Syntactic REAP Exercises on Word Formation

Pedro Henrique Santos Figueirinha

IST – Instituto Superior Técnico
L^{2}F – Spoken Language Systems Laboratory – INESC-ID Lisboa
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
pfigueirinha@l2f.inesc-id.pt

Abstract. The introduction of computers as an assisting tool in the field of education has affected the way teachers plan their classes and interact with their students. The Intelligent Computer Assisted Language Learning (ICALL) field addresses this evolution in the teaching process and has fomented great interest for the last few decades in the academic community. REAP.PT is an ICALL system that aims to teach Portuguese to both foreign and native speakers. This is done by employing a series of automatically generated exercises focusing on several lexical, phonetical, semantic and syntactic aspects of the language, specifically tailored to the users needs. The goal of this work is to expand the REAP.PT system, namely, its syntactic exercises, by introducing two new games focusing on word formation. As it is the case with the already developed exercises, these will not only maintain the REAP.PT visual style and interface, but they will also be automatically generated. The first exercise deals with the task of dividing words into syllables, a process that even tough it may look simple and straight-forward at first sight, requires language-specific rules to be followed and could help improve phonological and orthographical awareness in the student. The second exercise aims at prefixal derivation processes, by asking the student to identify words that, on surface, look like prefixed derived forms but whose meaning can not be regularly derived by said morphological process. Personalized support and feedback is also presented for exercise. The positive evaluation results, consisting of crowd-sourced tests and questionnaires, validates the presented approach.

Keywords: Intelligent Computer Assisted Language Learning, Portuguese Language, Syntactic Exercises, Automatic Exercise Generation

1 Introduction

With the increasing role of technology integration and computer usage in our society and daily life, more particularly in the educational area, language teaching has become an increasingly attainable activity to anyone with the means to use a computer. The way contents from this specific field can be easily accessed has allowed the average user to gradually improve his/her proficiency in both native and foreign languages, given that the variety of available studying materials can cover a potentially limitless number of shortfalls.

From the need to conveniently adapt these resources to the specific needs of the users, the Intelligent Computer Assisted Language Learning (ICALL) research area was born, an academic field devoted to improving and simplifying this learning process.

One of the systems that emerged in this field is the REAP (REAder-specific Practice) project[1]. Initially developed at CMU (Carnegie Mellon University) by LTI (Language Technologies Institute), the main goal of this project is to teach English by using a series of different exercises which focus on vocabulary, while also steadily improving the reading ability of the users. One aspect that sets REAP apart from the existing alternative systems, is the ability to automatically generate exercises on a variety of linguistic aspects.

With this in mind, we can describe the main goal of the REAP.PT[2] project as to port and adapt the original REAP features to the Portuguese language. The first module to be developed by Marujo [10] was the lexical learning component, consisting of the integration of various textual and multimedia resources. After this, a component capable of generating questions automatically was built by Correia [2]. Following these implementations, syntactic exercises were developed and integrated by Marques [9] and Freitas [4] and additional exercises were developed by Silva [14, 16]. The latest improvement to the REAP.PT system, was a complete redesign of its interface, by Filipe [3], giving the entire system, a more homogeneous visual style.

1.1 Goals

The goal of the present work is to expand the work developed by Marques [9] and Freitas [4], by introducing and implementing new exercises while maintaining, several features exhibited by the already developed games, namely, their correctness from a linguistic point of view and the ability to be automatically generated. The resulting output of this operation will be particularly tailored to the user's preferences and their level of proficiency in the Portuguese language.

As it is the case with the exercises already available in the REAP.PT system, the exercises developed in this project maintain the visual style defined by Filipe [3], they support several degrees of difficulty and they offer dynamic feedback, bases on the user's answers.

[1] `http://reap.cs.cmu.edu` (all url were last visited in October 2013)
[2] `http://call.l2f.inesc-id.pt/reap.public/`

2 REAP.PT

REAP.PT Architecture. The architecture of the REAP.PT consists of several components and the primary focus of the system is on reading comprehension and vocabulary exercises. Both teachers and students interact with the system via the Web interface module. The information exchange between the REAP.PT database and the DIXI Server [12] – the listening comprehension module – is also handled by the Web interface. The latter (unique to REAP.PT) provides text-to-speech features to the system. The Daily REAP.PT [16] component is a dataset consisting of texts gathered from Portuguese on-line newspapers and transcripts from television newscasts from the past seven days. These texts have to pass a series of filters [10] before being considered appropriate for educational purposes. One of the filters to be applied has to do with the presence of words from the Portuguese Academic Word List (P-AWL) [1] in the texts. The Question Generation module produces exercises that are presented to the students after they finish reading texts. These exercises consist of automatically generated multiple choice questions [10, 2], all based on the text read by the student. The alternatives to the correct answer in each multiple-choice question (distractors) are also automatically generated and were developed and integrated in the system by Correia [2].

