

Correcção automática de respostas de texto livre para a aplicação UNI_NET-Classroom

Lília Alexandra Soares Araújo

Orientador: Doutora Maria de Fátima Coutinho Rodrigues

Correcção automática de respostas de texto livre para a aplicação UNI_NET-Classroom

Lília Alexandra Soares Araújo

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática Área de Especialização em
Tecnologias do Conhecimento e Decisão**

Orientador: Doutora Maria de Fátima Coutinho Rodrigues

Júri:

Presidente:

Doutor José António Reis Tavares, Professor Adjunto, ISEP

Vogais:

Doutor Fernando Jorge Ferreira Duarte, Professor Adjunto, ISEP

Doutora Maria de Fátima Coutinho Rodrigues, Professora Coordenadora, ISEP

Porto, Outubro de 2011

Para aqueles que acreditaram

Agradecimentos

Em primeiro lugar gostava de agradecer à minha orientadora, Professora Doutora Fátima Rodrigues pelo apoio científico, dedicação e empenho que me passou ao longo de todo o período de desenvolvimento deste trabalho.

Agradeço ao INESC Porto e em particular à Unidade de Sistemas de Informação e de Computação Gráfica pelos meios facultados que permitiram desenvolver o trabalho que deu origem a esta dissertação.

A todos os meus colegas do ISEP e do INESC Porto que foram partilhando o seu conhecimento e experiência e me incentivaram ao longo de todo o percurso. Em especial gostava de agradecer ao Rúben, Jorge, Ricardo Henriques, Henrique, José Carlos, Paulo Melo e ao meu gestor de projecto Rui Barros, pelo incentivo e apoio dado durante todo o projecto.

Aos docentes que me acompanharam ao longo de todo o meu percurso académico, que contribuíram para a aquisição de conhecimento, partilha de experiências e para a minha evolução pessoal e profissional.

Agradeço ao Professor Alberto Simões da Universidade do Minho pela ajuda imprescindível na integração da ferramenta JSpell.

Por fim, agradeço à minha família e amigos pela compreensão, consideração e incentivo que me transmitiram durante este período.

A todos os que referi, e aos que possa não ter referido, o meu mais sincero Obrigado!

Resumo

Ao longo dos tempos foi possível constatar que uma grande parte do tempo dos professores é gasta na componente de avaliação. Por esse facto, há já algumas décadas que a correcção automática de texto livre é alvo de investigação. Sendo a correcção de exercícios efectuada pelo computador permite que o professor dedique o seu tempo em tarefas que melhorem a aprendizagem dos alunos.

Para além disso, cada vez mais as novas tecnologias permitem o uso de ferramentas com bastante utilidade no ensino, pois para além de facilitarem a exposição do conhecimento também permitem uma maior retenção da informação. Logo, associar ferramentas de gestão de sala de aula à correcção automática de respostas de texto livre é um desafio bastante interessante.

O objectivo desta dissertação foi a realização de um estudo relativamente à área de avaliação assistida por computador em que este trabalho se insere. Inicialmente, foram analisados alguns correctores ortográficos para seleccionar aquele que seria integrado no módulo proposto. De seguida, foram estudadas as técnicas mais relevantes e as ferramentas que mais se enquadram no âmbito deste trabalho.

Neste contexto, a ideia foi partir da existência de uma ferramenta de gestão de sala de aula e desenvolver um módulo para a correcção de exercícios. A aplicação UNI_NET-Classroom, que foi a ferramenta para a qual o módulo foi desenvolvido, já continha um componente de gestão de exercícios que apenas efectuava a correcção para as respostas de escolha múltipla. Com este trabalho pretendeu-se acrescentar mais uma funcionalidade a esse componente, cujo intuito é dar apoio ao professor através da correcção de exercícios e sugestão da cotação a atribuir.

Por último, foram realizadas várias experiências sobre o módulo desenvolvido, de forma a ser possível retirar algumas conclusões para o presente trabalho. A conclusão mais importante foi que as ferramentas de correcção automática são uma mais-valia para os professores e escolas.

Palavras-chave: avaliação assistida por computador; processamento de linguagem natural; classificação computadorizada

Abstract

Over the time, it was observed that teachers' spend a great amount of time evaluating. For that reason, the investigation on the field of correction tools had begun. The development of this kind of software allows the teacher to devote more time on tasks that improve student learning.

In addition, more and more new technologies allow the use of very useful tools of teaching, as well as facilitate the knowledge exposition also allowing a greater retention of information. This constantly shifting world provides a very interesting challenge.

The objective of this dissertation was to study the area of computer-assisted assessment. Initially, a state of the art of spell checkers software was done. Based on this analysis a selection was made, later it was integrated on the development of the proposed module. Following this analysis it were studied the most relevant techniques and tools, in correction assessment, that best fit in this work.

Based on the existing UNI_NET-Classroom application it was developed an automatic correction tool. The UNI_NET-Classroom is a software that allows the management of in class activities. This software evaluates the students using an exercises module and the developed module will support the teacher, correcting exercises and suggesting answer and exam quotation.

Lastly, several experiments were carried out on the developed module so that it is possible to draw some conclusions for this work. The most important conclusion was that the automatic correction tools are an asset to teachers and schools.

Keywords: Computer-based assessment, natural language processing, computerized classification

Índice

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	vii
Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Abreviaturas	xv
1. Introdução	1
1.1. Enquadramento	1
1.2. Motivação	1
1.3. Principais objectivos e descrição sumária do trabalho efectuado	2
1.4. Estrutura do Documento	3
2. Correctores ortográficos	5
2.1. Introdução	5
2.2. JSpell	6
2.3. FLiP 8	6
2.4. IESpell	7
2.5. Revisor	7
2.6. Googiespell	7
2.7. Orango	7
2.8. SpellJax	8
2.9. Análise comparativa das ferramentas	8
2.10. Conclusão	10
3. Correção automática de respostas de texto livre	11
3.1. Introdução	11
3.2. Técnicas	11
3.3. Sistemas de Correção Automática	16

3.3.1.	<i>Project Essay Grade (PEG)</i>	17
3.3.2.	<i>Intelligent Essay Assessor (IEA)</i>	18
3.3.3.	<i>Educational Testing Service (ETS I)</i>	19
3.3.4.	Concept- Rater (C-Rater) e Electronic Essay Rater (E-Rater)	20
3.3.5.	<i>Bayesian Essay Test Scoring sYstem (Betsy)</i>	21
3.3.6.	<i>AutoMark</i>	22
3.3.7.	<i>Paperless School free-text Marking Engine (PS-ME)</i>	24
3.4.	Análise comparativa das ferramentas.....	25
3.5.	Conclusão	27
4.	Implementação e Avaliação	29
4.1.	Introdução	29
4.2.	Aplicação UNI_NET-Classroom.....	29
4.3.	Módulo de Correção de Respostas de Texto Livre	31
4.3.1.	Actores que interagem com o Módulo	31
4.4.	Interacção entre as aplicações UNI_NET-Classroom e UNI_NET-Student	34
4.5.	Implementação	36
4.5.1.	Tecnologias utilizadas	36
4.5.2.	Integração do JSpell na aplicação UNI_NET-Classroom	37
4.5.3.	Tarefas de pré-processamento.....	38
4.5.4.	Comparação de respostas e atribuição da cotação	46
4.5.5.	Disponibilização dos resultados	51
4.6.	Caso de estudo.....	54
4.6.1.	Definição do Caso de Estudo	55
4.6.2.	Análise do Caso de Estudo	57
4.7.	Conclusão	62
5.	Conclusões	65
5.1.	Resumo.....	65
5.2.	Objectivos Alcançados.....	67
5.3.	Limitações e Trabalho Futuro	67

