MASTER'S THESIS

A Software/Design Method for Predicting Readability for ESL Students

Diana Cembreros Castaño

Advisor:
Dr. Paloma Tejada Caller

Departamento de Filología Inglesa I
Facultad de Filología
Universdad Complutense de Madrid

September, 2010

ABSTRACT

The objective of this research is to present a web application that predicts L2 text readability. The software is intended to assist ESL teachers in selecting texts written at a level of difficulty that corresponds with the target students' lexical competence. The ranges are obtained by statistical approach using distribution probability and an optimized version of the word frequency class algorithm, with the aid of WordNet and a lemmatised list for the British National Corpus. Additionally, the program is intended to facilitate the method of selection of specialised texts for teachers of ESP using proportionality and lists of specialised vocabulary.

This web application is a free and open source software system that enables ESL/ESP teachers to carry out a comprehensive speed analysis without requiring knowledge of either computational linguistics or word frequency distributions and the underlying logarithmic proportionality.

Table of Contents

List of Tables

List of Figures

Introduction

Reading in a foreign language can be frustrating for even the most motivated students. Therefore, L2 teachers are often faced with the challenging task of selecting texts that are likely to be comprehensible, usually driven by mere intuition and teaching experience. However, a piece of text seems to be easy or difficult depending on certain variables involved within the text which can be measured.

In the last century, considerable research has been devoted to the identification and possibility of measuring these parameters in order to predict text difficulty. In 1923, Bertha A. Lively and Sidney L. Pressey designed a method based on the index number of its words in a corpus sorted by frequency (Lively and Pressey, 1923) This study demonstrated the effectiveness of a statistical approach for predicting text difficulty and inspired most of the readability indices and formulas that would follow, such as Flesch reading ease, Flesch-Kincaid formula, Gunning-Fog reading index or Coleman-Liau index. (Dasa, 2006 : 148-21) These tests also take into consideration other parameters such as average sentence length, average number of syllables per word or lexical coreferentiality. (Fotzl, Kintsch and Landauer, 1998 : 240-258). Processing those variables in large pieces of text is an extensive and complex process only made possible by Computational Linguistics.

The different automated algorithms developed to calculate these readability indices are mainly designed to measure children's reading comprehension of a text written in their mother tongue, as factors such as reading ability or cognitive development expected for

each age rank are taken into consideration. (Miller and Kintsch, 1980 : 335–354) Therefore, educators who teach L1 reading abilities to children can benefit from these indices to determinate whether a book is appropriate for their students.

Yet, little research in the effectiveness of these tests for adult ESL students has been conducted; therefore, ESL teachers need a different method to select graded reading material. One possibility is to work with levelled books adapted for ESL students[1], but the catalogues can be very limited for the teacher's needs. Another possibility is to do a manual selection of texts without previous classification. In order to do this, teachers need to spend a great deal of time reading different books and grading them.

The process of text selection is even more complex for teachers of English for specific purposes (ESP), and teachers of translation. In these fields, it is essential to work with authentic texts, and for some sectors it is necessary to work with up-to-date material. Thus, finding appropriate texts for ESP require a considerable amount of work and time.

For a preliminary analysis, teachers can use software lexical tools such as Wordsmith, but a certain degree of knowledge and practice in Computational Linguistics is required. Also, software lexical tools are mainly designed for research rather than teaching, so "teaching material selection" is not an automated process and needs several steps for each text analysis. Additionally, these programs are very complex tools which are too expensive for widespread use.

[1] Companies such as Oxford University Press or Penguin Readers have published a series of classical books simplified and graded for ESL learners.

The aim of this study was to design and implement a computer program which can assist ESL teachers, regardless their computational skills, in selecting texts written at a level of difficulty that corresponds with the target students' lexical competence. Additionally, the software should be able to work with technical texts and determine if they might be adequate for its use as ESP class material, by analysing the proportion of core vocabulary within the text.

The remainder of this paper is organised as follows: in Section 2 we describe the materials used; in Section 3 we review the most relevant theories on language frequency distribution, as well as some previous work and technical State of the art in Computational Linguistics is; in section 4 we describe the approach, that is, the technical overview the and main linguistic functions of each component of the software; in section 5 we detail the implementation, architecture and algorithms; Section 6 presents and discusses the results of the several case studies analysed by the software; in Section 7 we draw some conclusions and discuss some proposals for future work.

1. Materials

1.1 Corpora and word lists

- British National Corpus (BNC).
- A list of 413 Political Economy terms compiled by Paul M. Johnson, Auburn University (Johnson, 2005).
- A lemmatized frequency list compiled by Adam Kilgarriff, which contains the 6,318 words with more than 800 occurrences in the whole 100M-word BNC. (Kilgarriff, 2006). According to the author, "the list-creation process replicated that used at Longman for marking dictionary frequencies in LDOCE 3rd edition" (Kilgarriff, 1997:12).
-

Kilgarriff's work has been modified to fit this application's objective. The most relevant modifications are the following:

Firstly, the original list excludes names and items that would usually be capitalised. The modified version includes months, days of the weeks, countries, major cities and languages. These terms are usually learned soon, so they were assigned a high frequency index.

Secondly, as the original list is lemmatised, the verbs appear as bare infinitives. Irregular past forms have been included as separate tokens that are searched before lemmatizing the INPUT text. As irregular verb forms are usually learnt systematically rather than by

exposure, the frequency index is assigned taking as a reference the lists of irregular verbs as published in the different levels of the Oxford Exchange series by OUP.[2]

1.2 Sample texts for analysis

a) A sample paper of the Certificate of Proficiency in English (CPE) examination

b) An Economy article from the NY Times (Henriques, 2008)

c) Six Penguin Readers books (levelled books for ESL students):

Table 1: Penguin Readers books

Easystarts	Keen, A., (2003). *Carnival*. Upper Saddle River: Pearson Education.
Beginner	Kehl, J. &Twain, M., (2000). *The Adventures of Tom Sawyer*. Upper Saddle River: Pearson Education.
Elementary	Stevenson, R. L., (2000). *Treasure Island*. Upper Saddle River: Pearson Education.
Pre-Intermediate	Mikes, G., (2000). *How to be an Alien*. Upper Saddle River: Pearson Education.
Intermediate	Mahood, J., & Walsh, M. (2001). *As Time Goes By*. Upper Saddle River: Pearson Education.
Upper Intermediate	Grisham, J., & Waterfield, R. (1999). *The Pelican Brief*. Upper Saddle River: Pearson Education.

Excerpt of all texts are reproduced in Appendix 1.

[2] (Wetz, 2003a, 2003b, 2003c & 2003d)

1.3 External software

WordNet API, a lexical database developed by the University of Princeton.

Ruby version of the Porter Stemming Algorithm. (Pereda, 2003)

2. Theoretical Background

2.1 Corpus linguistics in predicting text difficulty

The Lively-Pressey method was based on three parameters: (a) the number of different words, (b) the number of "zero-index words", which are words which do not appear in a 10,000-word corpus named Thorndike list; and (c) the median of the index numbers of the words in the Thorndike list. (Lively & Pressey, 1923)

The Flesch Reading Formula is still implemented in word processing software programs. Flesch (1943) takes into consideraton the proportion of of words per sentence syllable length. Dale and Chall (1948) also based their formula in sentence length.

Many other variables have also been studied, such as the amount of abstract vocabulary (Flesch, 1943, Cohen, 1975) and syntactical complexity (Kintsch, 1974) However, there is not a universal criteria to define what is or is not abstract, and difficulties when analysing syntactical complexity of long texts.

However, vocabulary difficulty still seems to be the on of the most reliable variables in predicting readability. Freebody and Anderson (1983 : 277-293) demonstrated that performance is lower when the passages contain difficult vocabulary, and “in half of these cases the effect was significant”.

Corpus Linguistics tools

In order to conduct a study of word frequency which is corpus-based, it is necessary to use a corpus, a program that processes concordances, and a lemmatizing or stemming application.

In computational linguistics, a text corpus is a large and structured set of texts electronically stored. The Oxford English Corpus, developed by the Oxford University Press' language research programme, contains over two billion words. The British National Corpus (BNC), freely available under a license, is a 100-million-word text corpus.

Concordancers are used in corpus linguistics to retrieve linguistic data from a corpus, which the linguist then analyses. Among the widely used concordance tools is WordSmith, developed by Mike Scott at the University of Liverpool.

Lemmatization and stemming programmes find the normalised forms of words as they appear in text. It is useful to pre-process language before using probabilistic information retrieval systems. Specifically, lemmatization is “the process of grouping together the

different inflected forms of a word so they can be analysed as a single item" (Abney, 1989 : 102)., considering the context in which the word appears. Stemming is a similar process; however, a stemmer only works with general inflection rules in order to obtain the root, but ignores the context and the part of speech of the word as it appears in text. Stemming rules remove the inflectional ending from words:

```
calling, call, calls, called → call
```

But they can mix together semantically different words:

```
gallery, gall → gall
```

On the contrary, a lemmatizer will discriminate between the inflected verb "helping" and the noun "helping". Thus, stemming usually works faster and it is easier to compile but the results are not accurate if the application needs to obtain valid lemmas**.**

The most widely used and adapted stemmer is the Porter Stemming algorithm**,** developed by Martin Porter and encoded in a variety of programming languages such as Python, java, Perl or php (Abney, 1989). On the other hand, a very popular open source lemmatizer is included in WordNet, a lexical database developed by the University of Princeton. Another frequently quoted and adapted application is NLTK (Natural Language Toolkit), which is an open source Python module for research in natural language (Edmonds, 2007).

2.2 Frequency lists

A frequency list is a sorted list of words according to their number of occurrences in a given corpus. Frequency lists can be presented in either alphabetical or descending order,

where the most frequent word is placed in the first position. Some frequency lists also may contain additional information about the words, such as part-of-speech codes.

This program uses an enriched version of a lemmatized frequency list of the 6,318 words with more than 800 occurrences in the whole 100-million-word British National Corpus, compiled by Adam Kilgarriff. (Kilgarriff, 2006).

2.3 Distribution theories in quantitative linguistics

This application classifies words according to their frequency index. In order to interpret the results it is necessary to know the patterns of regularity and frequency distributions in language. The algorithms used in this program are based on Zipf's Law and the averaged Word Frequency Class algorithm.

Zipf's Law is an empirical law formulated using mathematical statistics. Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its index in the frequency list. Therefore, the most frequent item occurs approximately twice as often as the second most frequent word, which occurs twice as often as the fourth most frequent word, etc. (Zipf, 1935) For example, in the British National Corpus "the" is the most frequently occurring word, and by itself accounts for nearly 7% of all word occurrences. True to Zipf's Law, the second-place word "of" accounts for slightly over 3.5% of words.

Table 2: BNC Word Distribution

Occurrences	Word	Class Freq.
6187267	the	0
3093444	of	1
2687863	and	1
2186369	a	2
1924315	in	2
1620850	to	2
1375636	have	2
1090186	it	3
	[…]	
807	prone	13
807	marsh	13

A word's Frequency Class can be obtained with an algorithm based on Zipf's Law (Meyer, 2006). It indicates a word's rate of recurrence using the base 2 logarithm of the ratio between a word frequency index and the frequency index of "the", which is the most frequently used word in English. High frequency words have a low Class Frequency and marked forms have a high Class Frequency. As shown in Table 2, the word "the" corresponds to the word frequency class 0. Any word that appears approximately half as frequently belongs in class 1, and so on.

The formal definition of Frequency Class can be expressed as:

$$N = \left\lfloor -\log_2 \left(\frac{Frequency\ of\ the\ most\ common\ item}{Frequency\ of\ this\ item} \right) \right\rfloor$$

Where N is the Frequency Class, and $\lfloor \ldots \rfloor$ is the floor function.

3. Approach

3.1 Technical overview

- **Web application**

The program is a web application; that is to say, it is a program which is accessed with a web browser over the Internet or another network, such as a school's intranet. This means that the program does not need to be installed locally in the user's computer, but it is hosted in a server and users can access and run the application with a common web browser from any computer in the world.

- **Authentication system**

If the server can run an authentication system (access through a user name and a password), users can save their preferences, analysed texts and custom specialized lists directly in the server, so the data can be accessed from different computers. Another advantage is that the files do not need to be physically transported in a data storage device such as a USB flash memory (pendrive) from one computer to another.

- **Programming language**

The programming language used to encode the program is Ruby on Rails, an open source web application framework for the Ruby programming language. Ruby can be considered as a derivative of Perl language, and it is similar in varying respects to Python.

- **External applications**

For dictionary lookups, the program uses an external thesaurus called WordNet, the access is gained through an open source Application Programming Interface (API). An API is an interface implemented by a software program which enables it to interact with other software. In other words, WordNet provides the code to gain access to the WordNet's lexical database from another program.