REAP.PT Vocabulary and Syntactic Exercises. REAP.PT currently offers several different vocabulary and syntactic exercises.

The "Lexical Mahjong"[9] exercise consists of establishing a correspondence between words and their definitions, both represented as a series of shuffled tiles.

The focus of the "Collective Names and Nominal Determinants" exercise [9] (a multiple-choice exercise) is on the constraints between determinative nouns and the nouns they determine, for example *um cesto de fruta* (a basket of fruit), and the relations between common nouns and their respective collective nouns, for example *um rebanho de cabras* (a herd of goats).

In the "Choice of Mood in Subordinate Clauses" exercise [9], users are asked select the correct tense, from several alternatives. The objective is to find which verb tense fits the syntactic restrictions that connect the main clause of a given sentence and the subordinate clause where the verb tense to insert is missing. Sentences are extracted from the CETEMPúblico [13] corpus, and later processed by STRING [8], the natural language processing chain of L^2F. The L^2F VerbForms[3] generator is then used to automatically generate the distractors.

Finally, in the "Pronominalization" exercise [4], users are asked to replace a constituent with a pronoun in a given sentence. Sentences without pronouns are selected from the REAP.PT corpus and later processed by STRING in order to identify the complement dependencies between phrases. Distractors are generated according to a series of parameters, involving the incorrect positioning of the pronoun and/or an incorrect identification of the pronoun's case.

[3] `https://www.l2f.inesc-id.pt/wiki/index.php/VerbForms`

3 State of the Art

3.1 ICALL Systems

The automatic generation of exercises is not a feature frequently found in other ICALL systems. WERTI (Working with English real-texts)[4] [11] is a system designed to help English as a second language students overcome their shortfalls. Users are prompted to install a toolbar plugin in their browser (only Firefox is supported), which allows for any web-page to be used as a studying material. Multiple types of exercises are available for a specific set of topics, such as articles, determiners and prepositions. Support for the German and Spanish languages is also available albeit at a Beta stage.

The Alpheios Project[5] is an open-source project focusing on classical literature and languages, such as Latin and ancient Greek. Like WERTI, this project uses a Firefox toolbar plugin that allows the user to perform several operations on selected texts. When these are presented to the user, one of two modes (reading mode or quiz mode) must be chosen by the user. In the first one, not only is an input field made available for looking up the morphological analysis and translation of words but, by double-clicking any word from the text, a window, including the translation of the given word and additional operations, pops-up. In quiz mode, the user is required to double-click any chosen word and is then presented with a different question depending on the language of the text. Regardless of the selected mode, additional resources for the Greek and Latin languages namely, inflection tables and grammars are always available from the toolbar.

3.2 Current syntactic exercises on word formation

On the one hand, traditional exercises focusing on syllable division typically require the student not only to perform said division, but also to analyse the words that are presented to him, according to certain parameters. This process usually revolves around categorizing words according to their number of syllables or according to the position of their tonic syllable.

On the other hand, traditional exercises focusing on the processes used to form new words from other existing words or from root morphemes (stems), usually revolve around the valid application of affixes to words and the identification of the process used to form certain words.

4 Architecture

4.1 YAH

Since both exercises focus on word formation, there was the need to automate the process of dividing words into syllables. Originally, the plan was to use the

[4] `http://sifnos.sfs.uni-tuebingen.de/WERTi/`
[5] `http://alpheios.net/`

syllable-divided vocabulary of P-AWL [1] in both exercises, however, given the size and the scope of the alternative available resources [17] for the Derivation exercise, the inclusion of the words belonging to the P-AWL, in this specific exercise, was dropped. The Crossword Puzzle exercise, on the other hand, only uses words which are present in the P-AWL.