5.4. Considerações Finais	68
Referências	71
Anexo I	77

Lista de Figuras

Figura 1 – Representação da resposta “Today Peter rides his bike to Munich” numa rede semântica [Lutticke, 2005].....	14
Figura 2- Visão Geral da abordagem proposta por Chang e Hsiao	15
Figura 3 - IEA, Janela de resultados	18
Figura 4 - Marcação da resposta “The Earth rotates around the Sun”	22
Figura 5 - Arquitectura do sistema	23
Figura 6 - Arquitectura da aplicação	30
Figura 7 - Diagrama de casos de uso realizados pelo professor	33
Figura 8 - Diagrama de casos de uso realizados pelo aluno	34
Figura 9 - Processo de elaboração e correcção de exercícios	35
Figura 10 – Formulário de correcção de exercícios	38
Figura 11 – Estrutura dinâmica com informação do exercício	39
Figura 12 - Algoritmo de remoção de caracteres especiais	40
Figura 13 - Algoritmo de correcção de erros ortográficos	41
Figura 14 - Algoritmo de remoção de <i>stop-words</i>	42
Figura 15 - Algoritmo de categorização das palavras	43
Figura 16 - Algoritmo de carregamento de sinónimos	44
Figura 17 - Evolução da resposta à pergunta “Caracteriza o Urbanismo Pombalino.”	45
Figura 18 – Algoritmo de comparação das respostas	50
Figura 19 - Formulário de demonstração de resultados da aplicação UNI_NET-Classroom.	52
Figura 20 – Formulário de visualização do exercício de um aluno	54
Figura 21 – Comparação das cotações atribuídas por questão.....	58
Figura 22 - Comparação das cotações atribuídas por aluno	62

Lista de Tabelas

Tabela 1 - Comparação dos sistemas de correcção ortográfica analisados.....	9
Tabela 2 - Comparação dos sistemas de correcção automática analisados	26
Tabela 3 - Correspondência das medidas quantitativa e qualitativa	56
Tabela 4 – Resultados gerais por questão.....	57
Tabela 5 – Análise de cotações coincidentes	59
Tabela 6 – Análise de Falsos Positivos e Falsos Negativos	60
Tabela 7 – Comparação das avaliações do sistema/professorma/professor.....	61

Lista de Abreviaturas

Abreviatura	Descrição
AAC	Avaliação Assistida por Computador
ASL	Análise Semântica Latente
Betsy	<i>Bayesian Essay Test Scoring sYstem</i>
C-Rater	<i>Conceptual Rater</i>
CSV	<i>Comma-separated values</i>
<i>E- Rater</i>	Electronic Essay Rater
<i>ETS</i>	<i>Educational Testing Service</i>
FF	Mozilla Firefox
GCSE	<i>General Certificate of Secondary Education</i>
GMAT	<i>Graduate Management Admission Test</i>
IA	Inteligência Artificial
IE	<i>Internet Explorer</i>
IEA	<i>Intelligent Essay Assessor</i>
IEEE	<i>“Institute of Electrical and Electronic Engineers”</i>
LMS	Learning Management System
INESC Porto	Instituto de Engenharia de Sistemas e Computadores do Porto
MSNLP	<i>Microsoft Natural Language Processing Tool</i>
NCA	<i>National Committee on Accreditation</i>
PEG	<i>Project Essay Grade</i>
PLN	Processamento de Linguagem Natural
PS-ME	<i>Paperless School free-text Marking Engine</i>
UML	<i>Unified Modeling Language</i>
VSM	<i>Vector Space Model</i>

1. Introdução

1.1. Enquadramento

Cada vez mais as novas tecnologias permitem o uso de ferramentas com bastante aplicabilidade no ensino pois para além de facilitarem a exposição de materiais didácticos permitem uma maior retenção da informação.

Desta crescente necessidade da utilização de novas tecnologias na educação surgiu o UNI_NET-Classroom. Esta aplicação desenvolvida pelo INESC Porto em parceria com a Nautilus tem como objectivo fornecer aos professores ferramentas que permitam tornar a aula mais interactiva e mais apelativa aos alunos. Se por um lado o professor tem um maior controlo da aula, pois pode monitorizar permanentemente os alunos durante a aula, por outro lado, pode também interagir mais com os alunos através do envio ou recolha de ficheiros e exercícios.

O módulo de exercícios despertou bastante interesse por parte dos Professores pois disponibiliza os resultados já com a correcção das perguntas de escolha múltipla efectuada. Daí surgiu a ideia de facilitar também a correcção das respostas de texto livre. Desta forma os professores passariam a ter mais uma ferramenta de apoio, pelo que reduziriam o tempo de correcção dos exercícios.

Esta tese tem como objectivo analisar técnicas e ferramentas da área de avaliação assistida por computador para posteriormente se desenvolver um módulo que faça a correcção automática das respostas abertas dadas pelo aluno tendo como referência a resposta dada pelo professor no momento da correcção do exercício. Além da correcção do conteúdo da resposta, pretende-se que também verifique os erros ortográficos e os assinale.

1.2. Motivação

Além da grande evolução da área das novas tecnologias nos últimos anos, ainda é complicado integrar novas ferramentas de apoio no ensino, pois os professores apresentam alguma resistência à mudança da forma de ensinar.

Para aumentar a credibilidade destas ferramentas nos estabelecimentos de ensino é necessário demonstrar a fiabilidade das mesmas e o quanto estas podem ser úteis para a evolução do processo de aprendizagem.

O objectivo é mostrar que estas ferramentas de apoio não pretendem substituir o professor mas sim contribuir para o aumento do seu desempenho, pois estas agilizam algumas tarefas morosas. Sendo a capacidade de processamento do computador muito superior à do ser humano, os resultados de determinadas tarefas são obtidos mais rapidamente. No caso concreto da correcção de exercícios isso é bem visível. Apesar de existirem várias componentes de avaliação, o desempenho dos alunos ainda é maioritariamente realizado através de exercícios. Por essa razão, o professor despende uma grande parte do seu tempo na elaboração e conseqüente correcção de exercícios.

A correcção de questões num processo de avaliação envolvendo um grande número de respostas em texto livre é uma tarefa que, além de ser dispendiosa em termos de tempo, de acordo com Mason [Mason & Grove-Stephenson, 2002] é uma tarefa para a qual os professores dedicam cerca de 30% do seu tempo, é também é muito difícil garantir que os critérios de avaliação são aplicados de uma forma equitativa. Isto é devido não só à natureza subjectiva da avaliação das respostas de texto livre, mas também pelo facto do processo de avaliação ser demorado. Os professores devem estar altamente concentrados por longos períodos de tempo, e, portanto, a avaliação está sujeita a variações do nível de concentração e do humor próprias do ser humano. Isto pode levar a diferentes graus de avaliação em relação a respostas com qualidade semelhante, criando assim desigualdades no processo de avaliação, que podem ser ainda mais acentuadas se a avaliação for realizada por diferentes professores.

Com este trabalho pretende-se desenvolver um módulo que apoie o processo de correcção de exercícios com respostas de texto livre na plataforma UNI_NET-Classroom e assim contribuir com mais uma ferramenta de apoio ao professor.