- **Copyright License**

The program is released under the Open Source Educational Community License. This license allows individual teachers, schools, universities and other educational communities to use the software free of charge without limitation, and it is also a donation for the research community in order to improve the software and add new functionalities.

According to the Open Source philosophy, it is granted, free of charge, to any person to deal in the program without restriction, including without limitation the rights to use, modify, and distribute copies of the program. Any modification of the program must be released under identical conditions. (*See* Appendix 2)

3.2 Main functions of the software

The program consists of (a) a core software, (b) additional modules and (c) software complements. The core software is the main function, which is the analysis of the vocabulary of any piece of text in order to obtain an index of vocabulary difficulty. Modules are additional functions designed for specific types of texts, such as legal or technical English. Complements are supplementary features that add extra data to the results provided by either the core program or a module, such as a glossary of predictably difficult terms. Advanced users can design and add new modules or complements to the application.

3.2.1 Core software

The software works on the premise that, by exposure, the most commonly used words are learned first (Lively and Pressey, 1923). Therefore, we can assume that, statistically, a standard ESL student with a lexical competence of 500 headwords is likely to understand a text whose words are located in the first 500 positions of a corpus-based frequency list.

The aim is to obtain an index of readability of a given text based on the percentage of words that belong to each Frequency Class. Difficult texts correspond to high indices and easy texts belong to low indices.

The default settings of the program present 7 levels of difficulty, which are based on Zipf's Law and a generalization of the Word Frequency Class algorithm. Table 3 below shows

the correspondence between the index of difficulty and an approximation of the student's lexical competence required to understand around 85% of the texts.

Table 3: Relation of Levels and Lexical Competence

Level	Lexical competence
1	100 words
2	200 words
3	400 words
4	800 words
5	1600 words
6	3200 words
7	6400 words

3.2.2 Modules

The software includes a module specially designed for teachers of translation or English for Specific Purposes (ESP). The objective is to determine whether a given text is appropriate to be used as class material based on the density of specific vocabulary.

Density is calculated as a proportion between the length of a given text and the number of key words than appear in it. A high density implies that the text is likely to be appropriate for vocabulary learning; that is to say, it has a significant proportion of words which belong to the core vocabulary of the specific area and. The default settings of the software return 3 density levels: low, medium and high.

In order to identify key words, the software must have one or more specialised lists of the area of interest installed. The core vocabulary of each area is compiled in word lists which must be uploaded to the server. These lists can be easily added by a basic user, either by writing a new list from scratch or by modifying specialised glossaries which can be found in the Internet or technical books.

This module includes a fully functional demo based on a built-in list of 413 Political Economy terms compiled by Paul M. Johnson of Auburn University (Johnson, 2005).

3.2.3 Complements

The software includes a complement that builds automated glossaries of the predictably most difficult words that appear in a given text, based on their frequency index in the BNC. Depending on their needs, users can choose the extension of the glossary, either in absolute numbers (for example, “20 words”) or ranks (all words which have an index over “6”). Once built, the glossary can be copied and edited in a word processor.

3.3 Scope and limitations

Firstly, text difficulty is a qualitative variable which, by definition, can not be measured. Statistical approaches using probabilistic functions as models can give helpful estimations, but we can not assume that the judgement will be accurate for every particular reader.

Secondly, the fact that a word has a low frequency does not necessarily mean that the word is difficult, and vice versa. Students can guess the meaning of low frequency words that have a common etymological origin with their L1 and, on the contrary, false friends can lead to erroneous assumptions. The results may vary for ESL students depending on their L1.

Thirdly, this software does not identify homonymy and polysemy, as it requires a complex study of context. Words with multiple meanings may have a high index of frequency, but the software can not predict whether students will identify the meaning in a given text.

Fourthly, only single words are employed in this study; combination of words, such as expressions or idioms, are not processed as a chunk.

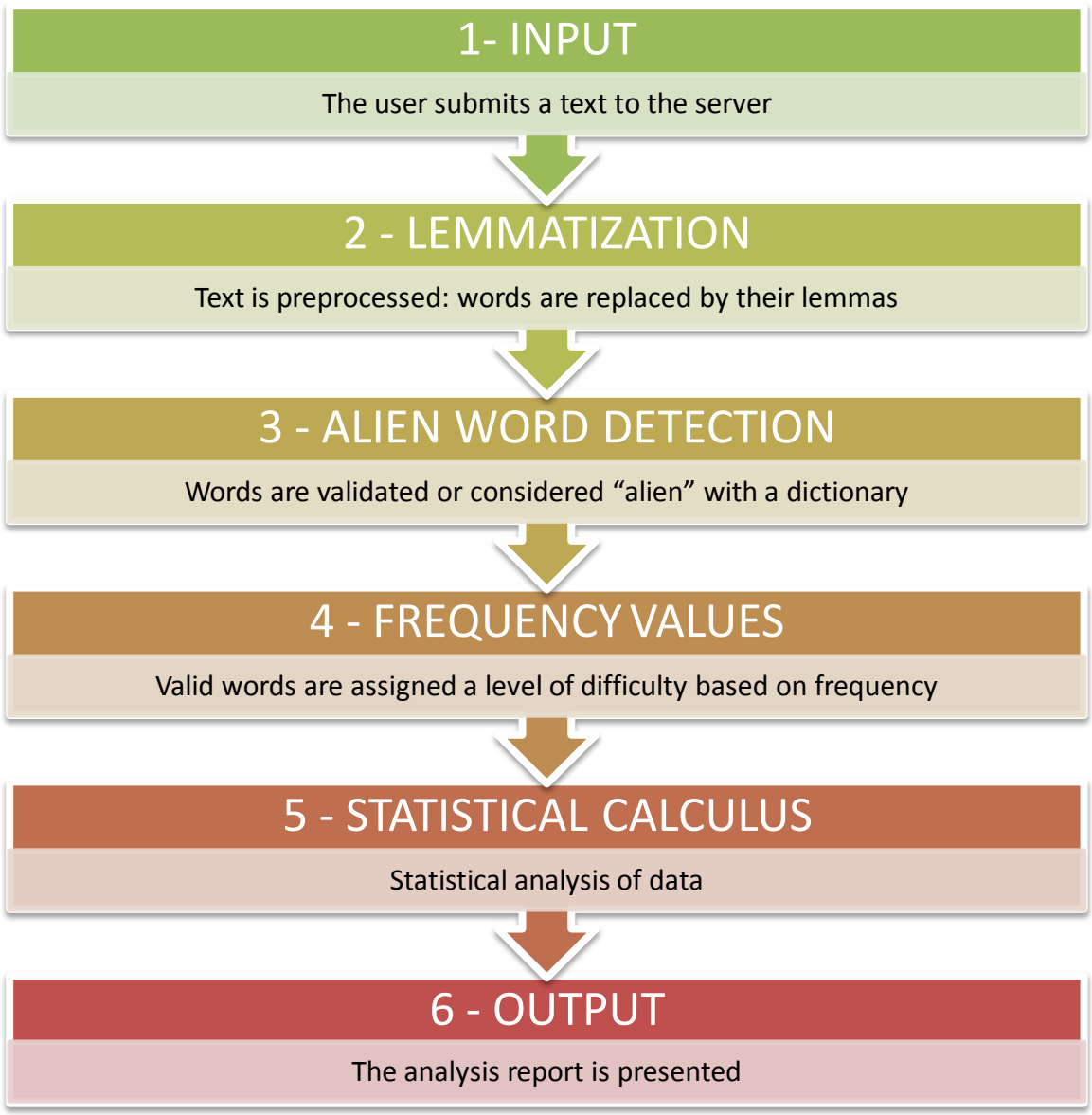
Finally, the core software study focuses on lexical competence; it can predict how many words a student is likely to understand. However, readability is affected by other factors, such as discourse coherence, syntactic complexity or cultural references, to name a few.

4. Implementation

In this section, each step of the algorithms is explained from a linguistic point of view; then, the statistical treatment of data and technical processes carried out in each step are detailed and illustrated with examples

4.1 Algorithm of the Core software

Figure 1. Algorithm of the Core Software



4.1.1 Linguistic justification of the Core Software algorithm

- Input

The user submits the text in electronic format. If the teacher only needs the difficulty level, then a sample of the text (at least 150 words) can give accurate results. For a complete analysis, including word count or the glossary complement, the full text is required.

- Lemmatization

Firstly, the text is preprocessed by lemmatization; that is to say, each word in the text is replaced by its headword or the morphological lemma this word-form belongs to.

In this process, the different inflected forms of a word are grouped together so they can be analysed as a single item. This process is essential for an analysis of the lexical competence required to understand a given text: if this lemmatization is not run, all the inflected forms in the original text would be analysed separately according to their actual frequency in the BNC, and the results would not be useful for this application's purpose. For example, the word "mothers" would return a level of difficulty much higher than the word "mother", as the singular form has seven times as many occurrences in the BNC as the plural "mothers". However, we can assume that if a student knows the meaning of the base-form of a given word, the meaning of most of its regular inflected forms can be deduced.

- Detecting alien words.

Each word is cross checked with a large lemmatized corpus in order to distinguish between actual English words and "Alien words". Alien words (also known as "zero words") are mostly foreign terms, misspelled words or proper nouns such as trademarks or names of places, and must be separately computed. If a word does not appear in the lexical database, the token is considered an Alien word and therefore the application will not be computed in some of the statistical operations. This step is essential in order to obtain accurate results; otherwise, a text with, for example, a considerable number of proper nouns would return a difficulty index higher than its correspondent level. Alien words are not expected to be learnt and do not usually affect the student's ability to understand a text.

- Assigning frequency values to each word

All validated words are assigned a level of difficulty in a scale from 1(easy) to 7(difficult). In order to do this, words are crosschecked with Kilgarriff's lemmatized frequency list, a 6,318-word corpus sorted by frequency of use based on the BNC. The words in the text that appear in the first positions of the list are frequent words and will be assigned a low level of difficulty, whereas the words towards the end of the list will be assigned a high level. Words validated in step 3 which do not appear in the frequency list are valid but very infrequent words, so they belong in the highest difficulty level.

- Statistical treatment of data

The original algorithm of word Frequency Class may return more than 20 different difficulty levels. Such precision can be useful for research but does not seem practical for selecting ESL teaching material, given the fact that reference frameworks do not usually divide the students' level of proficiency into so many categories. Therefore, the application presents the data in the number of categories that best matches the teacher's needs without losing accuracy, as this application is able to calculate new proportions based on the lexical distribution observed in Zimpf's Law. It seems practical that, by default, the application returns 7 levels of lexical competence: Level 1 texts are extremely easy and can be used with absolute beginners, while the remaining 6 levels can be adapted to the lexical competence expected in, for example, each of the 6 levels of the Common European Framework of Reference for Languages (CERF). However, the teacher can easily modify the settings to make the application return a different number of levels and adjust each of them to a specific lexical competence, such as each course of Primary or Secondary school, or any other reference framework.

- Output

The analysis report presents the following data:

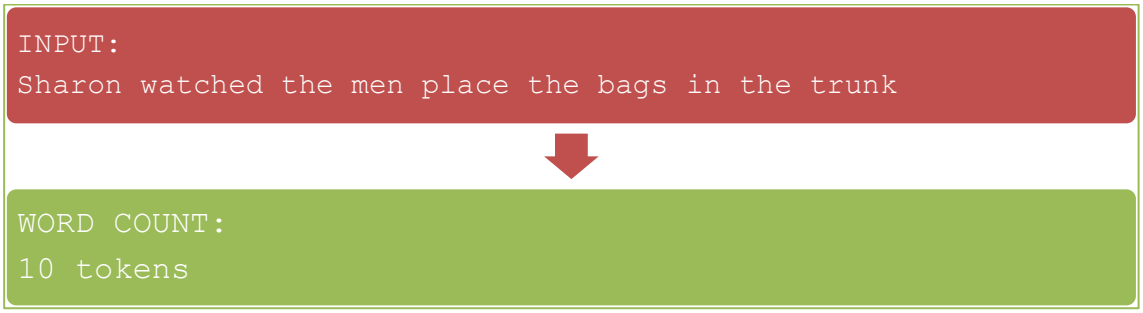
- Level of difficulty of the text
- Total word count
- Number of different words
- Number and distribution of unique words that belong to each difficulty level
- List of Alien Words (optional).

4.1.2 Technical processes and statistical operations of the Core Program algorithm

- Input

The application reads the text, divides the strings of characters into tokens (words) and counts the total number of tokens.