There are different interpretations of how words should typically be divided. Among these, are the processes of dividing words phonetically[6] and orthographically. The latter, which corresponds to the act of hyphenating words, is done according to the rules in the *Acordo Ortográfico da Língua Portuguesa de 1990*[7] (Portuguese Language Orthographic Agreement of 1990). In this exercise, this specific set of rules was followed, albeit with some changes (concerning the division of the digraphs ¡ss¿ and ¡rr¿), in view of the specific goal of the exercises.

During the hyphenation process of words containing one of these digraphs, such as *possibilidade* (possibility), the orthographic result is the splitting of said digraph (*pos-si-bi-li-da-de*). However, if we ask any native speaker how he or she would utter this word, syllable by syllable, since this digraph constitutes a single phone [s], both "s's" would be placed right next to each other in the second syllable, with the transcription of this division being *po-ssi-bi-li-da-de*. Therefore, in view of the phonological goal of the exercise, which is to identify syllables, we consider that this is the best approximation, even though not orthographically correct.

With this in mind, in order to prevent mistakes and to simplify the division process, an hyphenator entitled YAH[8], which stands for "Yet Another Hyphenator", was used as the basis for the task of dividing the P-AWL. This C++ implemented tool, which had already been developed at L^2F, receives a word as its input parameter and produces an hyphenated string as its output. This is done by applying a "mask" to the word, which classifies each letter according to its type (consonant or vowel).

Following this step, the input string is matched against a set of "word beginning rules", which, as the name suggests, concern the first letters of the word. Whenever there is a positive match, the first matching rule is applied. This process repeats itself throughout the hyphenation of the whole word, with the remainder of the applicable (generic) rules. It is important to note that even though the existence of the mask generalizes the division process, it is also possible to create and apply rules depending on a specific sub-string of characters.

After one of these beginning rules is applied or if none of them match the beginning of the word, a set of more generic rules is then used. This is where the main modifications were made, in order to achieve the desired results. The original YAH contained only eight word beginning rules and eight more generic rules. This did not guarantee the correct division of most of the words present in

[6] `http://www.academia.org.br/abl/cgi/cgilua.exe/sys/start.htm?infoid=11762&sid=784`

[7] `http://www.portaldalinguaportuguesa.org/acordo.php?action=acordo&version=1990`

[8] The hyphenator has been built by Ricardo Ribeiro, in 2012.

the P-AWL. The final version of the modified YAH contains sixteen word beginning rules, fifty six "generic" rules and thirteen exceptions that deal with word endings and other peculiar aspects. The latter were implemented to deal with certain syllables which are processed by the existing rules (because a valid mask is applied) but constitute exceptions which should be dealt with accordingly.

Besides not being able to cover all the aspects concerning the hyphenation of Portuguese words, such as the ascending diphthongs (where a semi-vowel precedes a vowel), descending diphthongs (the opposite), and the digraphs "qu","gu", "pn" or "pt", among others, which are all indivisible, the original YAH was not capable of processing words containing letters with diacritical marks, such as "ç" or "á". The latter problem was solved by modifying the way each word was parsed and how the mask was applied. The remaining aspects were corrected by implementing additional rules.

In order to assess the overall quality of these rules, a corpus consisting of 1722 words was created. This was done by extracting the contents of eight random on-line articles from the Público newspaper[9], as well as 126 words taken from the examples of hyphenation rules presented by the the *Acordo Ortográfico da Lígua Portuguesa de 1990*. The modified YAH hyphenator was capable of correctly dividing 97% of this corpus, which corresponds to 1671 correctly dividedwords, as well as 100% of the P-AWL, which corresponds to 2553 correctly divided words. In comparison, the original YAH was only capable of dividing 74,2% of this corpus contents and 63,1% of the P-AWL.

4.2 Crossword Puzzle exercise

In this exercise, the goal is to solve a crossword puzzle grid by dividing words into syllables. Crossword Puzzles are normally filled using individual letters. However, separating words into syllables is also an useful exercise when the objective is to teach both vocabulary and diction since analysing the syllables that compose a word and knowing how to pronounce them separately, makes the task of reading words aloud an easier one. Regarding Portuguese, and this exercise in particular, the modified YAH hyphenator was used to divide the P-AWL in its entirety. The result of this operation is then used as the corpus from where the divided words are extracted and used to automatically build the puzzle. This corpus consists of 570 lemmas and 1983 distinct inflected forms, for a total of 2553 words (most inflected forms were not included).