1.3. Principais objectivos e descrição sumária do trabalho efectuado

A aplicação UNI_NET-Classroom proporciona apoio à gestão das aulas aos professores. O módulo de avaliação de conhecimentos permite a realização de fichas ou testes de avaliação que compreendem perguntas com resposta de escolha múltipla e perguntas com

resposta aberta. No primeiro caso, o sistema faz a correcção automática baseado na resposta pré-definida, no segundo caso, pretende-se desenvolver um módulo que faça a correcção automática das respostas abertas dadas pelo aluno tendo como referência a resposta dada pelo professor no momento da correcção do teste. Além da correcção do conteúdo da resposta, pretende-se que também verifique os erros ortográficos e os assinale.

Para a correcta realização deste trabalho, foi feito um levantamento do estado da arte da área de correcção automática de respostas de texto livre.

Após o término da realização, e análise, do estado da arte na área referida é elaborado o módulo de correcção automática integrando-o na aplicação UNI_NET-Classroom.

Por fim, é feita uma análise comparativa entre os resultados definidos pelo professor e os resultados obtidos a partir da aplicação desenvolvida.

1.4. Estrutura do Documento

Nesta secção pretende-se apresentar a forma como estão organizados os capítulos para ser perceptível uma visão geral do tema que será desenvolvido ao longo do documento. Desta forma, será apresentada de seguida uma pequena descrição de cada um dos capítulos.

Capítulo 1 - É efectuado um enquadramento do tema deste trabalho, as motivações que levaram à sua realização e uma breve descrição dos principais objectivos acompanhados por uma descrição sumária do trabalho realizado;

Capítulo 2 – É efectuado um estudo de alguns correctores ortográficos de forma a seleccionar um para ser integrado no módulo a desenvolver;

Capítulo 3 – Tem como objectivo fazer uma contextualização do principal tema deste trabalho, que se foca na correcção automática de respostas de texto livre. Para tal, são analisadas algumas técnicas e sistemas existentes nesta área;

Capítulo 4 - Tem como intuito explicar os diferentes aspectos relacionados com a implementação do módulo e respectiva avaliação do mesmo através da análise de alguns casos de estudo;

Capítulo 5 - Neste último capítulo são apresentadas as conclusões de todo o trabalho realizado. Esta análise contempla as principais contribuições do trabalho realizado, as limitações e o que poderá ser melhorado no futuro.

2. Correctores ortográficos

2.1. Introdução

O objectivo deste capítulo é apresentar, sucintamente, as principais características de alguns correctores ortográficos de forma a seleccionar um, que futuramente será integrado na aplicação UNI_NET-Classroom.

As primeiras pesquisas relativamente ao processamento de linguagem natural surgiram na década de 50. A partir da década de 60 foram desenvolvidas ferramentas para a interpretação de linguagem natural com o intuito de melhorar a interacção homem-máquina. Desta forma, um sistema computacional poderia responder a questões de um ser-humano, através da interpretação de palavras-chave. Hoje em dia, os sistemas já fornecem informações morfológicas, sintácticas e semânticas [Oliveira, 2009]. A informação morfológica de cada palavra (género, número, categoria gramatical,...) está disponível através de um dicionário base e de um conjunto de regras de formação de novas palavras [Simões & Almeida, 2001]. A análise sintáctica permite verificar se a frase está bem construída gramaticalmente, tendo em consideração a ligação entre as palavras e os termos essenciais, como é o caso do sujeito e predicado [Oliveira, 2009]. A análise semântica pretende interpretar o significado das palavras relativamente ao contexto em que esta está inserida.

Neste estudo apenas são analisados correctores ortográficos para posteriormente ser efectuada a sua integração com a aplicação UNI_NET-Classroom. De uma forma genérica pode-se considerar que um corrector ortográfico é um programa computacional que faz a verificação de erros de um texto para detectar e dar sugestões relativamente a uma possível correcção de erros ortográficos. Na prática o seu funcionamento fundamenta-se na comparação das palavras de um determinado documento com as palavras existentes numa lista designada por dicionário. Se a palavra não constar na lista é reconhecida como errada e por conseguinte é apresentada uma sugestão de correcção [Figueira et al., 2009].

De seguida serão descritas algumas ferramentas de correcção ortográfica existentes.

2.2. JSpell

O Jspell é uma ferramenta livre desenvolvida em 1994 por Ulisses Pinto e José João Almeida sendo posteriormente integrado no projecto Natura. O seu código fonte foi baseado no Ispell. O principal objectivo é verificar a ortografia e morfologia de textos e/ou palavras da Língua Portuguesa. Esta aplicação está em constante evolução, sendo actualmente desenvolvida por: Rui Vilela, José João Almeida e Alberto Simões.

Neste momento a versão do Jspell é International Jspell Version 1.63 e contem os seguintes dicionários: Dicionário de Inglês (actualizado a 26 de Junho de 2009), Dicionário de Latim (incompleto e actualizado a 2 de Março de 2009), Dicionário de Português (actualizado a 24 de Janeiro de 2010) e Dicionário de Castelhana (incompleto e actualizado a 2 de Março de 2009).

Para tornar possível o acesso a esta ferramenta a um número elevado de utilizadores foi criada a ferramenta WebJspell.

Os dicionários utilizados pelo Jspell são de utilização livre. Não têm incorporado todo o vocabulário existente para tornar mais eficiente o seu funcionamento. Só utilizam as formas mais frequentes reduzindo assim a dimensão da procura e causando menos inconsistência [WebJspell, 2010].

2.3. FLiP 8

O FLiP 8 (Ferramentas para a Língua Portuguesa) é um auxiliar de escrita que permite a correcção ortográfica, apresentando sugestões com a vantagem de conter dicionários temáticos que facilitam a correcção de textos de áreas técnicas. Além disso, permite a correcção sintáctica e estilística, que pode ser personalizado pelo utilizador e um dicionário de sinónimos. Inclui uma ferramenta que auxilia a tradução entre português, espanhol, francês e inglês. Para além de português, tem o corrector ortográfico disponível em espanhol, alemão, francês, inglês e italiano.

Esta ferramenta é constituída pelos seguintes módulos: correctores ortográficos, dicionários temáticos, correctores sintácticos e estilísticos, dicionários de sinónimos e auxiliares de tradução [FLiP, 2010].

A utilização desta ferramenta implica a aquisição do produto à Porto Editora.

2.4. IESpell

O IESpell é um corrector ortográfico que verifica a informação inserida em formulários de páginas Web, funcionando como uma extensão para o Internet Explorer. Esta aplicação é útil para utilizadores que frequentemente insiram texto na Web, em serviços como *webmail*, *blogs*, diários, entre outros. A língua disponibilizada é o Inglês em três variantes: Estados Unidos, Reino Unido e Canadá. O seu funcionamento consiste em comparar as palavras que foram escritas incorrectamente para depois sugerir palavras semelhantes. Além disso, ainda é possível adicionar ou remover palavras ao dicionário de uma forma fácil e intuitiva.

Para fazer a verificação da ortografia das páginas Web é necessário executar a aplicação para que esta indique quais os erros e possíveis correcções [IeSpell, 2010].

2.5. Revisor

O Revisor é um corrector ortográfico que funciona *online* utilizando o motor de verificação do Google. Tem suporte para onze idiomas diferentes e funciona sem ter de recarregar a página [Revisor, 2010].

2.6. Googiespell

O Googiespell é um corrector ortográfico que pode ser utilizado em qualquer aplicação Web. Tem suporte em 27 idiomas e funciona em diferentes navegadores como o IE 5.5, IE 6.0, IE 7, FF 1.0 +, Safari, Opera, entre outros. A vantagem é que é extensível e fácil de instalar [GoogieSpell, 2010].

