowest level,

words consisting of one or more characters (from which we can take information about characters representing the individual graphemes) and then sentences consisting of one or more words. Text pre-processing is an essential part of any content classification component, since the characters, words, and sentences identified at this stage are the fundamental units from which the classification metrics will arise.

After having such a separation of the text components the phonetic conversion tools, the syllabification modules and also the dictionaries are used by the system in order to calculate the set of the matched problems for each word of the text.

5.2.2. Check words against user's problems

From the analysis of each word into graphemes we can then check if a single word matches to any problem of the user profile as we discuss above. After having a set of matched problems for each word then we compare these problems with the corresponding problems that the user profile contains. Then, we can check the severities of the user to each one of them and so, a weight is assigned to each word. We then complete the calculation of the text metrics since we have all the required information about the text and consequently we can apply each one of the formulas that represent text metrics.

5.3. Server-Side Component

Recall that the text classification module in order to complete its functionality needs to consult (and search) a relatively big set of data (modified/enhanced dictionaries). This makes the text classification a “heavy” task with respect to computational resources. So, we have chosen to move this component to the server side of our system in order to take advantage of its large storage and of course of its computational power. A client (tablet) then has to send its texts to the classifier each time the user profile is updated (because then the severities are changed yielding different text scores) or new texts are being added. Then, the server responds to the tablet by sending the metrics of the relevant texts.

6. Applications

In this section we discuss possible applications of the text classification. We present an application which was presented in the 1st annual review of the iLearnRW project and demonstrates an integrate usage of the text classification component.

Text classification with respect to the degree of appropriateness for a particular user (based on her profile) will be widely used in order to search for appropriate content for a particular user. The above makes this module to be a major component of the on-line recourse bank that will be supported by ILearnRW.

6.1. ILearnRW text classification tool

The application looks the profile of an individual user and it supports the following functionality: (i) profile viewer, (ii) file explorer with text ranking, (iii) word analysis and (iv) text analysis.

6.1.1. User Profile

We already have explained the user model and the usage of it. In Figure 3 we present another instance of a Greek profile (we prefer Greek profiles on our examples due to the smaller size of the Greek problems list compared to that of English) which represents a user who has a relatively large set of dyslexia difficulties.