Word Selection. In the first step, one word is randomly chosen from the previously divided P-AWL, and used as the target vertical word. This is the only word that has to make sense vertically. Since the selected word is already divided into syllables, these are separately used in order to find the matching intersections for the horizontal words. Once again, these are extracted from the P-AWL. To prevent similar words being chosen, the Levenshtein distance [5] (with a minimum value of five) between the vertical word and each of the intersecting words, is calculated. A simple PHP implementation[10] of this function

[9] `http://www.publico.pt/`

[10] `http://php.net/manual/en/function.levenshtein.php`

was used, which means that the deletion, insertion and replacement operations all have the same cost (one).

Grid Generation. After word selection is completed, a fully filled grid is generated. This is used as a basis for the grid which the user will be asked to fill. Using the (previously selected) difficulty level corresponding to the first parameter (grid difficulty level), several transformations are made to this grid. If the user selects the easiest level, the first and last cell of each word are left correctly filled as well as the cells corresponding to the vertical word. If the medium difficulty level is chosen, only the vertical word's cells are left filled in the grid. Finally, if the hard level is chosen, the grid's contents are completely erased, leaving nothing but blank, empty cells.

Definition Formatting. The definitions of the words in P-AWL had been previously extracted using both the *Infopédia*[11] and Priberam[12] on-line dictionaries, with the latter being used for adverbs, since the definitions in the *Infopédia* dictionary were considered unhelpful.

Almost all the definitions provided by the *Infopédia* dictionary are presented in similar fashion, with three distinct elements:

- A modifier stating the part-of-speech, tense, gender and/or number of the lemma;
- A subset of definitions;
- A list of compounds, collocations and idiomatic expressions containing the word.

Since we are only interested in keeping the first two elements, the list of idiomatic expressions was discarded. This process is done by extracting the relevant text from the web-page and delimiting each definition. After this, a regular expression is used to identify the first definition where the word appears. This definition typically corresponds to the first idiomatic expression, from which the remainder of the text can be discarded. Even if that is not the case, the definition is still discarded, because a clue containing the answer is not suitable for the puzzle at hand.

This trimming process, which is done in order to eliminate any unnecessary or redundant information, differs from word to word. This required each word do be analysed by hand. For example, in the modifier of the word *moedas* (coins), which is *"nome feminino plural de moeda"* (the feminine plural of coin), the information regarding the gender and the part of speech is not particularly helpful because in this specific case, it has no impact on how the word is transformed given its base form. The number however, indicates that some changes need to be made, namely, the addition of the *"-s"* suffix. Furthermore, the existence of information that states which word undergoes the modification (*"(...)de moeda"*), defeats the purpose of trying to find the target word using the definitions, since it clearly indicates what the answer is.

[11] `http://www.infopedia.pt`
[12] `http://www.priberam.pt/dlpo/`

The result of this whole process was then stored in the same directory as the exercise. This was done to speed up the lookup process. The second difficulty parameter (definition difficulty level) is then taken into account in order to concatenate the modifier and the most appropriate definition, which is selected depending on their position on the "extracted" list.

Support. At any point during the process of solving the puzzle, the user has access to three different help functionalities. These are accessible through a series of buttons located in the lower part of the interface. The first one, *Colorir (Color)*, changes the grid's default color scheme, painting each cell's border green or red depending on whether its contents (each cell's specific syllable) are correct or wrong. The second help option, called *Dica* (Hint), allows the user to automatically obtain the contents (the correct answer) of a group of cells, by filling the target cells with an 'X' character. The final help feature, called *Mais Definições* (More Definitions), is implemented in a similar fashion to the *Dica* assistance but this time, instead of filling the cells with the correct answer, the complete definitions, for the words to which the cells belong to, are presented in the blackboard.

Score and Feedback. After submitting the puzzle grid, the user is presented with a feedback screen, which includes both the submitted and the solution grids, information on the number of hints used and the number of both correct and incorrect answers. When combined with the difficulty levels selected in the beginning of the exercise, these parameters are used to compute the puzzle's maximum score and the user's score. To discourage an excessive use of the help features, each hint has a negative impact on the final score.

4.3 Derivation exercise

Derivation (id. in English) is one of the processes languages use to form new words from other existing words or from root morphemes (stems). In this exercise, the main focus is to identify words formed through prefixal derivation (that is, derivation by prefix addition to a base form) that constitute exceptions to the general rules of the prefixation process. A common example can be found in the word *correto* (correct). This word has its meaning negated if we apply the prefix *in–* changing it into *incorreto* (incorrect). However, there are certain exceptions to this rule and the application (or presence) of negative prefixes in certain words, or stems, will not always mean that the given word has its meaning negated if said prefix is removed. For example, the word *infiltrado* (infiltrated), does not mean the opposite of *filtrado* (filtered).