2.7. Orangoo

O Orangoo é um corrector ortográfico que funciona directamente em textos ou em páginas Web na íntegra, tendo como requisito o acesso à internet. Está disponível em 27 línguas diferentes. O código é livre pelo que pode ser utilizado em qualquer aplicação [Orangoo, 2010].

2.8. SpellJax

O SpellJax é um corrector ortográfico online gratuito. O seu principal objectivo é a verificação de palavras num determinado documento desde que exista um acesso à internet. O seu funcionamento é simples bastando colocar o texto no formulário que é disponibilizado [SpellJax, 2010].

2.9. Análise comparativa das ferramentas

De forma a seleccionar qual a ferramenta a integrar com a aplicação UNI_NET-Classroom fez-se uma análise comparativa das ferramentas anteriormente referenciadas. Existem várias características que são de elevada importância e que é necessário ter em consideração para a tomada de decisão. Uma das características mais importantes é ter a Língua Portuguesa no conjunto de idiomas visto que essa é a língua base da aplicação. Outra característica relevante é o tipo de licença, que sendo gratuita será dada preferência por uma questão de minimização de custos da solução. No entanto são comparadas outras características que diferenciam as ferramentas analisadas.

Tabela 1 - Comparação dos sistemas de correcção ortográfica analisados

Sistemas	Jspell	FLiP 8	IESpell	Revisor	Googiespell	Orangoo	SpellJax
Análise Ortográfica	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Análise Morfológica	Sim	Não	Não	Não	Não	Não	Não
Análise Sintáctica	Não	Sim	Não	Não	Não	Não	Não
Número de Línguas	4	5	1	11	27	27	11
Tipo de Utilização	Local	Local	Online	Online	Online	Online	Online
Tipo de Licença	Gratuita	Paga	Gratuita (uso pessoal)	Gratuita	Gratuita	Gratuita	Gratuita

A comparação realizada na Tabela 1 tem por base os seguintes parâmetros:

- Análise Ortográfica: este parâmetro é requisito mínimo para qualquer um dos correctores ortográficos, e tem como objectivo detectar erros ortográficos.
- Análise Morfológica: analisa cada palavra independentemente;
- Análise Sintáctica: verifica a estrutura da frase;
- Número de línguas: o número de línguas por si só não é muito relevante mas permite ter a percepção de que a ferramenta suporta mais do que uma língua. Actualmente o UNI_NET-Classroom é comercializado nos seguintes idiomas: Português, Espanhol, Inglês e Alemão.
- Tipo de Utilização: o tipo de utilização pode ser local ou então ter a necessidade de acesso à internet. No caso de estudo o corrector ortográfico será para ser usado localmente.

Tipo de Licença: o tipo de licença é um parâmetro bastante relevante visto que a procura incide sobre uma licença de livre utilização.

2.10. Conclusão

Depois de analisar as várias possibilidades, concluiu-se que o corrector ortográfico a utilizar será a ferramenta Jspell, visto que em comparação com os outros correctores ortográficos é o que melhor satisfaz os requisitos impostos anteriormente. Esta ferramenta é livre e tem o código fonte disponível para o caso de ser necessário efectuar alguma alteração: permite a análise ortográfica, que é o principal objectivo para a sua utilização; contém a análise morfológica que será útil na categorização das palavras para o módulo de correcção de respostas de texto livre. Oficialmente contém dicionários de Português e Inglês, contudo foram também disponibilizados os dicionários de Espanhol, Francês e Alemão que são muito úteis para as diversas disciplinas leccionadas no sistema de ensino. Outra vantagem é que este projecto está em constante desenvolvimento e por isso vão surgindo novas versões com correcções, que poderão ser benéficas para a integração.

3. Correção automática de respostas de texto livre

3.1. Introdução

A Avaliação Assistida por Computador (AAC) de respostas em texto livre é um problema de longa data que tem atraído o interesse de investigadores desde 1960. Os sistemas de AAC podem ser distinguidos pela forma como avaliam os textos, podendo ser pelo estilo ou pelo conteúdo, ou utilizando ambos os aspectos. As abordagens mais importantes encontradas em sistemas de AAC são a Análise Estatística, a Análise Semântica Latente (ASL) (*Latent Semantic Analysis*) e o Processamento de Linguagem Natural (PLN) (*Natural Language Processing*). Os primeiros sistemas de AAC, com foco em abordagens estatísticas capturam apenas a semelhança estrutural dos textos. Os sistemas seguintes, com base na ASL, fazem mais do que uma simples análise de co-ocorrência de termos. Estes sistemas fazem uma comparação com base no corpus do texto usando uma técnica algébrica que consegue considerar dois textos com palavras distintas, semanticamente similares [Thomas et al., 2004]. Os últimos sistemas baseados em técnicas de PLN fazem análises inteligentes de forma a captar o significado semântico de documentos de texto livre, como é o caso de Marin [Marin et al., 2004] que o faz com grande profundidade.

Estes sistemas têm interesse pois os professores despendem uma parte considerável do seu tempo a corrigir exercícios, sendo esta a forma mais comum de avaliar os alunos. A correção automática de respostas de texto livre facilita o processo de correção, pois o professor tem ao seu dispor a correção sugerida pelo sistema podendo sempre que necessário rectificar as cotações atribuídas. A vantagem destes sistemas prende-se com o facto de estes não serem influenciados por prazos, fadiga ou até mesmo preconceitos, permitindo assim uma visão coerente e objectiva da qualidade dos testes.

Nesta secção será feito um estudo de algumas das técnicas e ferramentas existentes de correção automática de respostas de texto livre.

3.2. Técnicas

Existem diversas técnicas de classificação de texto livre que podem ser utilizadas na avaliação assistida por computador. Foram estudadas algumas dessas técnicas e categorizadas da seguinte forma: técnicas estatísticas que podem ser divididas em análise

simples por palavra-chave, análise de características linguísticas superficiais, análise semântica latente e técnicas de categorização de texto; processamento completo de linguagem natural; técnicas de extracção de informação; *clustering*; comparação de redes semânticas e abordagens híbridas [Marin, 2007].

- **Técnicas Estatísticas:** Utilizam a análise estatística de uma ou diversas características dos textos e técnicas de PLN menos complexas sendo que os textos na maioria dos casos são processados através do uso de um divisor do texto em frases (*sentence splitter*) e de um divisor da frase em palavras (*tokenizer*). Como consequência, devem ser fáceis de funcionar em várias línguas e domínios. Normalmente precisam de uma fase de treino inicial para o cálculo dos parâmetros do sistema. As técnicas estatísticas podem ser divididas em várias subcategorias:
 - **Análise simples por palavra-chave:** É a técnica mais simples e consiste em procurar palavras-chave ou n-gramas coincidentes entre o texto do aluno e o modelo do professor. A partir deste método não é possível extrair uma representação do significado da resposta do aluno nem lidar com sinónimos e termos polissémicos. Por estas razões normalmente é utilizado como método auxiliar o Modelo Espaço Vectorial (*Vector Space Model*) que se baseia na representação do texto em vectores [Li & Hu, 2009] sendo calculada a similaridade entre documentos através do ângulo que formam os vectores representativos dos documentos [Wang, 2009].
 - **Análise de características linguísticas superficiais:** Nesta subcategoria são incluídos os sistemas que requerem: uma lista de recursos que vão ser medidos, uma fase de treino para descobrir os pesos para cada um deles e uma fase de calibração para ajustar os pesos aos valores ideais. Para avaliar um texto, é necessário efectuar o processamento tendo em consideração as características e utilizá-las como variáveis independentes na função de regressão linear cujo resultado é a pontuação. A principal desvantagem desta técnica é escolher quais os recursos que vão ser considerados [Marin, 2007].
 - **Análise Semântica Latente:** É uma técnica estatística complexa para extrair e representar a utilização contextual do significado das palavras através de cálculos estatísticos aplicados a um grande corpus de texto. Através da indução de conhecimento global, ASL adquiriu conhecimento de vocabulário da língua inglesa com uma taxa comparável com as crianças em idade escolar. Esta técnica é baseada unicamente num método matemático de aprendizagem genérico atingindo grandes efeitos indutivos através da