Figure 2. Input and word count



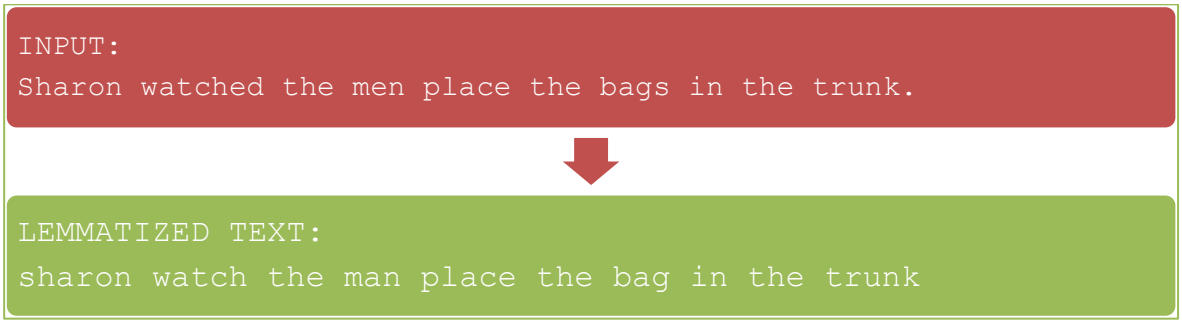
- Lemmatization

In order to obtain each word's base form, the application needs a stemmer or a lemmatizer. We have used the WordNet lemmatizer to fit our system. WordNet lemmatizer only removes affixes if the resulting word is in its dictionary, so it is a good choice to compile the vocabulary of a text and obtain a list of valid lemmas.

However, the application can support other algorithmic processes of determining the lemmas. Advanced users and researchers can use a different lemmatization API, write their own set of lemmatization rules, or choose a stemmer, such as the Porter algorithm.

Regardless the algorithm used, the result of this process is a clean version of the original text obtained by replacing all the words with their base form, so the different inflected forms of a word can be analysed as a single token.

Figure 3. Lemmatization



- Alien word detection

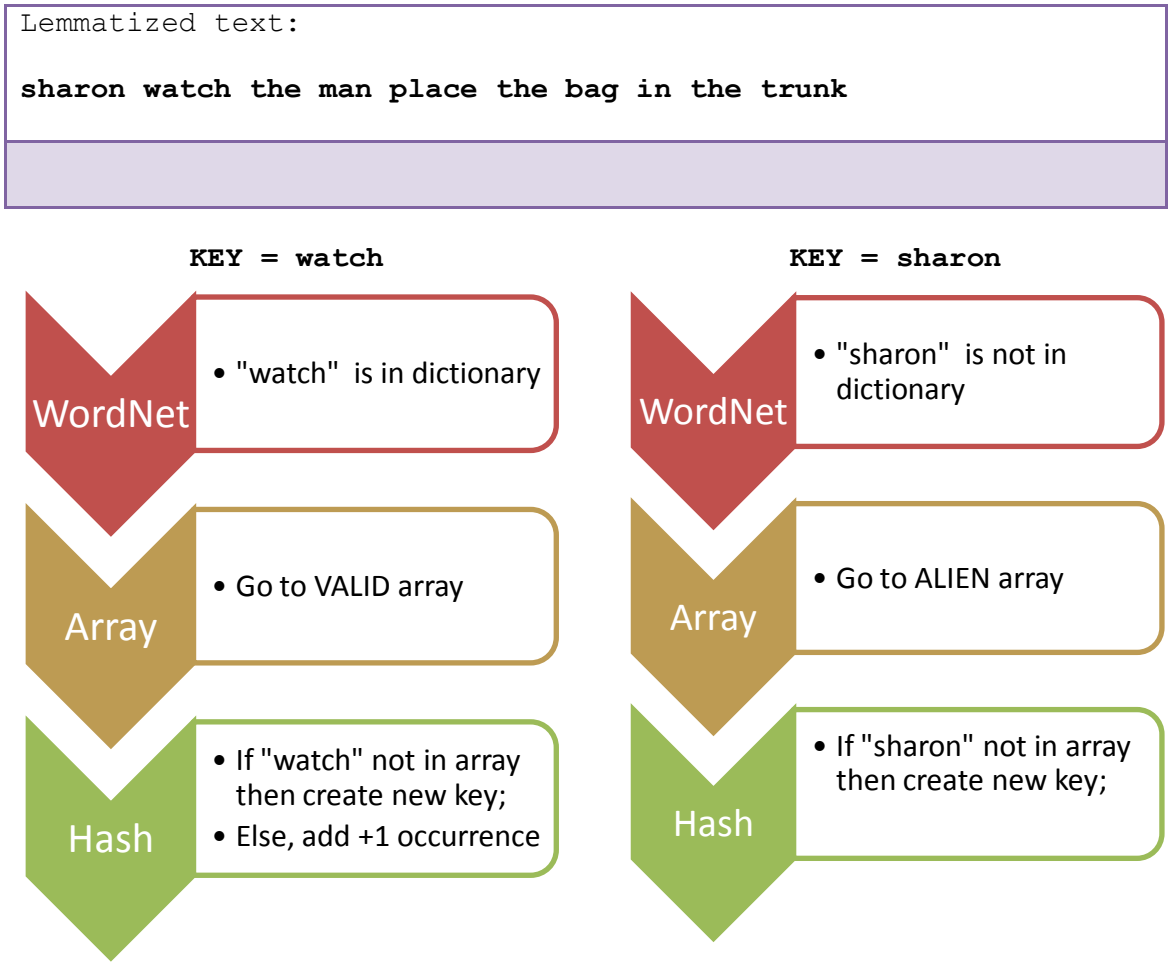
Once the text is lemmatized, each unique token is looked up in a dictionary. By default, this application uses WordNet's lexical database, but advanced users can use any other dictionary API.

Words which are in the dictionary are tagged as "valid" and stored in the VALID associative array; the remaining tokens are tagged as "Alien" and stored in the ALIEN array (See Figure 4).

In computer science, arrays are ordered, indexed collections of objects. An "Associative array" (also known as map) is of a collection of keys and a collection of values, where each key is associated with one value. This application creates associative arrays where the keys or identifying values are all the unique words that appear in the text. The array is implemented with a "hash function", which makes it possible to map two or more keys to the same hash value. In other words, the hash function places repeated occurrences of the

same word in a unique slot which is associated with a value (its frequency index in the BNC).

Figure 4. Lemmatization algorithm diagram



VALID ARRAY	
KEY	OCCURRENCES
the	3
bag	1
in	1
man	1
place	1
trunk	1
watch	**1**

ALIEN ARRAY
KEY
sharon

- Frequency values

Tokens stored in the VALID array are sought in a lemmatized frequency list and are assigned a frequency index. ALIEN words do not require frequency values as they will be ignored in the statistical analysis of frequency. This application uses a modification of Kilgarrif's list based on the BNC, but advanced users can use lists obtained from different corpora.

Figure 5. Frequency values algorithm

BNC FREQUENCY LIST

FREQ. INDEX	WORD	CORPUS OCCUR.
514	history	20064
515	parent	20060
516	land	20001
517	trade	19928
518	**watch**	**19869**

VALID ARRAY

KEY	OCCURRENCES	FREQ. INDEX
the	3	1
in	1	6
man	1	101
place	1	184
watch	**1**	**518**
bag	1	1389
trunk	1	5500

- Statistical treatment of data

Firstly, the Frequency Class of each token is obtained from the following formula:

$$N = \left\lfloor -\log_2 \left(\frac{Frequency\ index\ of\ the\ most\ common\ item}{Frequency\ index\ of\ this\ item} \right) \right\rfloor$$

Secondly, Frequency Classes are adjusted to the number of levels the user wants to have, observing Zimpf's Law of distribution of words in a corpus[3]. Classes 1 to 5 are always merged, since they have a very small number of elements.

[3] Table 4 verifies the prediction stated in Zimpf's Law: each element in the corpus should appear approximately twice as often as the next frequent element, therefore each Frequency Class should have approximately twice as many elements as the next class.

By default, the application has 7 levels; in absolute numbers the correspondence is:

Table 4. Levels and correspondence to Frequency Classes with raw values

Level	Frequency Class	Required words
1	1 - 5	101
2	6	192
3	7	411
4	8	843
5	9	1583
6	10	2685
7	11	4252

Thirdly, the levels can be adjusted by rounding the number of required words, so that the analysis presented to the user is clear. By rounding, the output can be "In order to understand 85% of this text, the student needs a 800-word vocabulary", instead of "a 843-word vocabulary".

By default, the program rounds the levels to the following values:

Table 5. Levels and correspondence to Frequency Classes with rounded values

Level	Approx. Frequency Class	Required words
1	1 - 5	100
2	6	200
3	7	400
4	8	800
5	9	1600
6	10	3200
7	11	6400

Once the levels are set, each token in the array is assigned a level. Then, the percentage of unique words in each rank is calculated. (See Figure 6 below)

Figure 6. VALID array and distribution

INPUT: Sharon watched the men place the bags in the trunk[4]

VALID ARRAY					DISTRIBUTION	
KEY	OCCURRENCES	FREQ. INDEX	LEVEL		LEVEL	%
the	3	1	1		1	28,6 %
in	1	6	1		2	28,6 %
man	1	101	2		3	14,3 %
place	1	184	2		4	14,3 %
watch	1	518	3		5	0 %
bag	1	1389	4		6	14,3 %
trunk	1	5500	6		7	0 %

In order to obtain the level of difficulty of the text, the application compares the resulting percentage in each rank and the expected frequency distributions of rank data, in which the relative frequency of the nth-ranked item is given by the Zeta distribution,

$$\frac{1}{ns\ \zeta(s)}$$

where the parameter s > 1 indexes the members of this family of probability distributions.

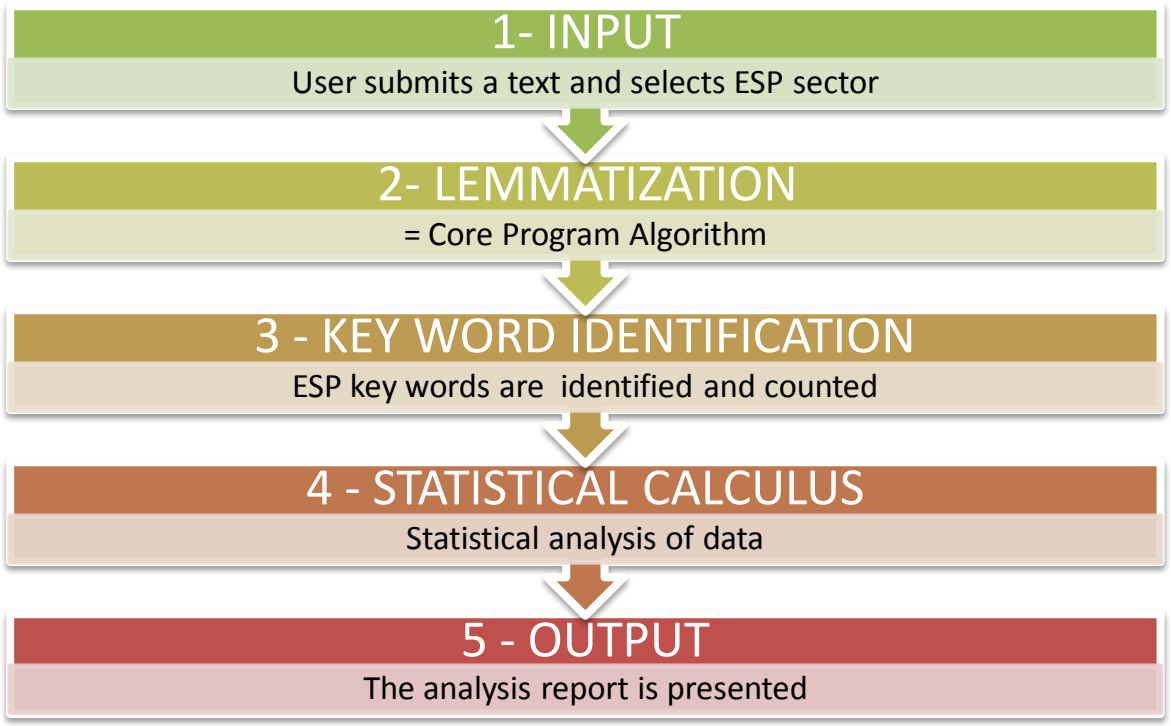
- Output

The results are presented in HTML. The text and the analysis report are stored in the user's database so that the history of work can be accessed and modified in the future.

[4] This example seeks to illustrate the different algorithms performed by the computer. In order to make all the process human-readable, the input is a short 10-word sentence. However, according to probability theory (Feller, 1963), the frequency distribution of such a small sample is likely to be far from the expected values. The study of large texts is presented in the Analysis section.