With that being the case, both the exceptions and the distractors used in this exercise were the result of the update process of the LexMan [17] dictionaries. LexMan is part of STRING [8] and besides being responsible for the tokenization and morphological analysis steps, it is also capable of identifying prefixed words. For this exercise, the words which were valid combinations of prefixes and lemmas were used as the distractors. Those which were not, were used as the exceptions (target words).

Prefix Selection. The process of updating the LexMan dictionaries yielded 522 distinct files (each corresponding to a different prefix) containing exceptions, and 15328 distractors. Given that there were some prefixes with only a handful of exceptions and/or others that had very few distractors, these 522 files were reduced to 48 (which make up a total of 6623 exceptions). Difficulty levels are attributed to these prefixes, depending on the number of available exceptions and distractors (in total 12997 word forms remained eligible).

Finally, the generation of a puzzle consists of randomly choosing a prefix from the list, depending on the difficulty level selected by the user. Similar to what happens in the Crossword Puzzle exercise, the user is asked to choose between three difficulty levels for two different parameters: the prefix and the exception (target word).

Exception Selection. Further parametrization was required to distinguish between easy exceptions and the more difficult ones. This was done by calculating the frequencies of each exception (uni-gram count), using the total number of forms found in the Linguateca corpora[13]. Since these files are updated frequently, the file[14] corresponding to the totals is also updated. For this reason, the 28 of March 2013 version was stored locally and used for the remainder of this process. Words which did not appear in this list were assigned a count value of "1".

Difficulty level assignment was done in similar fashion to what happened with the prefixes. This is where the second difficulty parameter comes into play. Depending on the user's choice, an exception with the matching difficulty level is randomly extracted from the list of exceptions corresponding to the prefix chosen in the previous stage.

Distractor Selection. In this specific exercise, three different distractors are chosen for each puzzle. The first one, which can be referred to as the "easy distractor", is, once again, randomly extracted from the distractor file. This file also contains, for each distractor, the corresponding count in the total forms file. For this distractor specifically, only words with a count greater than 62 are considered (the average frequency found for the used distractors in the Linguateca totals file). The second distractor, labelled as the "difficult distractor", is chosen in a manner similar to its "easy" counterpart (only words with an occurrence count of 1 are eligible).

It is important to note that both processes of difficulty level attribution, are based solely on quantitative data and therefore, do not always guarantee that a given word with a lower number of occurrences when compared to another, is indeed more difficult than said word. Since no human evaluation was done, other parameters such as the nature of the prefix, the number of meanings it can convey (one or many), the application of the hyphen character and the need to adjust the base form in order to apply the prefix, were not considered.

Finally, and contrary to what happens in the previous cases, the final distractor is chosen using the Levenshtein Distance[15] as a similarity standard. The

[13] `http://www.linguateca.pt/acesso/contabilizacao.php`
[14] `http://www.linguateca.pt/acesso/tokens/formas.total.txt`
[15] `http://php.net/manual/en/function.levenshtein.php`

initial threshold is set to the length of the exception. After this, 10 distractors are randomly chosen from the distractor list, and the one with the lowest Levenshtein Distance, when compared to the threshold value, is chosen as the "similar" distractor.

Support. During the process of solving the exercise, the user has the option of removing a random distractor, by clicking on a specific button of the interface. This can be done up to three times, leaving the final correct answer as the remaining option. This operation has a direct impact on the final score of the exercise, in order to discourage the excessive use of this feature. Besides this, accessing the help screen allows to user the view information on the prefix being used in the exercise. This information was extracted from the Priberam[16] on-line dictionary, and was stored locally in order to speed up the lookup process.

Score and Feedback. After choosing one of the words, the user is presented with a feedback screen where real sentences using the correct and/or incorrect words are presented, whenever they are available. These sentences are extracted from the CETEMPúblico corpus [13], using the Ensinador[17] Tool from Linguateca. This extraction occurs each time the user submits the puzzle. Since the use of this tool requires a form to be filled each time a query is made, the Snoopy[18] PHP Class was used to process these forms and to extract the relevant text from the output web-page. Finally, the user is also presented with his score and the potential maximum score of the puzzle he or she just solved.