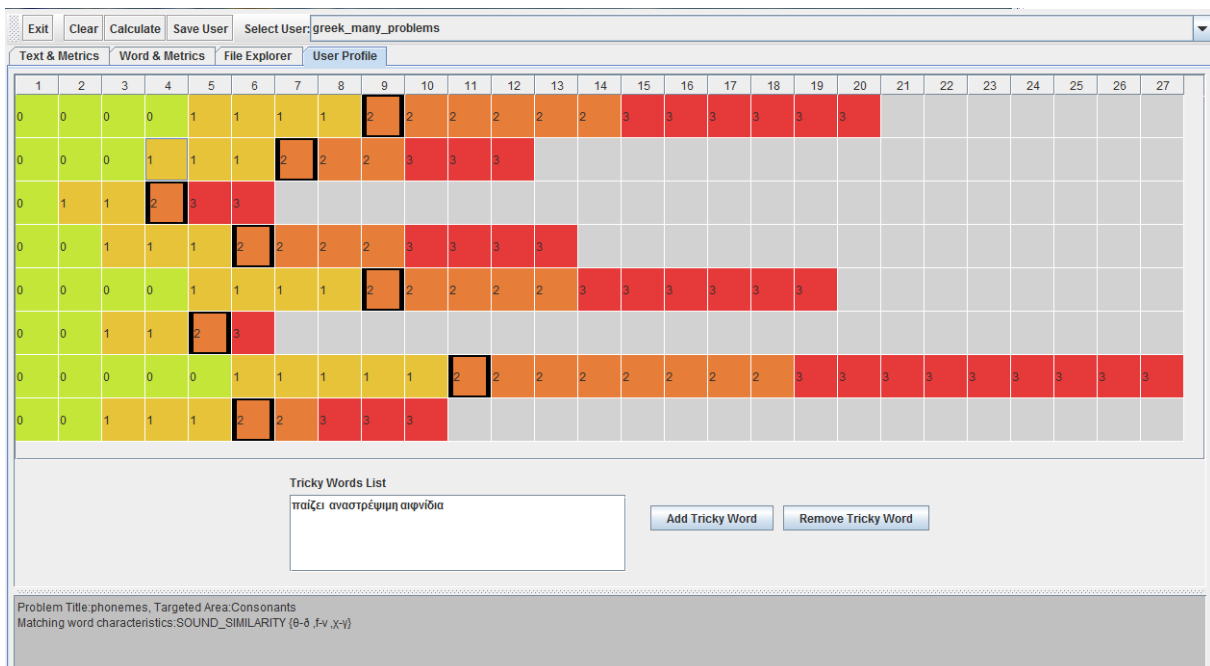
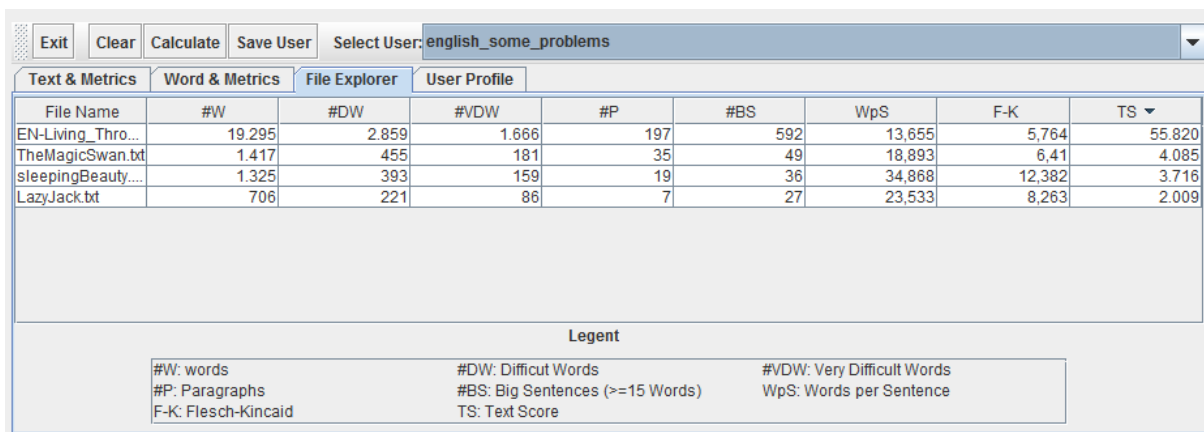


Figure 3. User Profile Viewer

The text classification tool that we have at the moment is able to display and examine a user profile. It also supports edit operations on profile entries, making it a natural tool for *profile creation* (or *manual initialization* of a profile by experts).

6.1.2. File Explorer

Another feature of the text classification application is that it is able to display a list of texts (text files loaded from a fixed directory) and display some important text metrics to the user. The user has also the ability to sort the texts by her chosen metric. This tool is language sensitive, that is, it presents only texts that are in the same language of the user's language (this information is available in the user's profile). In Figure 4 we give a screenshot of this application for a set of four English texts and an English profile.



File Name	#W	#DW	#VDW	#P	#BS	WpS	F-K	TS
EN-Living_Thro...	19,295	2,859	1,666	197	592	13,655	5,764	55.820
TheMagicSwan.txt	1,417	455	181	35	49	18,893	6,41	4.085
sleepingBeauty...	1,325	393	159	19	36	34,868	12,382	3.716
LazyJack.txt	706	221	86	7	27	23,533	8,263	2.009

Legend

#W: words	#DW: Difficut Words	#VDW: Very Difficut Words
#P: Paragraphs	#BS: Big Sentences (>=15 Words)	WpS: Words per Sentence
F-K: Flesch-Kincaid	TS: Text Score	

Figure 4. File Explorer

6.1.3. Word Metrics

The iLearnRW software also allows an expert to insert words and then check the dyslexia problems (i.e profile entries) that are relevant to the word and the specific user. In Figure 5 we present the operation of this component when it has as input a Greek student (with a large number of problems) and the Greek word 'παίζει'. We can see the Greek profile matrix with "highlighted" (i.e. in red) the entries which satisfy the following conditions:

- 1) The word matches to this problem
- 2) The user has a severity larger that 0 to this problem

The above means that a cell is colored red if the corresponding problem is both an issue for the child and also the word's structure matches to its description.