4.2 Algorithm of the ESP Module

Figure 7. ESP Module algorithm



4.2.1 Linguistic justification of the ESP Module algorithm

- Input

The user submits a text and manually selects the ESP sector to which the text belongs. The user can choose to combine 2 different lists, such as "Economical English" and "Legal English", or "Legal – Easy" and "Legal – Difficult".

- Lemmatization

This process has been explained in the Core Program algorithm. However, this module does not run "Alien words detection" in order to avoid errors of commission related to

technical vocabulary: the application could tag as Alien some words belonging to the target vocabulary of some sectors, such as neologisms, abbreviations or acronyms which might not appear in the dictionary.

- Key word identification

The text is cross checked with the selected list(s) of ESP target vocabulary. All the tokens in the text which are found in the list(s) are tagged as KEYWORD, the remaining words are tagged as GENERAL.

- Statistical treatment of data

The objective is to obtain the "keyword density" (δ) of target vocabulary in the text; in this context, δ is defined as the amount of keywords per text size. Since the ultimate objective is to analyze whether the text is appropriate for ESP vocabulary learning, the application does not take into consideration repeated occurrences of a either a single word or the various inflected forms of the same headword. Therefore, the statistical operations return a density level (low, medium or high) that relates to the number of different keywords in relation to the number of different total words.

- Output

The analysis report presents the following data

- Keyword density level
- Absolute number and percentage of keywords
- Absolute number and percentage of general words
- Word count
- Unique word count

4.2.2 Technical processes and statistical operations of the ESP Module algorithm

- Input

The user submits a text and manually selects the specialised list (or combination of lists) related to the ESP sector to which the text belongs.

In order to analyse the density of words belonging to a specific sector, the application needs to load at least one list of related keywords. These lists must be either preinstalled in the server or submitted by the user.

Basic users can easily create new lists or add new terms to existing lists. In order to generate a new list, the user uses creates a plain text file (.txt) using any word processor and writes one keyword per line, as shown in Table 6. It is highly recommended to write the base form of each keyword, as the lemmatization of input texts could cause errors by omission. Once uploaded to the server, the user will be able to update the lists and add new words in future sessions.

Table 6. Sample of specialised list

File: EconomyKeywords.txt
[…] capital carrier cash CEO change check CIF circulate claim clean clear client […]

- Lemmatization

The algorithm is the same as in the Core Software.

- Keyword identification

Lemmas are sought in the specialised list(s) selected. Similarly to the "Alien word detection" process in the Core Software, tokens are tagged as KEYWORD if they are in the list(s), or GENERAL - for "general vocabulary"- if they are not. Then, two associative arrays named KEYWORD and GENERAL are created. Each tokens - together its number of occurrences - is stored in its correspondent category. (See Figure 8)

- Statistical treatment of data

The Density of target vocabulary in the text is obtained from the percentage of unique terms which are keywords.

We can assume that any list of ESP target vocabulary will exclusively consist of content words - as opposed to function words, such as particles, pronouns, determiners, and so on. However, most function words are high-frequency tokens which, in addition, represent a significant percentage of the total number words in any text (Chung, C. & Pennebaker, J., 2007). Therefore, the best choice seems to be to discriminate function words before calculating the proportion of keywords in the text.

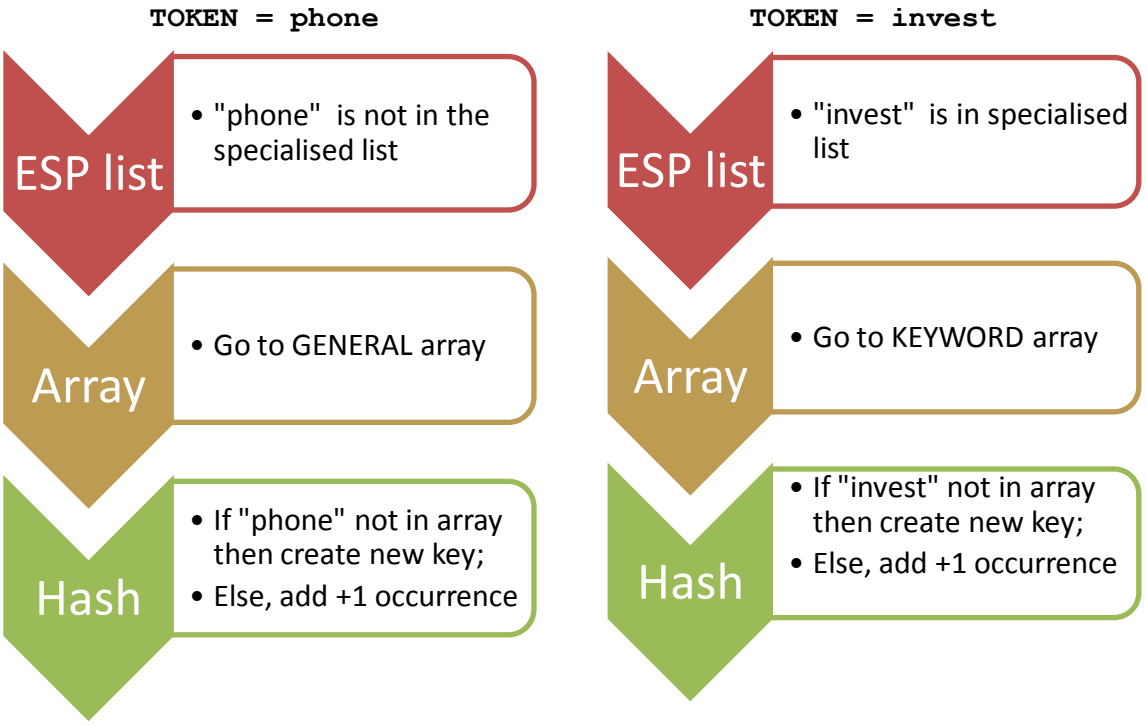
Figure 8. Diagram of the Keyword identification algorithm

INPUT:

The Mexican phone company rose to its highest level in a month on speculation that Congress may ease a limit on foreign investment in the telecommunications industry.

Lemmatized text:

the mexican phone company rise to its high level in a month on speculate that congress may ease a limit on foreign invest in the telecommunication industry



GENERAL ARRAY	
WORD	OCCURRENCES
the	2
mexican	1
phone	1
to	1
its	1
[…]	

KEYWORD ARRAY	
WORD	OCCURRENCES
company	1
rise	1
speculate	1
invest	1
industry	1
[…]	

This software does not support part-of-speech (POS) tagging, but is able to work with the statistical frequency distribution of both function and content words observed in large corpora and Zimpf's Law. Consequently, the ESP Module fits in its system some of the algorithms defined in the Core Software in order to obtain a statistical estimation of the ratio of tokens which are likely to be content words.

Finally, the Keyword Density is calculated from the actual number of elements in the KEYWORD array in relation to the estimated total number of content words.

In order to assign the Keyword Density the values "low", "medium" or "high", the application needs to have some reference values set. By default, this module has the following settings, which can be modified by the user:

Table 7. Keyword density and default associated levels

Keyword Density	Associated level
$\delta \leq 10\%$	"LOW"
$10\% < \delta \leq 20\%$	"MEDIUM"
$\delta > 20\%$	"HIGH"

Users must be cautious in interpreting some data in the final report, since the possibility of working with user-made lists might alter the scale of the reference values. Several factors must be taken into consideration, especially the number of terms in the list and the level of exclusivity of the words. Very extensive lists – or combinations of several lists -, containing a significant amount of words which are also used in non-technical contexts can identify as "high density" some texts which do not meet the teacher's needs.

For this reason, the final report presents the values "low", "medium" or "high" accompanied by the absolute number and the distribution of keywords.

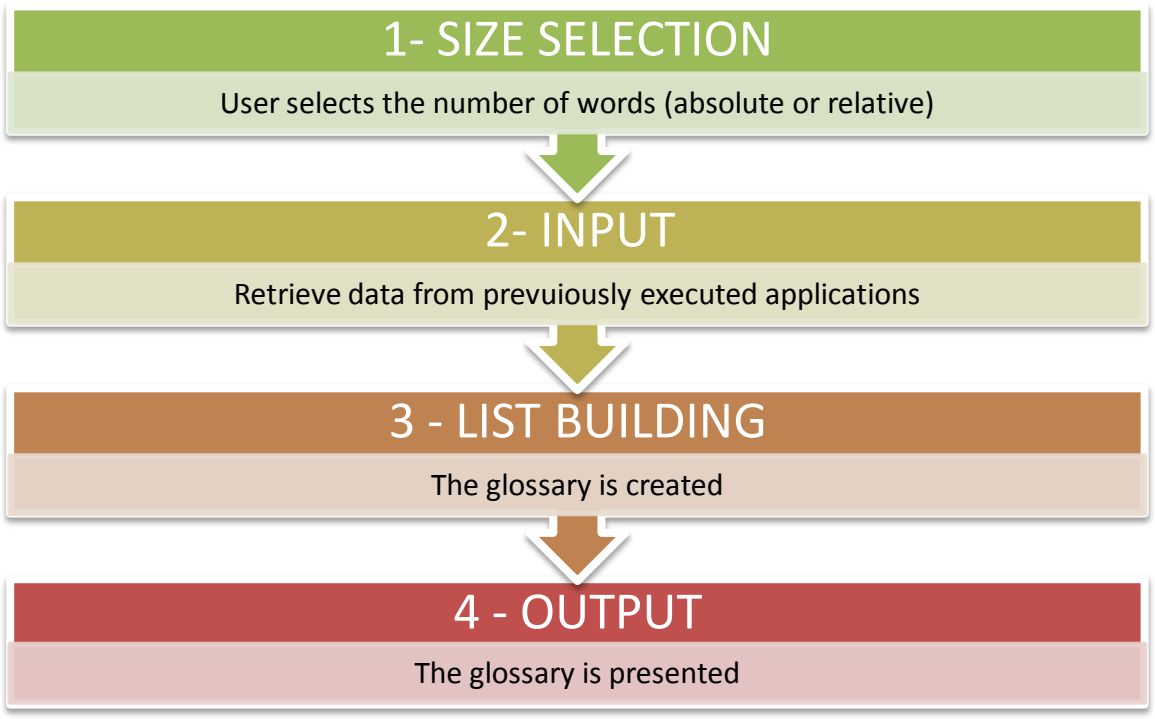
- Output

The results are presented in HTML. The text, analysis report, associated specialized list and user-defined settings are stored in the user's database.

4.3 Algorithm of the Automated Glossary Builder complement

The complements are not autonomous applications; they work on the data obtained from a text by either the Core Software or a module.

Figure 9: Glossary Builder Algorithm



- Size selection

Users can select the number of elements in the glossary. The application supports four types of variables:

- Absolute numbers (e.g., 20 words)
- Percentages (e.g., 5% most difficult words)
- Ranks (words which have a difficulty index over “6”).
- Specialised list (all the Medical terms in the text).

- Input

This Automated Glossary Builder retrieves the data stored in the associative arrays of a previously analysed text, which are created by the hash function of either the Core Software or the ESP Module.

- List building

- General English texts:

The aim of this complement is to help teachers create a glossary of the predictably most difficult words in the text; therefore, the application will look for low-frequency words. In order to do so, the VALID array is sorted by frequency and its tokens are copied into a new list until the number of elements is equal to the size selected by the user in Step 1.

- ESP texts:

If the Glossary Builder works on a technical text processed by the ESP Module, the glossary will contain all the words in the text which have been tagged as keywords. In order to do so, the application lists all the elements stored in the KEYWORD array.

- Output

The glossary is listed in alphabetical order. It is presented in HTML, so it can be copied and edited in a word processor. The glossary is stored in the user's database.

5. Results

In this section, several sample texts of the levelled Penguin Readers series, a Cambridge Proficiency examination and an article published in economical press, are analysed in order to answer the following research questions: 1) Does the lexical competence estimated by the software differ from Penguin and Cambrigde ESOL's levelling systems? 2) Does the software succeed in classifying an economical article? 3) Is the glossary builder useful for assisting the teacher produce reading-aid material for ESL students? 4) Is the software useful to help ESL teachers select texts which are not specifically adapted for ESL students?

5.1 Results of the Core Software

Table 7 below shows the books analysed and their level according to the publisher's criteria. Column three includes the expected lexicon related to each level as described in the Penguin Readers Guideline.

Table 8: Penguin books and classification

Title	Penguin Level	Number of words
Carnival	Easystarts	200
The Adventures of Tom Sawyer	Beginner	300
Treasure Island	Elementary	600
How to be an Alien	Pre-Intermediate	1200
As Time Goes by	Intermediate	1700
The Pelican Brief	Upper Intermediate	2300

The following tables shows the results of some of the internal processes carried out by the software before the final output. They provide the following information:

Rows:

- **Rank**: the category to which each word belongs depending of its frequency index. This analysis uses the default settings; that is, ranks are 100, 200, 400, 800, 1600, 3200 and 6400.