5 Evaluation

5.1 Evaluation setup

Two different types of evaluation were carried out for both exercises, crowdsource testing and a questionnaire. In order to obtain a significant number of results, websites were made available for testing both exercises.

Users were asked to solve between three and five puzzles for the Crossword exercise and between five and seven puzzles for the Derivation exercise, in order to try out different combinations of difficulty levels. However, it is important to note that no limitations were made to prevent the users from solving too many or too few exercises, that is why the number of solved puzzles exceeds the value one might have expected by multiplying the number of answers to the questionnaires times the upper limit of the recommended exercises to solve. In similar fashion, no restrictions were applied in order to force the users to try out different combinations of difficulty levels. Again, this was done in order to allow them to progress and to solve more complex puzzles at their one pace.

Several parameters were recorded for each solved exercise. Furthermore, due to the architecture of both puzzles it was possible to record approximately the same information. These parameters consist of: Score, maximum (possible) score,

[16] `http://www.priberam.pt/dlpo/`
[17] `http://www.linguateca.pt/Ensinador/index.pl?corpus=CETEMPUBLICO`
[18] `http://sourceforge.net/projects/snoopy/`

difficulty levels, the number of hints that were used and, in the derivation exercise, the prefix used.

In order to evaluate other aspects of the exercises such as their interface, questionnaires were introduced after the feedback screens of both exercises, if the users followed the corresponding links. These questionnaires were a combination of the *USE Questionnaire* [7] which stands for Usability, Satisfaction and Ease of use, and on GameFlow [15], a model used for evaluating the player's enjoyment in games. Even tough GameFlow is commonly applied to more complex games, such as the ones with time constraints and those which require the user to perform several different tasks, it was possible to apply several of its aspects to the developed exercises. Both questionnaires were comprised of a set of Likert scale questions [6] and a free-form text box (allowing for personalized feedback). Due to size constraints, this section will focus primarily on the Questionnaire results.

5.2 Evaluation results

Crowd-sourced test results. In total, 407 puzzles were submitted for the Crossword Puzzle exercise and 570 for the Derivation exercise. The default combination of difficulty levels (easy-easy) was the preferred one for both exercises (51% and 34% respectively). While users performed better on the second exercise (each level combination had average scores over 50%), more hints were used on the first one (the averages were 2.34 and 0.14 respectively). This was expected, given the amount of time and effort each of these Crossword puzzles took to solve.

Questionnaire results. 59 users completed the Crossword Puzzle questionnaire. Their average age was 23.1, ranging from 18 to 41 years old and the majority of them (73%) had already graduated college. Concerning the Derivation questionnaire, 53 users filled-out the questionnaire form. Their average age was 25.5, ranging from 18 to 57 years old, with the majority of them (75%) having also completed their college education.

In both exercises, users agreed that the systems were easy to use and that the objectives had been quickly understood. Users once again agreed on the existence of differences between difficulty levels, for both exercises (80% and 77% respectively). Concerning the usability and the satisfaction factor from the *USE Questionnaire*, the results were positive and similar in both exercises. Overall, 58% of the inquired populations stated that they had fun while using both systems and only 24% claimed not to have learned anything with them. Finally, 71% of the users claimed to be satisfied with the Crossword Puzzle exercise and 79% claimed the same thing concerning the Derivation exercise. The results obtained in this evaluation stage validate the approach followed in this project.

Questionnaire comments. Users suggested several different changes and tweaks to the interfaces of both exercises, like the placement of the graphical elements. Alternative resources, such as other corpora, and adaptive difficulty levels (depending on the user's performance while solving the puzzles) were also suggested for the Derivation exercise.

6 Conclusion and Future Work

Given the current expansion and ongoing integration of ICALL systems in the educational framework for many languages, it is important that new and innovative approaches are taken when building new tools to be integrated in these systems. It is our opinion that the work developed in this dissertation consists of various valuable assets, which can be applied to serve several of these purposes.

Even though, during the development stage, some unexpected difficulties were unearthed, mainly concerning the YAH hyphenator, the positive results obtained from hyphenating both the P-AWL and the alternative corpus, prove that this tool can still be used effectively and with a reasonable degree of confidence. The correction of the errors that existed in the original version of this tool allows for future updates and revisions, to focus solely on the expansion and revision of the existent rules, in order to increase the efficiency of this tool.