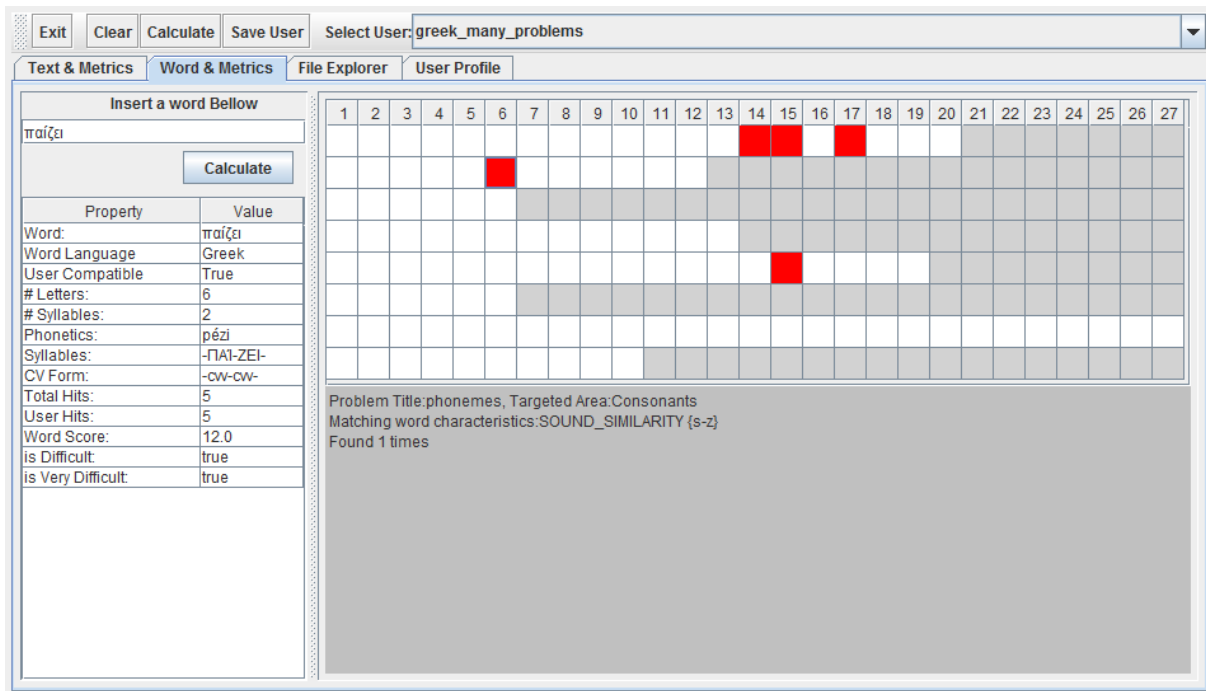


Figure 5. Word Metrics Panel

Furthermore, a list of word related properties are displayed in a matrix such as the syllabification of it, its phonetics, the number of total hits to the problems matrix (independent of the user’s problems), the number of user hits (i.e. the number of “red” cells) the *Wscore*, the metrics “difficult word” and “very difficult word”.

6.1.4. Text Metrics

Maybe the most important feature of this demo application is the panel that can display information about a text compared against the problems of a child. In Figure 6 we can see this panel for a Greek user with a large number of problems and a Greek text. The two dimensional matrix displays either white or variations of red cells. The rule to add colors to the matrix is, roughly, the following: the more the words of the text having structure that matches to the cell’s corresponding problem, the bigger the “redness” of the cell is.