extracção do número certo de dimensões para representar objectos e contextos [Landauer & Dumais, 1997]. ASL não faz o processamento de linguagem natural tradicional nem é um programa de inteligência artificial, recebe apenas como entrada texto bruto, separado em trechos ou amostras de textos mais significativos [Landauer et al., 1998]. A ideia desta técnica baseia-se na seguinte fórmula: “significado palavra 1” + “significado palavra 2” + ... + “significado palavra n” = “significado da passagem de texto” [Dikli, 2006].

- **Técnicas de categorização de texto:** Os sistemas que utilizam esta técnica assumem que a correcção automática de questões é um problema de classificação e para tal existe uma fase inicial de treino para a obtenção de valores relativos às características mais relevantes para a pontuação final e depois a classificação é feita a partir do modelo.

- **Processamento completo de linguagem natural:** Esta técnica que é mais conhecida como PLN consiste na aplicação de métodos computacionais para processar a linguagem natural. Os textos utilizados podem ser orais ou escritos e têm de estar numa linguagem usada pelos humanos em situações reais. O objectivo desta técnica é realizar o processamento da linguagem utilizada por humanos representando o verdadeiro significado e intenção da questão do utilizador que pode ser expressa numa linguagem comum ou mais cuidada. Um sistema que aplique esta técnica deve ser capaz de parafrasear um texto de entrada, traduzir o texto para outra língua, responder a questões e inferir conhecimento sobre o texto [Liddy, 2001].

- **Técnicas de Extracção de Informação:** Os sistemas que utilizam esta técnica adquirem informação relevante a partir de texto livre. Essa informação é isolada, de forma a ser retirado o conteúdo mais relevante para de seguida ser estruturado [Cohen, 1996]. O processo de extracção de informação pode ser dividido em duas partes: na extracção de factos individuais a partir da análise de textos locais e na integração desses factos, produzindo factos maiores ou novos factos (através de inferência). No final, os factos mais relevantes são transformados no formato de saída [Grishman, 1997].

- **Clustering:** É uma técnica que permite dividir textos em grupos de objectos similares (imagens, palavras, documentos, entre outros) [Wanner, 2004]. Os objectivos desta técnica centram-se em dois pontos: agrupar objectos com características o mais

similares e separar o mais possível os grupos, ou seja, tenta-se maximizar a intra-similaridade dentro dos grupos e maximizar a inter-similaridade entre grupos. [Visnick, 2003].

- **Redes semânticas:** As respostas dos alunos podem ser comparadas com uma resposta de referência através do uso de redes semânticas. Através de entidades e relações é possível verificar se a resposta contém erros ou palavras em falta. Um exemplo da utilização desta técnica é a ferramenta desenvolvida por Lutticke [Lutticke, 2005]. Esta ferramenta de correcção automática está integrada no sistema e funciona da seguinte forma: recebe as redes semânticas do professor e do aluno e devolve como resultado as entidades e relações que estão erradas, em falta e não estão ligadas correctamente. Além disso, o resultado da análise é apresentado textualmente.

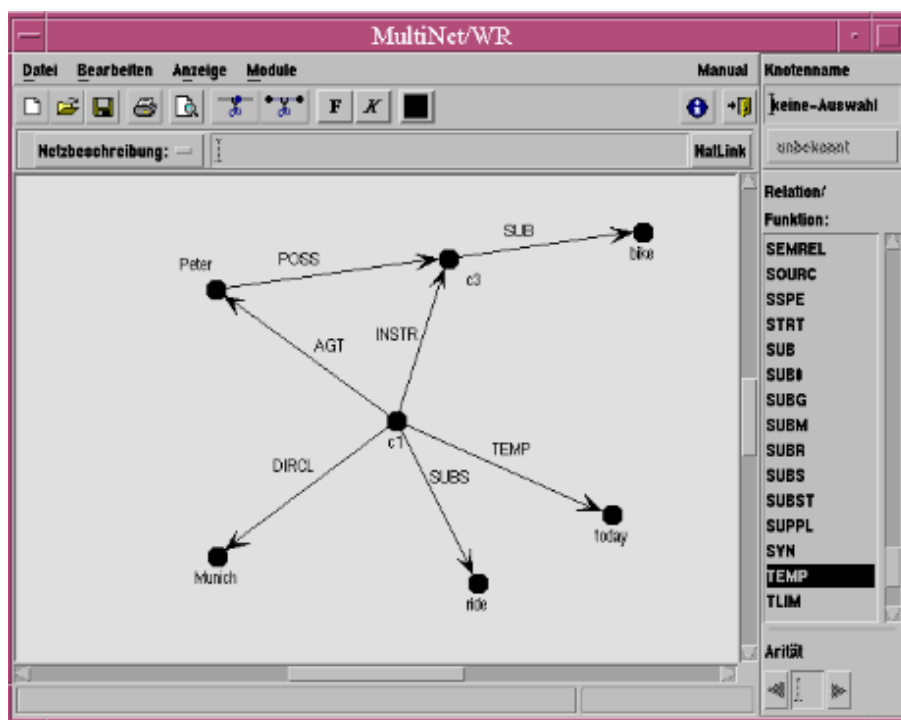


Figura 1 – Representação da resposta “Today Peter rides his bike to Munich” numa rede semântica [Lutticke, 2005]

A Figura 1 ilustra um formulário da aplicação Multinet com a representação de uma rede semântica que poderia representar a resposta do professor.

- **Abordagens híbridas:** Esta técnica é frequentemente utilizada para gerar automaticamente resumos de textos. As abordagens híbridas normalmente consistem em quatro tarefas: pré-processamento, determinação do peso do termo, exploração do relacionamento do termo e geração do resumo [Chang & Hsiao, 2008].

A Figura 2 ilustra a forma como as tarefas se relacionam entre si.

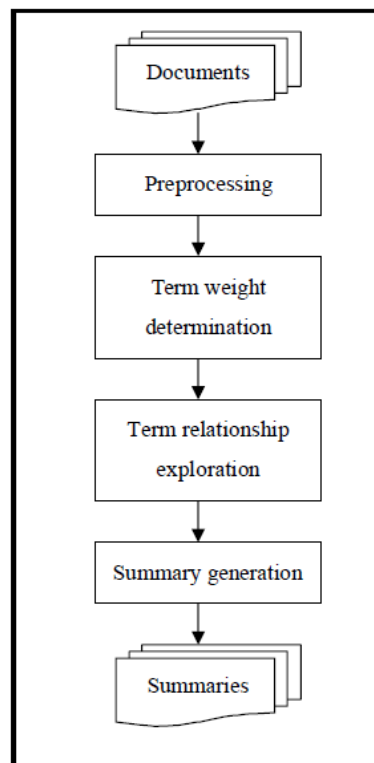


Figura 2- Visão Geral da abordagem proposta por Chang e Hsiao

De seguida serão descritas, de uma forma simplificada, todas as tarefas indicadas na figura, de forma a facilitar a compreensão de todo o processo.