The Crossword Puzzle exercise offers a unique "twist" on the traditional crossword puzzles, opting to focus on syllabic structure by requiring users to correctly divide the words in question. Due to its modular architecture, which has the linguistic resources completely separated from the source code, it can be easily adapted to foreign languages.

Similarly, the Derivation exercise can also be easily updated and adapted to fit different requirements. Once more, the linguistic resources are also independent from the code, which allows for completely new different types of multiple-choice exercises to be implemented altogether.

The fact that both systems present the user with personalized feedback, means that instead of simply informing the users of their performance's result, an extra step is taken in order to give them more information on their errors, and to prevent their repetition in the future. This is especially important in the Derivation exercise, where examples of real sentence offer precious information on how and when to apply the words in question.

Regarding future work, three items were identified:

- A redesign of the YAH's architecture, including the inclusion of "word-ending" rules and "hybrid" rules (capable of handling both mask and word characters), would not only allow it to be ported to other languages, but to increase its effectiveness as well;

- Human maintenance on the Derivation exercise resources would improve the overall quality of the generated exercises. More specifically, the manual correction of the lists of exceptions and distractors, as well as the introduction of additional restrictions to prevent prefixes from using the resources belonging to other prefixes;

- Introduction of alternative sources for the linguistic resources. New dictionaries should be incorporated in the Crossword Puzzle exercise leading to additional definitions. Other corpora and tools should be incorporated in the Derivation exercise, allowing for a larger set of eligible sentences to be displayed on the feedback screen.

Bibliography

[1] Jorge Baptista, Neuza Costa, Joaquim Guerra, Marcos Zampieri, Maria Cabral, and Nuno Mamede. P-AWL: Academic Word List for Portuguese. In Thiago Pardo, António Branco, Aldebaro Klautau, Renata Vieira, and Vera Lima, editors, *Computational Processing of the Portuguese Language*, volume 6001 of *Lecture Notes in Computer Science*, pages 120 – 123. Springer Berlin / Heidelberg.

[2] Rui Correia. Automatic Question Generation for REAP.PT Tutoring System. Master's Thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 2010.

[3] Hélder Filipe. Redesign da Interface de Utilizador do REAP.PT. Master's Thesis, Universidade do Algarve, 2013.

[4] Tiago Freitas. Syntactic REAP.PT - Exercises on Clitic Pronouning. Master's Thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 2012.

[5] VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

[6] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55, 1932.

[7] Arnold M. Lund. Measuring Usability with the USE Questionnaire. *STC Usability SIG Newsletter*, 8(2), October 2001.

[8] Nuno Mamede, Jorge Baptista, Cláudio Diniz, and Vera Cabarrão. STRING: An Hybrid Statistical and Rule-Based Natural Language Processing Chain for Portuguese. In *Proceedings of the 12th international conference on Computational Processing of the Portuguese Language - Demo sessions http://www.propor2012.org/demos/DemoSTRING.pdf*, PROPOR 2012, 2012.

[9] Cristiano Marques. Syntactic REAP.PT. Master's Thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 2011.

[10] Luís Marujo. REAP em Português. Master's Thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 2009.

[11] Detmar Meurers, Ramon Ziai, Luiz Amaral, Adriane Boyd, Aleksandar Dimitrov, Vanessa Metcalf, and Niels Ott. Enhancing authentic web pages for language learners. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, IUNLPBEA '10, pages 10–18, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[12] Sérgio Paulo, Luís C. Oliveira, Carlos Mendes, Luís Figueira, Renato Cassaca, Céu Viana, and Helena Moniz. DIXI — A Generic Text-to-Speech System for European Portuguese. In *Proceedings of the 8th international conference on Computational Processing of the Portuguese Language*, PROPOR 2008, pages 91–100, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] Diana Santos and Paulo Rocha. Evaluating CETEMPúblico, a free resource for Portuguese. In *Proceedings of the 39th Annual Meeting on the Association for Computational Linguistics*, ACL '01, pages 450–457, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[14] André Silva. Pictorial REAP.PT. Master's Thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 2011.

[15] Penelope Sweetser and Peta Wyeth. Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3):3, 2005.

[16] Isabel Trancoso, Thomas Pellegrini, André Freire Silva, Rui Correia, Nuno J. Mamede, and Jorge Baptista. Learning portuguese with speech technologies. In *IBERSPEECH*, pages 411–414, November 2012.

[17] Alexandre Vicente. LexMan: um Segmentador e Analisador Morfológico com Transdutores. Master's Thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 2013.