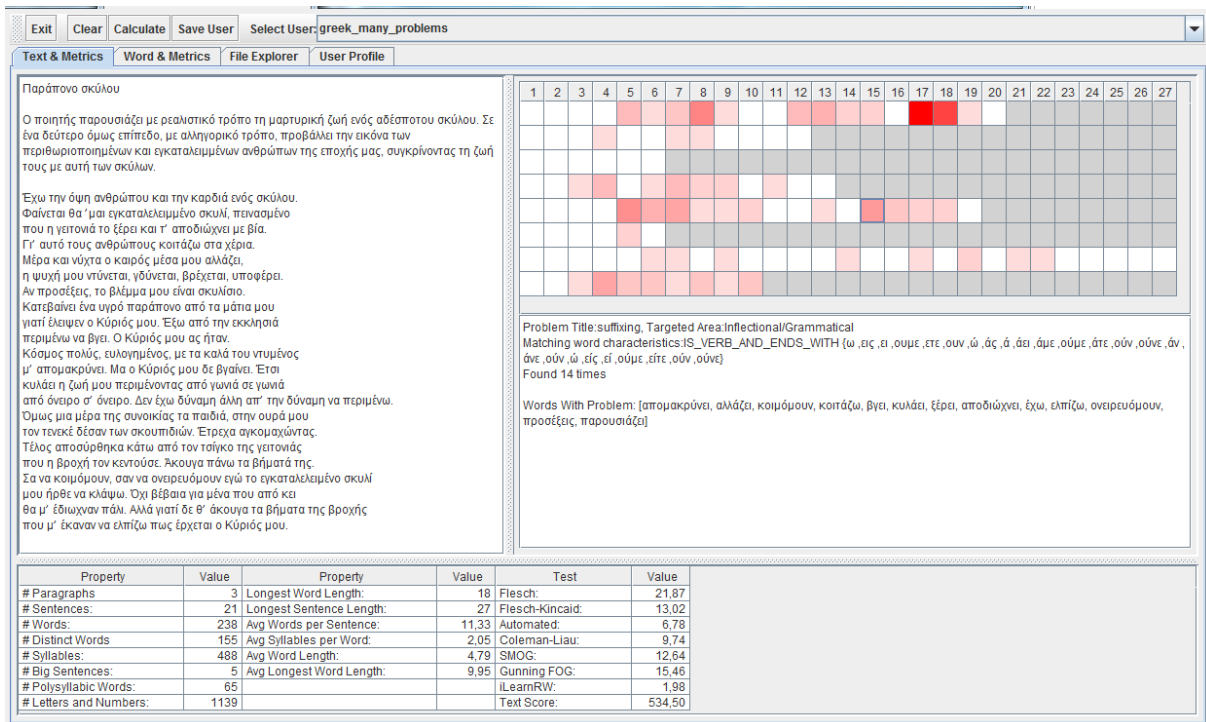


Figure 6. Text Metrics Panel

By clicking a cell one can see on the right of the panel inside the white text area the description of the corresponding problem and also the words of the text that found to match this problem.

On the bottom of this table we present all the available text metrics that we can extract from the text.

7. Conclusions

In this deliverable we have described the text classification component of the iLearnRW system. Central to the success of the system is the quality of the available language resources. The current version of the resources sufficiently supports the operation of the text classification, however, we continue working toward their improvement. If required an updated version of this deliverable which reflects our continuing efforts will be issued.