Pré-processamento: Tem como intuito analisar os documentos para extrair os termos chave;

Determinação do peso do termo: Esta tarefa atribui pesos aos termos que foram extraídos anteriormente. A atribuição de pesos aos termos pode ser feita de várias formas, de um modo simplista com base no número de ocorrências dos termos em todo o texto, ou com base em modelos, com base num modelo de probabilidades;

Exploração do relacionamento do termo: O objectivo desta tarefa, tal como o nome indica, é explorar o relacionamento dos termos entre substantivos e entre substantivos e verbos.

Geração do resumo: No final as frases com maior pontuação são seleccionadas para formar o resumo.

3.3. Sistemas de Correção Automática

Durante os últimos anos foram desenvolvidos vários sistemas nesta área, dos quais serão apresentados os principais sistemas que se distinguem pelo uso das diferentes técnicas apresentadas anteriormente.

A filosofia dos investigadores, o tipo de avaliação, o método de avaliação e o formato dos sistemas, são descritos nas obras estudadas. São analisados também alguns comentários dos autores sobre a eficácia dos sistemas. É importante notar, no entanto, que os dados de teste e as métricas usadas para analisar todos os sistemas não são consistentes, logo os resultados não são necessariamente comparáveis.

Um dos maiores problemas actualmente enfrentados nesta área é a falta de materiais de referência para testes. Não existe um grande corpus de textos que possa ser utilizado como uma medida padrão dos sistemas de classificação automatizada [Valenti et al., 2003]. Com a maioria dos projectos de pesquisa a utilizar o seu próprio conjunto de testes e a analisar de acordo com os seus próprios critérios de correlação, é difícil comparar com exactidão os sistemas. Por este motivo, existe muita discussão e discordância na tentativa de determinar qual o melhor sistema e métodos. Conforme mencionado, a distinção é feita na classificação de testes baseados no conteúdo e no estilo. Embora existam sistemas que avaliam baseando-se principalmente no estilo, como o *Project Essay Grade* (PEG), ou no conteúdo, como o *Intelligent Essay Assessor* (IEA), *Educational Testing Service* (ETS I) e *Conceptual Rater* (C-Rater), ou em ambas as dimensões, como *Bayesian Essay Test Scoring sYstem* (Betsy) [10], *AutoMark* e *Paperless School free-text Marking Engine* (PS-ME). Actualmente, as principais implementações do mercado variam muito entre os métodos utilizados. Algumas destas implementações, por exemplo, IEA é baseado em ASL, E-rater em LPN, Betsy em Redes Bayesianas e PEG sobre os recursos linguísticos através de elementos (proxies).

O objectivo desta secção é apresentar, de uma forma resumida, as principais características destes sistemas. Na realização desta análise, utilizou-se artigos/publicações referentes a cada um dos sistemas.

3.3.1. Project Essay Grade (PEG)

Nos anos 60 alguns investigadores começaram a desenvolver a ideia de que o computador poderia desempenhar um grande papel na avaliação dos alunos [Page, 1994]. Desses investigadores pode-se destacar Kuno em 1964, Stone, Dunphey, Smith & Ogilvie em 1966. No desenvolvimento de PEG (de 1966 a 1973) começou a ser testada a classificação feita por computador. Em PEG-1 foi utilizado um conjunto de testes que já tinham sido avaliados por professores de Inglês (dados cedidos por McColly & Remstad, 1963) e com recurso a um computador *Mainframe* (que utilizava cartões perfurados) foram escritos programas que contavam vários eventos. No final, com os resultados obtidos foi aplicada uma regressão múltipla. Os resultados dos testes foram surpreendentes (correlação de 0.5) quando comparados com os resultados dos avaliadores humanos pois a divergência era semelhante à que ocorria entre eles.

Desde o início, existia a noção de que para as avaliações terem qualidade deviam ser definidas por avaliadores humanos. No entanto, era claro que a avaliação por humanos também é falível e a discordância entre avaliadores é bastante elevada. Estas desvantagens poderiam ser minimizadas com o aumento de avaliadores, pois com o uso de grupos a confiabilidade era maior apesar dos custos aumentarem substancialmente.

A teoria estatística de PEG

A estratégia de classificação é encontrar e adaptar os melhores elementos (*proxes*), de forma a prever a classificação de um grupo de avaliadores humanos. Depois de seleccionar e modificar os elementos, a teoria estatística de PEG baseia-se na fórmula da regressão múltipla:

$$J' = a + b_1P_1 + b_2P_2 + b_kP_k,$$

onde J' é a previsão otimizada da média da avaliação humana, a é uma constante, e b_1, b_2, \dots, b_k são os coeficientes de regressão associados aos *proxes* P_1, P_2, \dots, P_k .

Este sistema pode ser utilizado em diversas áreas e conseguiu obter, de acordo com os testes realizados, uma correlação de 87% entre a pontuação obtida pelo sistema e pelos avaliadores humanos [Marin, 2007].

3.3.2. *Intelligent Essay Assessor (IEA)*

Intelligent Essay Assessor foi desenvolvido no final dos anos 90 por Thomas K. Landauer da *University of Colorado*, Boulder e Peter W. Foltz da *New Mexico State University* nos USA. Neste projecto foi utilizada a ASL, a qual foi concebida inicialmente para indexar documentos e recuperar textos no final dos anos 80 [Jerrams-Smith et al., 2001].

IEA funciona como um serviço *Web* e providencia uma avaliação imediata com comentários sobre traços específicos, ortografia e erros gramaticais, tal como ilustra a Figura 3 [IEA, 2011].

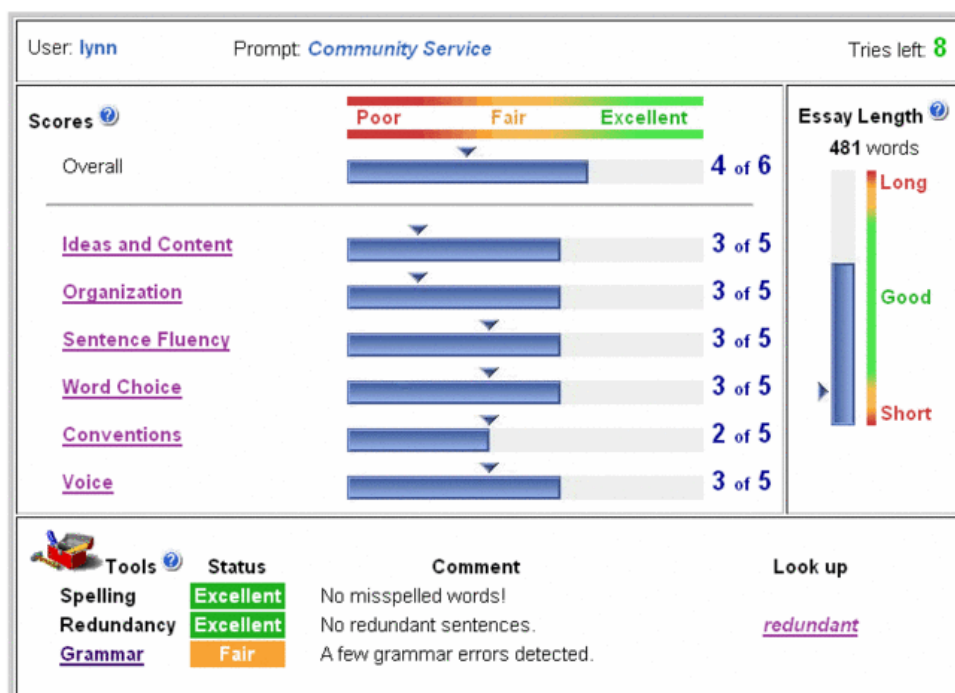


Figura 3 - IEA, Janela de resultados

Visto que este sistema utiliza a ASL tem de ser treinado para o domínio sobre o qual deve intervir [Foltz et al., 1999]. Essa informação pode ser adquirida em textos sobre o domínio,

livros, artigos entre outros e é representada pelo sistema através de uma alta dimensão semântica.