Columns:

- **Words**: number of valid, unique words in that rank.
- **Actual %:** percentage of unique words in that rank.
- **Expected %:** percentage of words expected in each rank, according to Zimpf's Law.
- **Difference:** Difference between the expected % and the actual %.

Some charts have been created in order to illustrate the data presented in the tables. The charts present the distribution of words by frequency ranks and its relation to a standard text. However, more than 50% of the words in any text are likely to be in ranks 1 and 2, so the variations in higher ranks are difficult to interpret from a graphical representation of raw data. In order to illustrate this variations more clearly, we have made a second series of charts which show the difference between the text word distribution and the standard distribution.

Analysis of *Carnival* (Easystarts = 200 words)

Table 9. Analysis report of *Carnival*

Word count	Unique words	Alien words	Valid words
877	291	12	279

TEXT LEVEL: 2.

A lexical competence of 200 words is required to understand at least 85% of the text.

	Words	Actual %	Expected %	Difference
Rank 1	172	61,7	56,6	5,1
Rank 2	34	11,8	7,4	4,3
Rank 3	41	14,7	7,7	7
Rank 4	16	5,9	8,3	-2,4
Rank 5	16	5,9	8,0	-2,2
Rank 6	0	0	7,0	-7
Rank 7	0	0	4,9	-4,9
Total	**279**			

Figure 10. Difference between actual and standard word distribution in *Carnival*

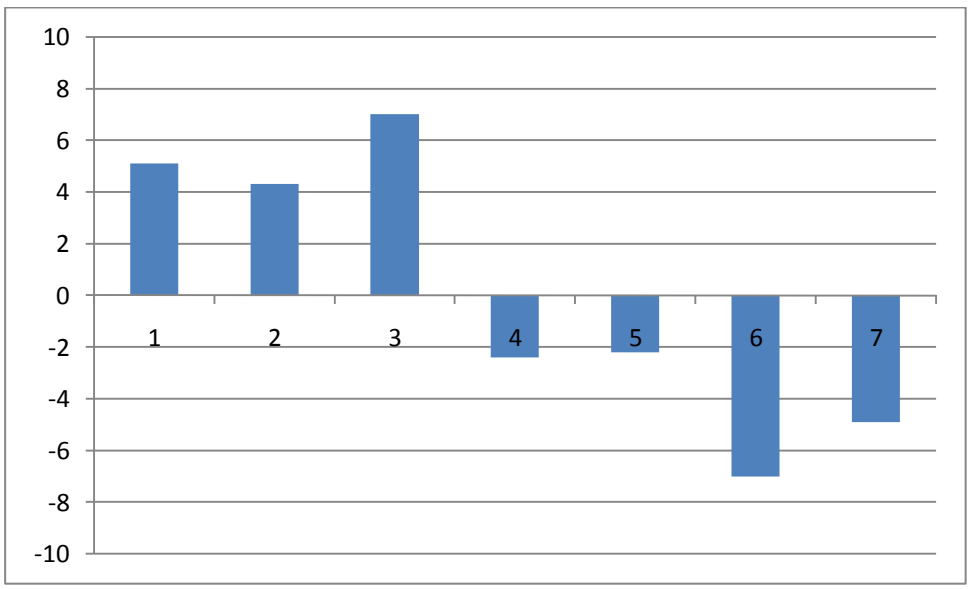
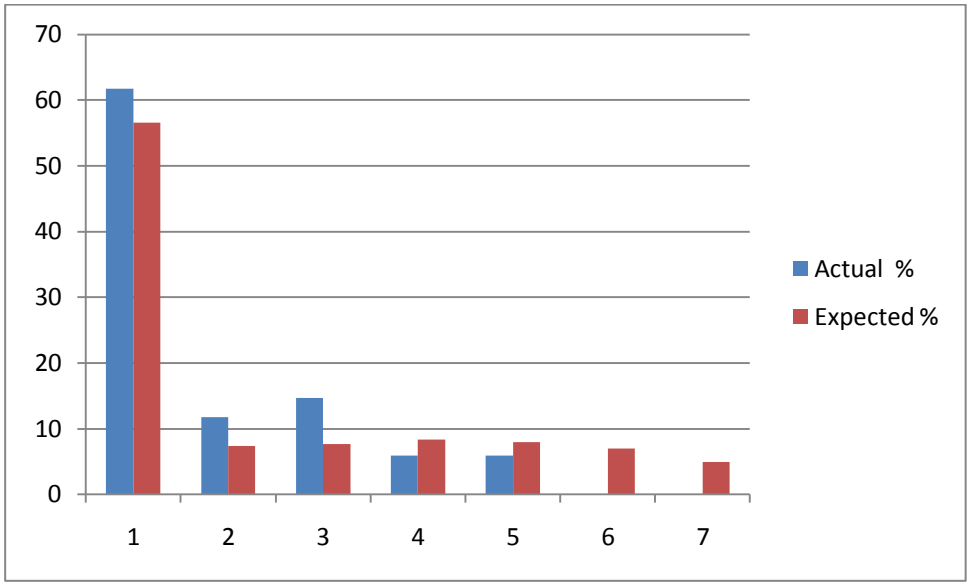


Figure 11. Word distribution in *Carnival*



Analysis of *Tom Sawyer* (Beginners = 300 words)

Table 10. Analysis report of *Tom Sawyer*

Word count	Unique words	Alien words	Valid words
3646	410	15	395

TEXT LEVEL: 3.

A lexical competence of 200-400 words is required to understand at least 85% of the text.

	Words	Actual %	Expected %	Difference
Rank 1	243	61,6	56,6	5
Rank 2	48	12,0	7,4	4,6
Rank 3	51	13,0	7,7	5,3
Rank 4	32	7,1	8,3	-1,2
Rank 5	24	6,0	8,0	-2
Rank 6	1	0,2	7,0	-6,8
Rank 7	0	0,0	4,9	-4,9
Total	395			

Figure 12. Difference between actual and standard word distribution in *Tom Sawyer*

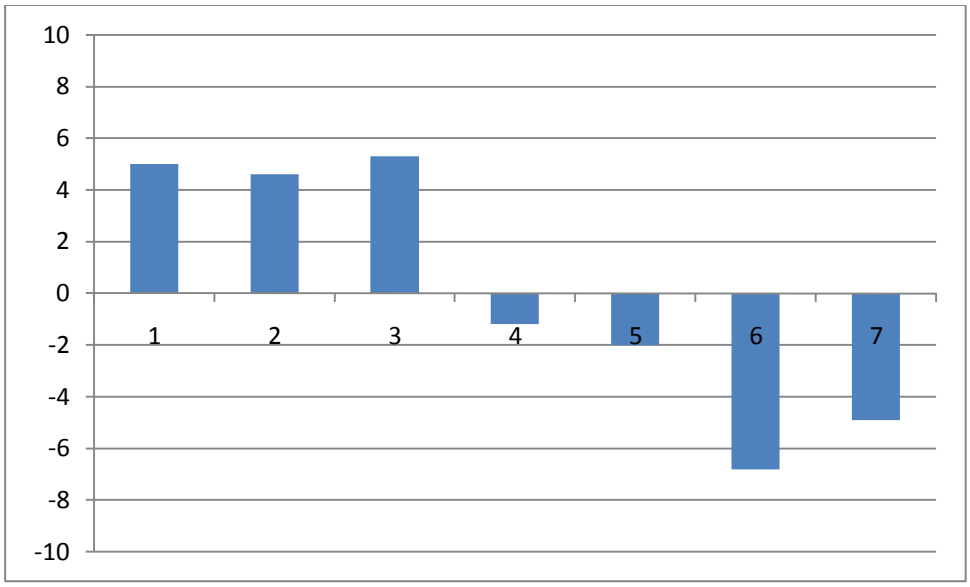
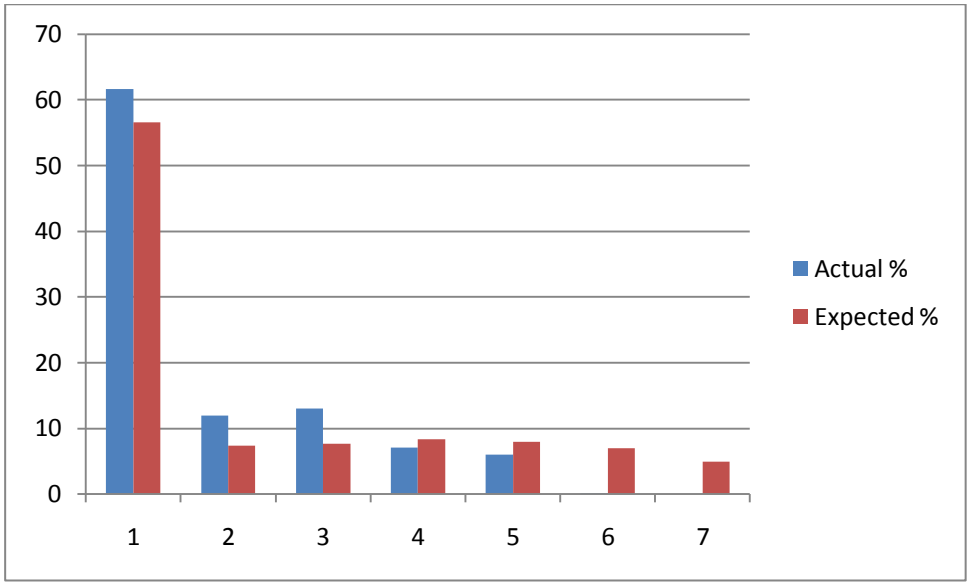


Figure 13. Word distribution in *Tom Sawyer*



Analysis of The Treasure Island (Elementary = 600 words)

Table 11. Analysis report of *The Treasure Island*

Word count	Unique words	Alien words	Valid words
5523	619	21	598

TEXT LEVEL: 4.

A lexical competence of 400-800 words is required to understand at least 85% of the text.

	Words	Actual %	Expected %	Difference
Rank 1	363	60,7	56,6	**4,1**
Rank 2	65	10,9	7,4	**3,5**
Rank 3	68	11,3	7,7	**3,6**
Rank 4	51	8,5	8,3	**0,2**
Rank 5	46	7,6	8,0	**-0,4**
Rank 6	3	0,6	7,0	**-6,4**
Rank 7	2	0,3	4,9	**-4,6**
Total	598			

Figure 14. Difference between actual and standard word distribution in *The Treasure Island*

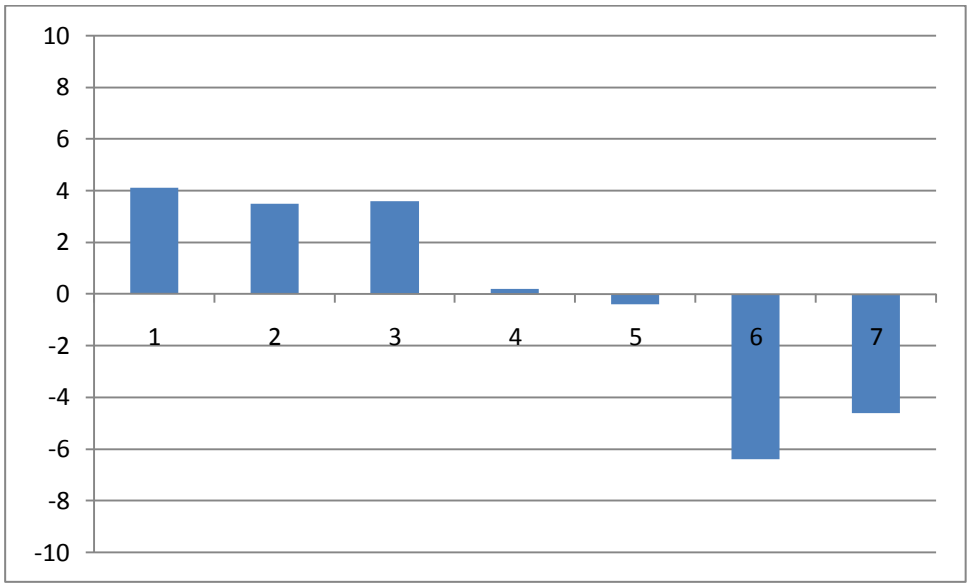
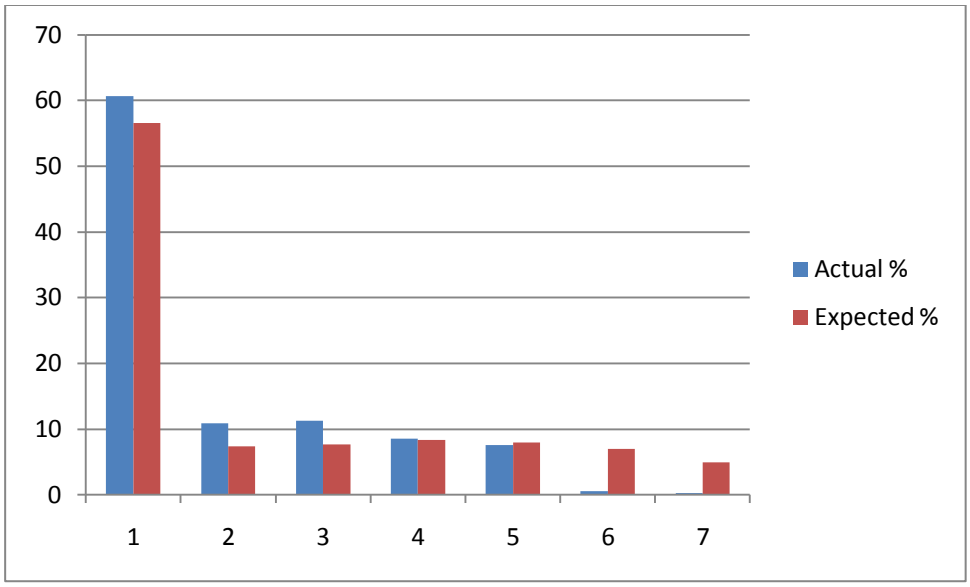


Figure 15. Word distribution in *The Treasure Island*



Analysis of *How to be an Alien* (Pre-Intermediate= 1200 words)

Table 12. Analysis report of *How to be an Alien*

Word count	Unique words	Alien words	Valid words
8818	1284	52	1232

TEXT LEVEL: 5.