References

- Arends, J. (2001). Simple grammars, complex languages. *Linguistic Typology* 5 (2/3): 180–182.
- Blache, P. (2011). *A computational model for linguistic complexity*. In proceedings of the first International Conference on Linguistics, Biology and Computer Science.
- Chall, J.S. & Dale, E. (1995). *Readability revisited: The new Dale–Chall readability formula*. Cambridge, MA: Brookline Books.
- Chall, J.S., Bissex, G.L., Conard, S.S., & Harris-Sharples, S.H. (1996). *Qualitative Assessment of Text Difficulty: A Practical Guide for Teachers and Writers*. Cambridge, MA: Brookline Books.
- Flesch, R. (1948). A new readability yardstick. *Journal of applied psychology*, 32, 221-233.
- Fry, E.B. (1968). A readability formula that saves time. *Journal of reading*, 11, 513–516.
- Gil, D. (2008). How complex are isolating languages? In: Matti Miestamo, Kaius Sinnemäki, and Fred Karlsson (eds.), *Language Complexity: Typology, Contact, Change*, 109–131. Amsterdam, Philadelphia: Benjamins.
- Hess, K., & Biggam, S. (2004). A Discussion of “Increasing Text Complexity”. An article produced in partnership with the New Hampshire, Rhode Island, and Vermont Departments of Education.
- Kincaid, J.P., Fishburne, R.P., Rogers, R.L., & Chissom, B.S. (1975). *Derivation of new readability formulas (Automated Readability Index, Fog Count, and Flesch Reading Ease Formula) for Navy enlisted personnel*. CNTECHTRA Research Branch Report.
- Klare, G. (1984). Readability. In: P.D. Pearson, R. Barr, M.L. Kamil, & P. Mosenthal (eds.), *Handbook of reading research* (pp. 681-744). NY: Longman.
- Kusters, W. (2008). Complexity in linguistic theory, language learning and language change. In: Matti Miestamo, Kaius Sinnemäki, and Fred Karlsson (eds.), *Language Complexity: Typology, Contact, Change*, 3–22. Amsterdam, Philadelphia: Benjamins.
- Lipson, M., & Wixson, K. (2003). *Assessment and instruction of reading and writing difficulty: An interactive approach* (3rd ed.). Boston: Allyn & Bacon.
- McWhorter, J. (2008). Why does a language undress? Strange cases in Indonesia. In: Miestamo, M., Sinnemäki, K. and Karlsson, F. (Eds.), *Language Complexity: Typology, Contact, Change*, 167–190. Amsterdam, Philadelphia: Benjamins.
- McWhorter, J. (2001). The world’s simplest grammars are creole grammars. *Linguistic Typology* 6: 125–166.
- McWhorter, J. (2007). *Language Interrupted: Signs of Non-native Acquisition in Standard Language Grammars*, Oxford University Press.
- Miestamo, M. (2008). *Grammatical complexity in a cross-linguistic perspective*. In: Miestamo, M., Sinnemäki, K. and Karlsson, F. (Eds.), *Language Complexity: Typology, Contact, Change*, 23–42. Amsterdam, Philadelphia: Benjamins.

- Sawyer, M.H. (1991). A review of research in revising instructional text. *Journal of Reading Behavior*, 23(3), 307–333.
- Shosted, Ryan K 2006 Correlating complexity: A typological approach. *Linguistic Typology* 10: 1–40.
- Stenner, A.J., Horabin, I., Smith, D.R. and Smith, M. (1988). Most comprehension tests do measure reading comprehension: A response to McLean and Goldstein. *Phi Delta Kappan*, 69, 765-767.
- Szmrecsanyi, B. & Kortmann, B. (2012). Introduction: Linguistic complexity – Second Language Acquisition, indigenization, contact. In Kortmann, B., & Szmrecsanyi, B. (Eds.), *Linguistic Complexity: Second Language Acquisition, Indigenization, Contact*. De Gruyter.
- Ager, S. (n.d.). *Omniglot - the online encyclopedia of writing systems and languages*. Retrieved from Omniglot: <http://www.omniglot.com/writing/greek.htm>
- Triantaphyllidis, M. (1991). *Νεοελληνική Γραμματική (της Δημοτικής) / Greek language, Modern Grammar..* Thessaloniki: Aristotle University of Thessaloniki.
- Németh, L. (2011, 2 16). *Hunspell*. Retrieved from Sourceforge: <http://hunspell.sourceforge.net/>
- PhoTransEdit*. (n.d.). Retrieved from PhoTransEdit: <http://www.photransedit.com/>