Este sistema pode ser aplicado em diversas áreas da educação. Pode ser usado simplesmente para validar a consistência para posteriormente o professor avaliar os testes e posteriormente IEA reavalia-os e indica as divergências entre as duas correcções. Num nível mais interactivo, este sistema pode interagir com os alunos de forma a apoiá-los na escrita de texto sobre um determinado tema. O sistema pode intervir dando uma avaliação imediata ao texto ou então fornecendo comentários sobre conteúdos que estejam em falta, as perguntas que o aluno deve abordar, ou sugestões sobre onde encontrar a informação. IEA teve uma taxa de acordo entre 85% e 91% num teste que envolveu a correcção de um exame GMAT [Valenti et al., 2003]. Este sistema também é utilizado em psicologia, medicina e textos de história tendo, numa escala de 0 a 100 pontos, uma taxa de acordo entre 80% e 90% [Marin, 2007].

3.3.3. Educational Testing Service (ETS I)

Educational Testing Service foi desenvolvido por Jill Burstein e Randy Kaplan nos anos 90. O objectivo foi desenvolver um sistema automático de atribuição de cotação a respostas de texto livre baseado no conteúdo. Assim sendo, foi necessário identificar o conteúdo relevante de uma resposta e classificá-lo por categoria [Burstein et al., 1996]. Foi considerado que a atribuição da cotação às respostas era um problema de classificação. O sistema para além de classificar correctamente as respostas pelo conteúdo, também deveria ser capaz de determinar as respostas que têm duplo significado [Whittington & Hunt, 1999]. Este sistema apenas funciona se o conteúdo a pesquisar ou respectivos sinónimos ocorrerem nos dados de treino. Os dados de treino servem para indicar ao computador o que este deve procurar mas não procura outra estrutura sintáctica que possa ser utilizada.

A técnica utilizada, que designaram de técnicas léxico-semânticas (PLN), é baseada em pequenos conjuntos de dados e funciona para um domínio específico, baseada nos conceitos léxico e gramatical, ambos construídos a partir dos dados de treino e com alguma intervenção manual.

Os testes de treino são analisados pelo *Microsoft Natural Language Processing Tool* (MSNLP) e todos os sufixos e *stopwords* são removidos manualmente.

As regras de gramática também são construídas manualmente, por categoria da resposta, utilizando a análise sintáctica e lexical das respostas a partir dos dados de treino.

O sistema envolve muito pré-processamento e grande parte dele é feito manualmente, no entanto os autores argumentam que o custo, ao longo do tempo, compensa.

Nos testes realizados ao sistema foi alcançada uma precisão de 93% na marcação do conjunto de teste e de 96% na marcação dos conjuntos de treino e de teste [Valenti et al., 2003].

3.3.4. Concept- Rater (C-Rater) e Electronic Essay Rater (E-Rater)

C-Rater e E-Rater foram desenvolvidos pela organização American Educational Testing Service (ETS).

C-Rater é um protótipo baseado em PLN que visa a avaliação de respostas curtas relacionadas com o conteúdo tendo como referência questões como as que podem aparecer na secção de revisão de um determinado capítulo de um livro. O principal objectivo desta ferramenta é distinguir se a resposta do aluno está correcta ou errada em conformidade com os seus conteúdos [Valenti et al., 2003].

E-Rater utiliza PLN e estatística para o processamento de linguagem natural de forma a modelar o desempenho dos avaliadores humanos. Para cada resposta, é seleccionada uma amostra que contém respostas que tenham sido corrigidas pelo menos por dois avaliadores humanos e que representam a gama completa de pontos possíveis [Powers et al., 2001].

C-Rater está relacionado com E-Rater na medida em que ambos usam a linguagem natural, ferramentas e técnicas de processamento, mas diferem em alguns aspectos importantes [Burstein et al., 2001]. A atribuição da pontuação pelo E-Rater tem como base a capacidade de escrita e não um conteúdo específico pelo que não há resposta correcta, apenas uma descrição de como identificar uma boa resposta. Para C-Rater distinguir uma resposta correcta de uma resposta incorrecta tem de identificar se a resposta contém informações específicas de alguns conceitos em particular, e nesse caso a resposta é correcta, caso contrário é errada, sem levar em conta a capacidade de escrita. Relativamente à estrutura, E-Rater identifica e disponibiliza uma grelha baseada na estrutura retórica de um ensaio. A forma da estrutura retórica e a sua organização são os principais pontos da resposta. No caso de C-Rater, é necessário identificar o conteúdo específico, para de seguida gerar uma análise detalhada da estrutura predicado-argumento, ou relações lógicas entre os componentes sintácticos (por exemplo, sujeito, verbo, objecto) para cada resposta. Para ser

possível efectuar testes, E-Rater contém uma colecção de 270 testes corrigidos manualmente por avaliadores. No caso de C-Rater não é necessário uma grande recolha de respostas para treino pois este usa uma única resposta correcta que é encontrada na resolução do avaliador ou resposta-chave.

Relativamente ao desempenho destes sistemas, E-Rater obteve um acordo de 97% na correcção de 750 000 exames GMAT enquanto que C-Rater obteve um acordo de 80% relativamente à classificação atribuída pelo instrutor [Marin, 2007].

3.3.5. *Bayesian Essay Test Scoring sYstem (Betsy)*

BETSY foi desenvolvido entre 2001 e 2003 por Lawrence M. Rudner e Tahung Liang [Marin, 2007] com o principal intuito de classificar textos com base em material de treino. Foi projectado para atribuir pontuação automaticamente e pode ser aplicado a qualquer tarefa de classificação de texto [Betsy, 2011].

Este sistema tem como objectivo determinar a classificação mais provável com base nos itens analisados categorizando cada um deles, por exemplo, como extenso, essencial, parcial ou insatisfatório. Os itens são um amplo conjunto de funcionalidades que contêm as características da resposta como é o caso dos conteúdos e outras características como a ordem correcta dos conceitos [Rudner & Liang, 2002].

Na prática o sistema determina três probabilidades: probabilidade de que a característica está incluída no teste, dado que o aluno tenha fornecido uma resposta apropriada; probabilidade de que a característica está incluída no teste, dado que o aluno forneceu uma resposta parcialmente apropriada e a probabilidade de que a característica está incluída no teste, dado que o aluno forneceu uma resposta inadequada. Desta forma a resposta será classificada como apropriada, parcial ou inapropriada com base num grande número de respostas avaliadas por humanos. Inicialmente as probabilidades são iguais, mas vão sendo actualizadas através da aplicação do Teorema de Bayes que consiste na seguinte fórmula: $P(A|B) * P(B) = P(B|A) * P(A)$ [Rudner & Liang, 2002].

O sistema obteve uma precisão de 80% em testes efectuados na área de Biologia mas segundo os autores pode ser usado em qualquer área [Marin, 2007].