A lexical competence of 800-1600 words is required to understand at least 85% of the text.

	Words	Actual %	Expected %	Difference
Rank 1	714	57,9	56,6	**1,3**
Rank 2	141	11,4	7,4	**4**
Rank 3	110	8,9	7,7	**1,2**
Rank 4	83	6,8	8,3	**-1,5**
Rank 5	62	5,0	8,0	**-3**
Rank 6	74	6,0	7,0	**-1**
Rank 7	36	2,9	4,9	**-2**
Total	**1232**			

Figure 16. Difference between actual and standard word distribution in *How to be an Alien*

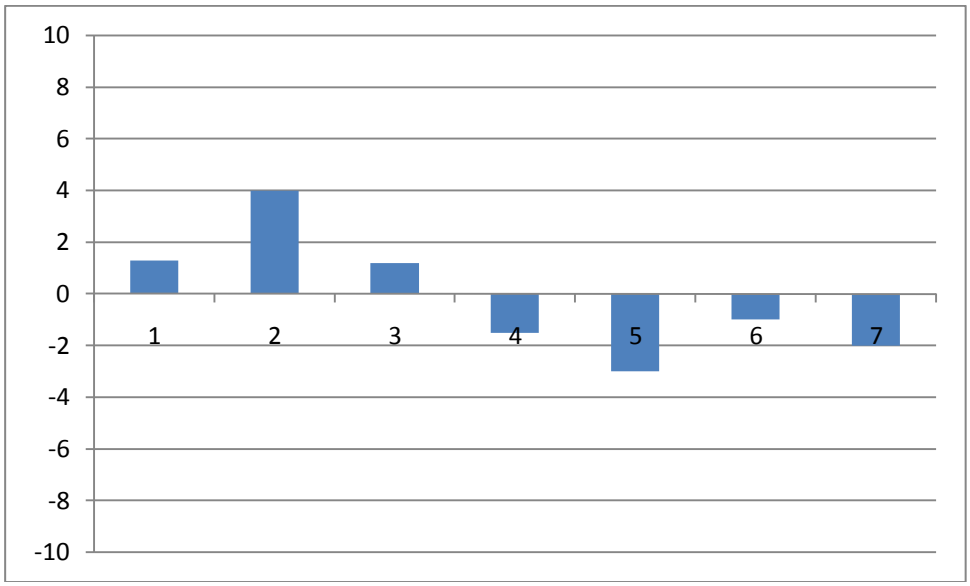
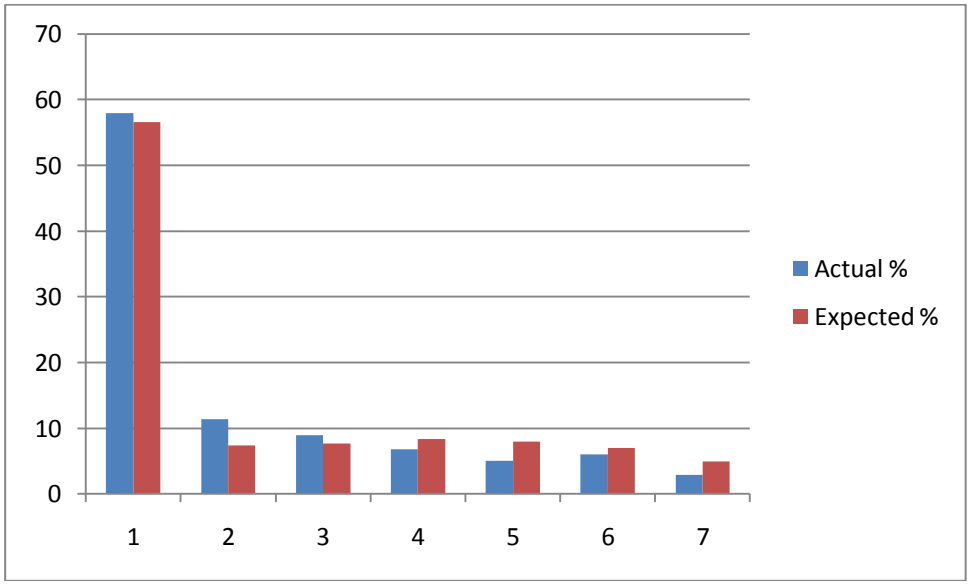


Figure 17. Word distribution in *How to be an Alien*



Analysis of *As Time Goes By* (Intermediate= 1700 words)

Table 13. Analysis report of *As Time Goes By*

Word count	Unique words	Alien words	Valid words
14913	1905	43	1862

TEXT LEVEL: 6.

A lexical competence of 1600-3200 words is required to understand at least 85% of the text.

	Words	Actual %	Expected %	Difference
Rank 1	995	53,4	56,6	**-3,2**
Rank 2	105	5,6	7,4	**-1,8**
Rank 3	153	8,2	7,7	**0,5**
Rank 4	165	8,9	8,3	**0,6**
Rank 5	180	9,6	8,0	**1,6**
Rank 6	156	8,4	7,0	**1,4**
Rank 7	101	5,4	4,9	**0,5**
Total	1862			

Figure 18. Difference between actual and standard word distribution in *As Time Goes By*

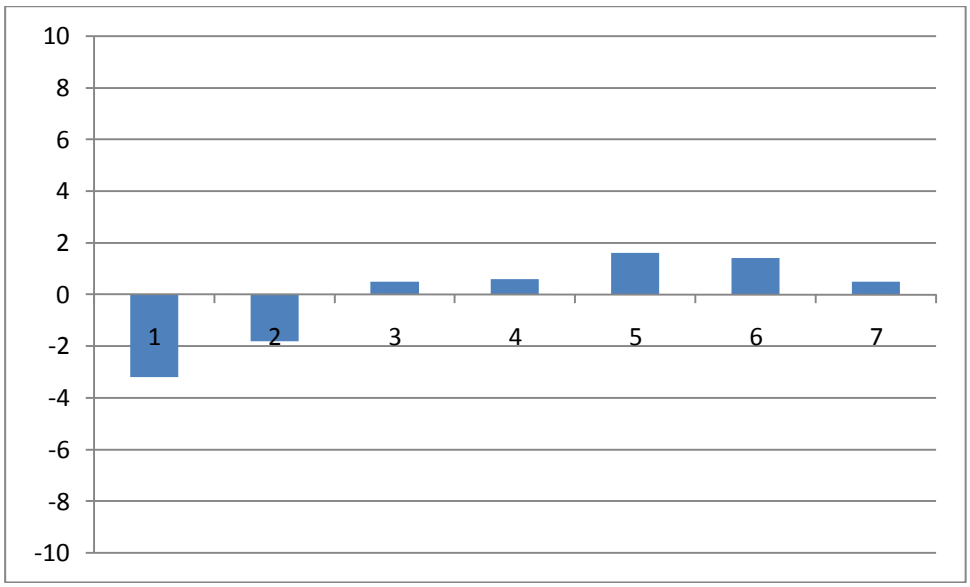
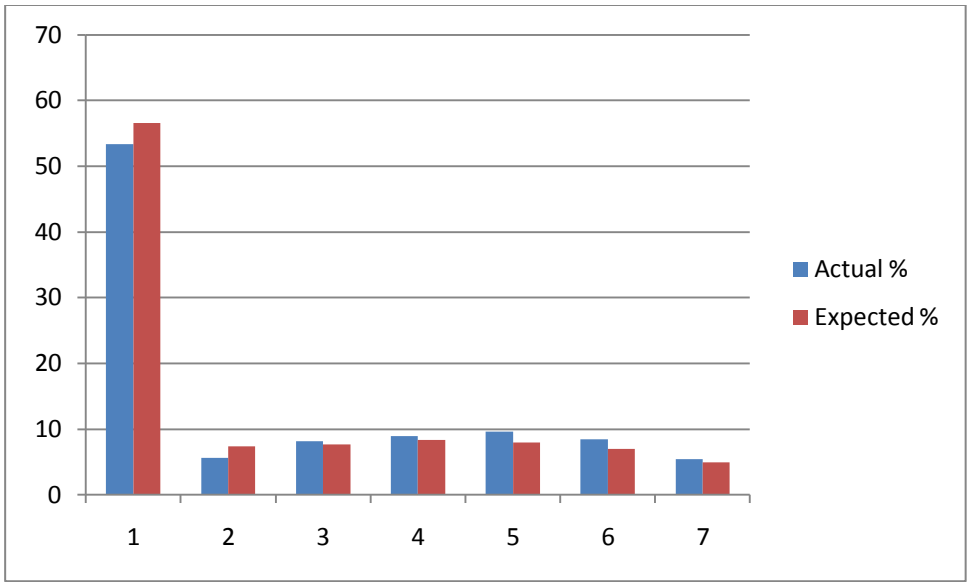


Figure 19. Word distribution in *As Time Goes By*



Analysis of *The Pelican Brief* (Upper Intermediate = 2300 words)

Table 14. Analysis report of *The Pelican Brief*

Word count	Unique words	Alien words	Valid words
15450	1920	62	1858

TEXT LEVEL: 6.

A lexical competence of 1600-3200 words is required to understand at least 85% of the text.

	Words	Actual %	Expected %	Difference
Rank 1	883	47,5	56,6	**-9,1**
Rank 2	200	10,7	7,4	**3,3**
Rank 3	93	5,0	7,7	**-2,7**
Rank 4	165	8,9	8,3	**0,6**
Rank 5	179	9,6	8,0	**1,6**
Rank 6	169	9,1	7,0	**2,1**
Rank 7	170	9,1	4,9	**4,2**
Total	1858			

Figure 20. Difference between actual and standard word distribution in *The Pelican Brief*

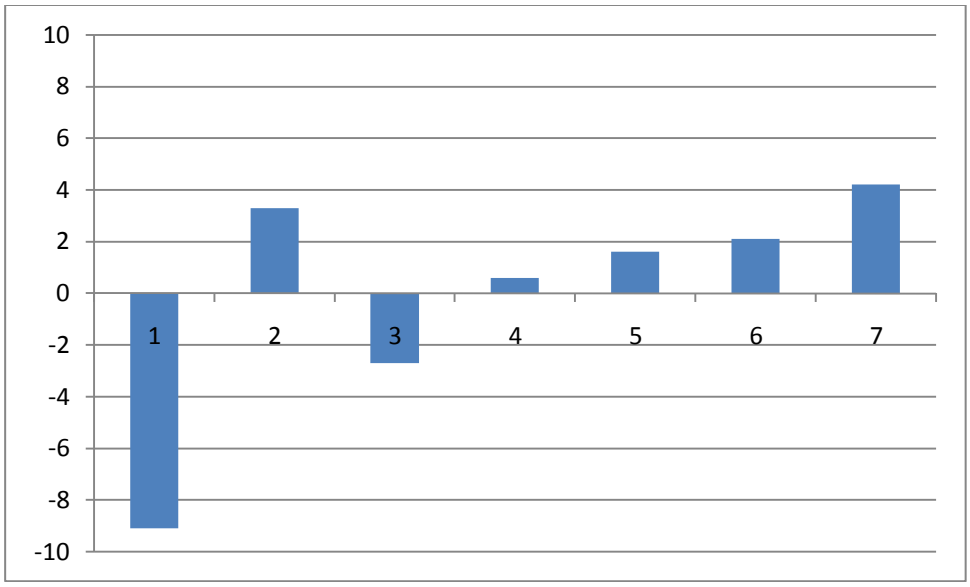
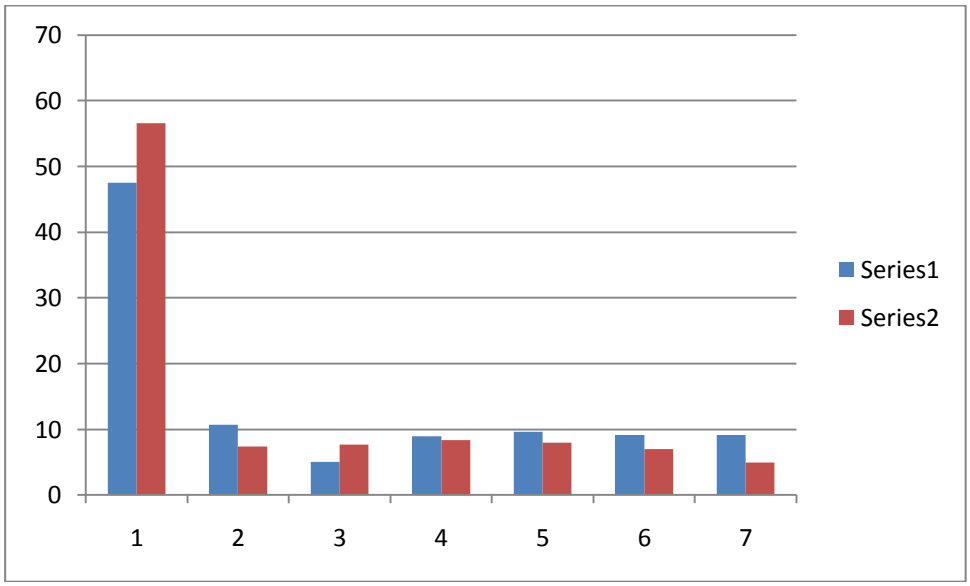


Figure 21. Word distribution in *The Pelican Brief*



Analysis of Proficiency (CPE) Sample examination

Table 15. Analysis report of *CPE examination*

Word count	Unique words	Alien words	Valid words
15450	1920	62	1858

TEXT LEVEL: 7

A lexical competence over 3200 words is required to understand at least 85% of the text.

	Words	Actual %	Expected %	Difference
Rank 1	53	39,4	56,6	**-17,2**
Rank 2	19	14,1	7,4	**6,6**
Rank 3	11	8,5	7,7	**0,7**
Rank 4	17	12,7	8,3	**4,4**
Rank 5	10	7,0	8,0	**-1,0**
Rank 6	8	5,6	7,0	**-1,3**
Rank 7	17	12,7	4,9	**7,7**
Total	135			

Figure 22. Difference between actual and standard word distribution in *CPE examination*

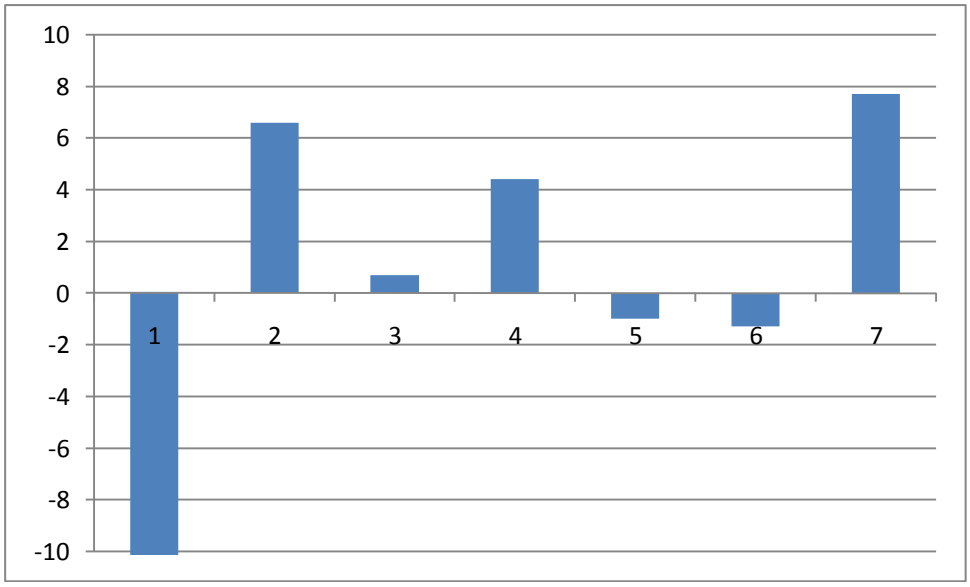


Figure 23. Word distribution in *CPE examination*

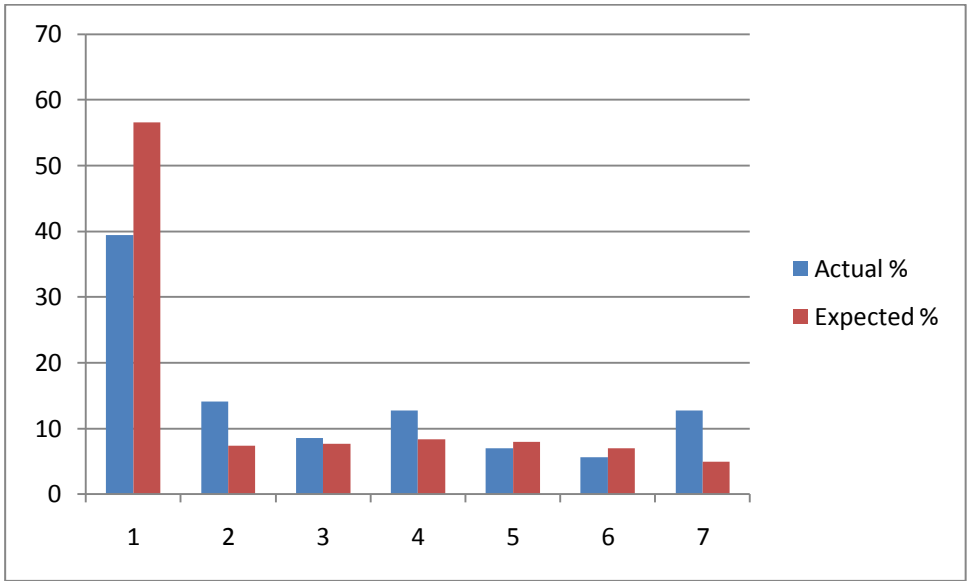


Table 16. Global ranks

	Easystarts	Elementary	Beginners	Pre-Int	Interm	Upper-Int	CPE
Rank 1	61,7	61,6	60,7	57,9	53,4	47,5	39,4
Rank 2	11,8	12	10,9	11,4	5,6	10,7	14,1
Rank 3	14,7	13	11,3	8,9	8,2	5	8,5
Rank 4	5,9	7,1	8,5	6,8	8,9	8,9	12,7
Rank 5	5,9	6	7,6	5	9,6	9,6	7
Rank 6	0	0,2	0,6	6	8,4	9,1	5,6
Rank 7	0	0	0,3	2,9	5,4	9,1	12,7

Table 17. Comparison between revised levelling and automatic levelling

	External criteria		Automatic levelling	
Title	**Level**	**Lexical competence**	**Level**	**Lexical competence**
Carnival	Easystarts	200	2	200
Tom Sawyer	Elementary	300	3	200-400
The Treasure Island	Beginners	600	4	400-800
How to be an Alien	Pre-Int	1200	5	600-1200
As Time Goes By	Interm	1700	6	1200-3200
The Pelican Brief	Upper-Int	2300	6	1200-3200
CPE examination sample	CPE	n.a.	7	>3200

Figures 10 through 22 show the word frequency distribution of each text, and the variations with respect to an ideal text model which would follow the standard frequency distribution described in Zimpf's Law.

It is remarkable that 56% of the expected words belong in rank 1, which implies only 100 words is required to account for half the corpus. From the results observed we can estimate that at least 40% of a text comes from only 1.5% of the corpus.

Regarding differences between levels, *Carnival* shows a significant positive difference in the ranks 1, 2 and 3, and a negative difference in ranks 4, 5, 6 and 7. As the text level increases, the tendency is reversed. The variation of *As Time Goes By* (Intermediate) approaches zero. The CPE examination has the largest difference with respect to the ideal model. In rank 1, there is a negative difference of 17.2 points under the expected value.

As for the unique word count, some texts present results that may seem contradictory with the information provided by the Publisher. According to Penguin, Carnival contains a maximum of 220 words[5] but the analysis reports 279. Similarly, *As Time Goes By*, *How to be an Alien* and *The Pelican Brief* present a variation between 30 and 90 words over the publisher's reference. This can be due to several reasons:

- The lemmatiser failed to group together different inflected forms of the same word
- The software failed to identify hyphenated words as units (e.g. T-shirt)

[5] There are twenty extra words, in addition to the 200 headwords,used in each Easystarts title (Penguin Readers Guideline)

- The publisher's information about the top number of words is an approximation, not a meticulous reference.

Table 17 presents the classification of each book according to the publisher together with the levels assigned by the software. The results show that both levelling systems are coherent even though they use a different division of ranks.

5.2 Results of the ESP Module

Table 18. Economy article analysis report

Word count	Unique words	Key words	General vocabulary
147	85	12	73
DENSITY LEVEL: 3 (HIGH)			
Keywords:			
broker, corporation, credit, financial, liquidate, loan, loss, mortgage, net, profit, revenue, share			

This text was contains 85 unique words. The software predicts that, at least, 40% of them are function words and are ignored. The proportion of key words in relation to the estimated number of content words is δ =22,6%. The default settings return a HIGH density if $\delta > 20\%$

The key words were correctly identified and listed.

5.3 Results of the Glossary Builder Complement

Table 19. Glossary Builder analysis report

20-word glossary as listed in the original work[6]
carnival, crowd, map, bench, balloon, camera, costume, float, drum, procession, feather, band, shout, tourist, wife, wave, T-shirt, point, sergeant, surprised
Software Glossary – Automatic extraction
carnival, crowd, map, bench, balloon, camera, costume, float, drum, procession, feather, tourist, wife, wave, point, policeman, policewoman, sergeant, shirt, surprised
Differences:
Not included: bench, point, T-shirt **New words:** shirt, policeman, policewoman

The Software Glossary lists 85% of the words in Penguin's glossary. The differences are due to the limitations of the lemmatiser in processing hyphenated and complex words. Hyphenated combinations of words are split into separate tokens; therefore "T-shirt" is processed as "T" – which is ignored – and "shirt". On the contrary, the software can not discriminate the different elements in complex words; therefore, "policemen" and "policewoman" are considered vocabulary units with a low frequency index.

[6] Every Penguin Readers' Eastystarts book includes a 20-word glossary and an Activities section.

6. Conclusion

6.1 Overview

The above discussion has evaluated the effectiveness of the algorithms of the three main functions of the program. Regarding the Core Software, the application succeeded in grading the texts according to Penguin's classification and the expected lexical competence required for a CPE examination. As for the ESP Module, the NY Times article was evaluated as appropriate material, as expected. Regarding Glossary Builder, we can conclude that it gives accurate results and it is useful for obtaining preliminary lists of predictably difficult words, but human involvement is needed in order to eliminate unnecessary items.

The user interface presented in this program effectively addresses the complexities of computational linguistic analysis for basic computer users. Simply by submitting an input text and appropriate options as required, the user can carry out a comprehensive speed analysis of a particular text and view the results with several options to save and export the analysis for further work.

One of the weaknesses of our approach is that easy vocabulary does not necessarily mean that the text is comprehensible; other factors such as syntactic complexity must be taken into consideration. Although the results might suggest that texts with basic vocabulary also

contain simple grammatical structures, we do not have enough evidence to support this statement. On the contrary, difficult vocabulary is a reliable indicator of text difficulty. Thus, we can assume that the software's negative evaluation of a text is more reliable than a positive judgement.

Considering this, results suggest that that:

- The software can provide ESL/ESP teachers with relevant and precise information about a text before reading it.
- By shortening the list of candidates, the investment of time in reading and determining the best choice is highly reduced.
- With this software, commercial publishers' evaluation is not required in order to obtain levelled texts, so this enables teachers to choose among a great variety of texts, and work with authentic and up-to-date materials.

.

6.2 Contributions

- **To target users**

This research has produced a free, open source program that enables teachers not familiar with computational linguistics tools or probability theory to benefit from its possibilities in preliminary material selection. Additionally, expert users can develop and add new modules to the program.

- **To Science**

  The major contributions to computational linguistics are the following:

  - The definition of a framework for software metrics refinement, where texts are compared to an ideal model defined by probability theory.
  - The definition of an algorithm that evaluates proportion of core vocabulary using editable corpora
  - The definition of an algorithm based on the frequency distribution of language.
  - The definition of a framework for measuring proportion of core vocabulary by comparing data with an estimated number of content words

6.3 Further research

The opportunities for improvement proposed are the following:

- Implementation of an algorithm that analyses word clusters
- Development and implementation of a part-of-speech tagger
- Development of a new module which retrieves texts from on-line specialised journals, according to certain parameters introduced by the user.
- Implementation of JAWS, a WordNet API which can retrieve definitions for the Glossary Builder.
- Implementation of Intranet teamwork features with an authentication system that manages user accounts, groups and permissions.
- Implementation of on-line collaborative working environments for virtual communities of teachers.

- Integration with Moodle Course Management System to facilitate its incorporation in a virtual campus.

Bibliography

Abney, S. P. (1989). A Computational Model of Human Parsing. Journal of Psycholinguistic Research. *Journal of Psycholinguistic Research , 18*.

Chung, C. & Pennebaker, J. (2007). The Psychological Functions of Function Words. (K. Feidler, Ed.) *Social Communication* .

Dale, E., and Chall, J.S. (1948). A formula for predicting readability: Instructions. *Educational Research Bulletin , 27*.

Dasa, S. (2006). Readability Modelling and Comparison of One and Two Parametric Fit: A Case Study in Bangla. *Journal of Quantitative Linguistics , 13* (1), 17 - 34 .

Edmonds, P. (2007). *Word Sense Disambiguation: Algorithms and Applications.* NY: Springer.

Feebody, P, and Anderson, R.C. (1983). Effects of Vocabulary Difficulty, Text Cohesion and Schema Availability on Reading Comprehension. *Reading Research Quarterly , 18*.

Feller, W. (1963). Laws of Large Numbers. *An Introduction to Probability Theory and Its Applications , 1, 3rd Ed.*, 228-247.

Flesch, R. (1943). Marks of readable Style: A Study in Adult Education. *Bureau of Publications, Teachers College, Columbia University* .

Fotzl, P. W., Kintsch, W. and Landauer, T. (1998). The Measurement of Textual Coherence with Latent Semantic Analysis. *Discourse Processes , 25*.

Gunning, R. (1952). *The Technique of Clear Writing.* New York: McGraw-Hill International Book Co.

Kilgarriff, A. (1997). Putting Frequencies in the Dictionary. *International Journal of Lexicography 10 (2)* , pp. 135-155.

Kintcsh, W. (1974). *The Representation of Meaning in Memory.* Hillsdale, NJ:: Erlbaum.

Lively, B. A. and Pressey, S. L. (1923). A Method for Measuring the 'Vocabulary Burden' of Textbooks". *Educational Administration and Supervision , 9*, 114-123.

Meyer, S. &. (2006). Intrinsic Plagiarism Detection. In Lalmas, M., MacFarlane, A. Rüger, S. Tombros, A. Tsikrika, S. and Yavlinsky, A. (ed.). In *Advances in Information Retrieval: Proceedings of the 28th European Conference on IR Research (ECIR 06)* (pp. 565-569). London: Springer.

Miller, J. R., and Kintsch, W. (1980). Readability and Recall of Short Prose Passages: A Theoretical Analysis. *ournal of Experimental Psychology: Human Learning and Memory , 6*, 335–354.

Wetz, B. (2003a). *Oxford Exchange 1 - SB.* Barcelona: OUP España.

Wetz, B. (2003b). *Oxford Exchange 2 - SB.* Barcelona: OUP España.

Wetz, B. (2003c). *Oxford Exchange 3 - SB.* Barcelona: OUP España.

Wetz, B. (2003d). *Oxford Exchange 4 - SB.* Barcelona: OUP España.

Zipf, G. (1935). *The Psycho-Biology of Language; an Introduction to Dynamic Philology*. Boston: Houghton-Mifflin.

Web resources

CPE Reading Sample Paper. (n.d.). Retrieved July 12, 2010, from University of Cambridge ESOL examinations: http://www.candidates.cambridgeesol.org/cs/digitalAssets/122565_CPEReading_Final.pdf

Henriques, D. (2008). *Loss Shrinks at E-Trade Financial.* Retrieved August 2, 2010, from NYTimes.com: http://query.nytimes.com/gst/fullpage.html?res=9B0DE6D6133BF930A15754C0A96E9C8B63&ref=etrade_financial_corporation

Johnson, P. M. (2005). *A Glossary of Political Economy Terms*. Retrieved June 12, 2010, from Auburn University Website: http://www.auburn.edu/~johnspm/gloss/

Kilgarriff, A. (2006). *Lemmatised BNC Frequency List* . Retrieved July 14, 2010, from Adam Kilgarriff Home Page: http://www.kilgarriff.co.uk/BNClists/lemma.al

Penguin Readers Guideline. (n.d.). Retrieved July 6, 2010, from Pearson Longman Web Site: http://www.longman.de/downloads/GRGuidelines.pdf

Pereda, R. (2003). *Porter Stemmer Algorithm enconded in Ruby.* Retrieved July 10, 2010, from The Porter Stemming Algorithm Official Homepage: http://tartarus.org/~martin/PorterStemmer/ruby.txt

APPENDIX 1. Sample texts.

Excerpt from the analysed texts sorted in ascending level of difficulty.

Carnival
Jake arrives at Euston station in London. It is a holiday weekend and it is his first time away from Manchester. Jake is eighteen years old and he lives with his family. Now he is in London. He is very happy. He stops and looks at his map. 'I can go to the Notting Hill Carnival and I can see some interesting places from the bus too,' he thinks. Jake is sitting on a red London bus behind a big family. The children are standing at the windows. They are looking for famous places. 'Look! There's Madame Tussaud's! Can we go there?' 'Not today,' their mother answers. 'We're going to the carnival.' 'They're going to Notting Hill too,' Jake thinks.

The Adventures of Huckleberry Finn
Becky wanted to talk to Tom, but he didn't look at her. Then Tom talked to Amy. Becky watched him and she was angry. She said to her friends, "I'm going to have an adventure day. You can come on my adventure." But she didn't ask Tom. Later in the morning, Tom talked to Amy again. Becky talked to her friend Alfred and looked at a picture-book with him. Tom watched them and he was angry with Becky. In the afternoon, Tom waited for Becky at the school fence. He said, "I'm sorry." But Becky didn't listen to him. She walked into the school room. The teacher's new book was on his table. This book wasn't for children, but Becky wanted to look at it. She opened the book quietly and looked at the pictures. Suddenly, Tom came into the room. Becky was surprised. She closed the book quickly, and it tore. Becky was angry with Tom and quickly went out of the room. Then the children and the teacher came into the room and went to their places. The teacher looked at his book.

The Treasure Isle

Everybody ran to see the island. I waited for a minute, then I climbed out of the barrel and ran, too. The ship was now quite near an island.
'Does anybody know this island?' Captain Smollett asked.
'I do,' said Silver. 'There were a lot of pirates here in the old days. That hill in the centre of the island is called the Spy Glass.'
Then Captain Smollett showed Silver a map of the island. Silver looked at the map very carefully, but it was not Billy Bones's map. It did not show the treasure.
I went to Dr Livesey. 'Can I speak to you please, doctor?' I said.
'What is it, Jim?' he asked.
Then I told the doctor, Mr Trelawney and Captain Smollett about Long John Silver. 'Most of the sailors are pirates,' I said. 'They want to kill us and take the treasure.'
'Thank you, Jim,' said Mr Trelawney. 'And Captain Smollett, you were right. I was wrong. I'm sorry.'
'Silver is a very clever man,' said the doctor. We all liked him.'
'What are we going to do, captain?' asked Mr Trelawney.

How to be an Alien

In England people are rude in a very different way It somebody tells you an untrue story, in Europe you say, 'You are a liar, sir.' In England you just say, 'Oh, is that so?' Or, 'That's quite an unusual story, isn't it?'

A few years ago, when I knew only about ten words of English and used them all wrong, I went for a job. The man who saw me said quietly, 'I'm afraid your English is a bit unusual.' In any European language, this means, 'Kick this man out of the office!'
A hundred years ago, if somebody made the Sultan of Turkey or the Czar of Russia angry, they cut off the person's head immediately But when somebody made the English queen angry, she said, 'We are not amused,' and the English are still, to this day, very proud of their queen for being so rude. Terribly rude things to say are: 'I'm afraid that ...', 'How strange that ...' and Tm sorry, but ...' You must look very serious when you say these things. It is true that sometimes you hear people shout, 'Get out of here!' or 'Shut your big mouth!' or 'Dirty pig!' etc. This is very un-English. Foreigners who lived in England hundreds of years ago probably introduced these things to the English language.

As Time goes by

"Run," Rick told Ilsa, as they reached the airplane. "And when you're inside, tell them to take off. Understand?"
"I won't leave you."
"Run!"

Rick jumped out and fired at the truck. He wanted them to shoot at him, and not at the airplane. He was ten meters away, and the airplane was starting to move. He was almost there, when a bullet hit his left leg. He reached forward. There were fingers touching his. Someone shot at the Germans from inside the airplane. Another bullet hit him on the shoulder . . . and then . . . he was inside, in someone's arms.

The door shut. He lay on the floor, wondering which parts of his body still worked. He looked up. The fear in Ilsa's face had turned to worry, and then happiness.
"Good morning, Mr. Blaine," said Major Miles, as the plane left the ground. "And congratulations."

The Pelican Brief

The Supreme Court is the highest court in the USA. It consists of nine judges, who hear only the most difficult cases in the country - those cases which might actually threaten the Constitution. Judges are appointed to the Supreme Court by the government, so a Republican government will try to get Republican judges appointed and a Democratic government will try to get Democrats appointed. Judges become members of the Supreme Court for life. They can retire if they want, but if not the job ends only with death. Judge Rosenberg was so old that he found it hard to stay awake sometimes, even during trials. He was a liberal, and proud of it. He defended the Indians, the homosexuals, the poor, the blacks, the Mexicans and the Puerto Ricans. Some people loved him for it, but more people hated him.

Throughout the summer there had been the usual number of messages threatening death to the judges of the Supreme Court, and as usual Rosenberg had received more than the others. The FBI had to behave as if the judges really were in danger, although they were threatened year after year and it was very rare for anything to happen.

CPE Sample examination paper

Like many 18th and 19th century composers, Wolfgang Amadeus Mozart spent a large part of his life on the road. During this time, he impulsively poured his unexpurgated thoughts into copious letters home. These are of crucial biographical importance, but their translation is problematic. Mozart had no formal education and wrote in a mixture of German, French and Italian. His grammar and spelling were unruly and his literary efforts idiosyncratic in the highest degree. Although the words themselves are easily decoded with the help of bilingual dictionaries, the real problem lies in the tone and, as Robert Spaethling observes, previous translators have ducked this. He points to the inappropriateness of reading the letters in impeccable grammar, and aims rather to preserve the natural flow and flavour of Mozart's original style.

Spaethling clearly loves words, and linguistic nuance, as much as Mozart did himself. And when the linguistic games are at their most complex, he democratically prints the original alongside the translation so that we can quarrel and do better. The beauty of this work is that now we can see how - casually and seemingly without trying - Mozart parodies the epistolary modes of the day.

Loss at Loss at E-Trade Is Bigger Than Expected (NY Times)

The E-Trade Financial Corporation, an online broker, reported a bigger-than-expected second-quarter loss Tuesday and warned that credit-related troubles could result in more losses.E-Trade posted a net loss of $94.6 million, or 19 cents a share, in contrast to a profit of $159.1 million, or 37 cents a share, a year earlier. Revenue was down 20 percent, to $532.3 million.

E-Trade said it liquidated about 65 percent of $330 million in preferred equity held in the mortgage lenders Fannie Mae and Freddie Mac, a move that will result in an $83 million pretax loss in the third quarter.
It also warned that "the current economic environment may impede our expectations to return to profitability from continuing operations this year."

The loss from continuing operations was $119.4 million in the quarter, in contrast to a profit of $157.7 million in the comparable quarter of 2007.

APPENDIX 2. Educational Community License.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The name and trademarks of copyright holder(s) may NOT be used in advertising or publicity pertaining to the Original or Derivative Works without specific, written prior permission. Title to copyright in the Original Work and any associated documentation will at all times remain with the copyright holders.