3.3.6. AutoMark

AutoMark utiliza técnicas de Extração de Informação para efectuar a marcação de respostas curtas de texto livre [Mitchell et al., 2002]. O sistema é constituído por módulos específicos que têm como intuito marcar erros ortográficos, de escrita, de sintaxe e de semântica. AutoMark procura conteúdo específico dentro das respostas de texto livre, baseando-se em modelos criados anteriormente. Inicialmente é feita uma análise às respostas dos alunos combinando-as posteriormente com os modelos. A forma de representação dos modelos permite que estes sejam mapeados de forma robusta mesmo com múltiplas variações no texto de entrada. AutoMark emprega técnicas de PLN para realizar uma pesquisa inteligente de respostas de texto livre em respostas pré-definidas. Tal como os humanos, o sistema tenta identificar a ideia expressa na resposta de texto livre, sem penalizar indevidamente o aluno por erros ortográficos, de gramática ou de semântica. O sistema utiliza um esquema de marcação que identifica as respostas aceitáveis e inaceitáveis para cada questão. Cada modelo especifica uma forma particular de resposta aceitável ou inaceitável. Através da Figura 4 pretende-se demonstrar a marcação de uma resposta utilizando um modelo simples.

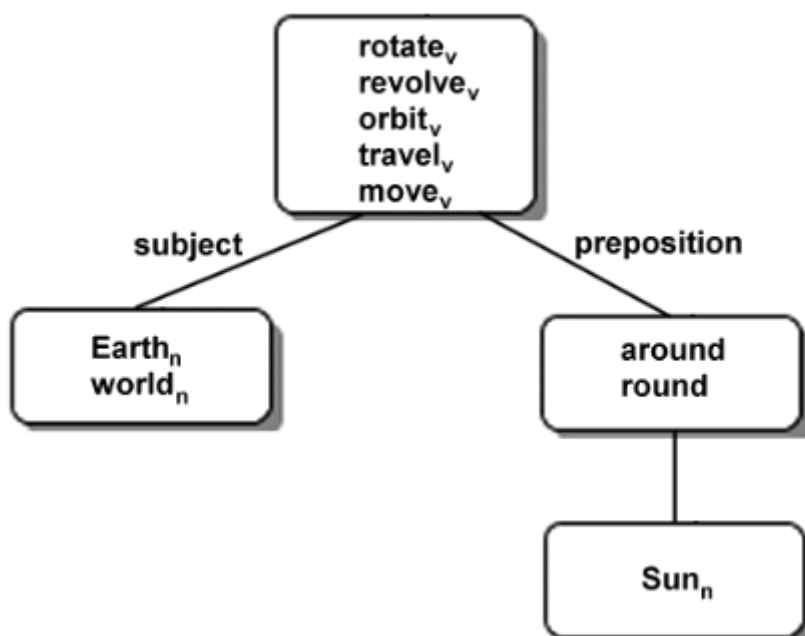


Figura 4 - Marcação da resposta “The Earth rotates around the Sun”

Para que o modelo apresentado coincida com a resposta do aluno, é necessário que a resposta inclua um dos verbos indicados no modelo (*rotate, revolve, orbit, travel, move*), um dos nomes (*Earth, world*) como sujeito e uma das duas posições (*around, round*) que faça a ligação com *Sun*. Para aumentar a abrangência do modelo, os verbos da resposta do aluno são reduzidos à sua forma base.

A elaboração dos modelos é efectuada em modo *offline*, através de uma interface de configuração. Através da análise da Figura 5 pode-se identificar as principais operações efectuadas pelo sistema:

- Syntactic Preprocessing: tem como intuito fazer o pré-processamento do texto de entrada para o padronizar em termos de pontuação e ortografia;
- Sentence Analysis: identifica os constituintes sintácticos do texto e de que forma eles estão relacionados;
- Pattern Matching: procura a correspondência entre os modelos e os constituintes sintácticos do texto do aluno;
- Feedback Module: processa os resultados.

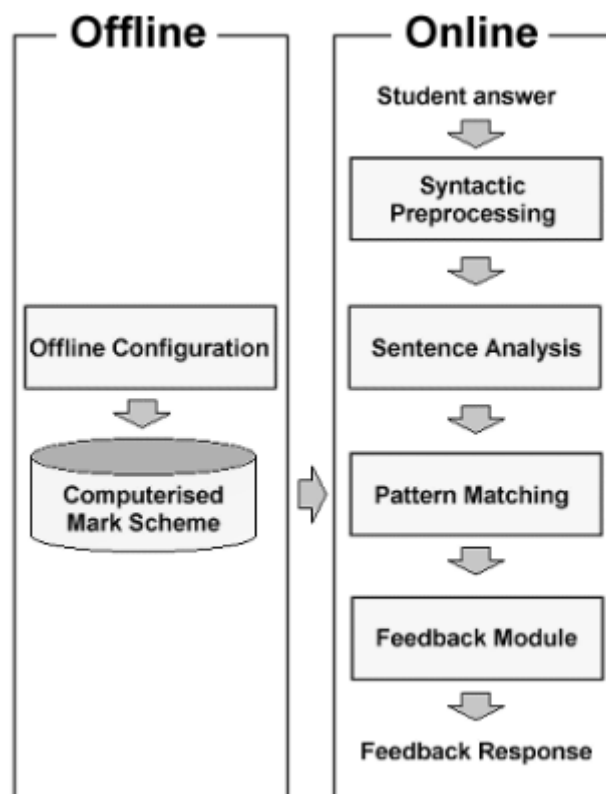


Figura 5 - Arquitectura do sistema

Foram efectuados testes ao sistema que resultaram numa correlação entre 93% e 96% na correcção de questões com quatro tipos de resposta: palavra única, valor único, explicação resumida e descrição de um padrão de dados [Marin, 2007].

3.3.7. *Paperless School free-text Marking Engine (PS-ME)*

PS-ME utiliza várias abordagens para fazer avaliações com pouca ou nenhuma intervenção humana [Mason & Grove-Stephenson, 2002]. O sistema foi desenvolvido para funcionar como um serviço de *back-end* para o sistema Learning Management System (LMS) com o intuito de libertar os professores para outras tarefas do ensino. Foi implementado como um componente de uma rede baseada em ambiente de aprendizagem. Funciona sobre o servidor LMS e devido às exigências de processamento não analisa os testes em tempo real.

Quando o aluno submete um teste, este é enviado para o servidor com a informação sobre a tarefa, a fim de serem identificados os textos de referência para a comparação. Cada tarefa é definida através de um conjunto de textos de referência relevantes para a questão que podem incluir textos de referência “falsos” que foram definidos como erros típicos dos alunos. Após esta selecção, o teste do aluno é comparado com os textos de referência para derivar alguns parâmetros que reflectem o conhecimento e compreensão do aluno relativamente à matéria.

PS-ME utiliza técnicas PLN e a avaliação é calculada através de uma análise linguística baseada em três taxonomias: conhecimento, entendimento e avaliação. O conhecimento sobre um determinado tema é indicado através do número de termos, objectos e entidades descritos correctamente. Se o aluno a responder a uma questão referir objectos, entidades ou conceitos que não estejam relacionados com o tema, pode-se concluir que a resposta não é muito relevante. Se pelo contrário o aluno utilizar um grande número de itens relevantes significa que o aluno possui conhecimento suficiente sobre a área o que implica que teria uma boa pontuação neste componente. Para avaliar o conhecimento é necessário identificar os conceitos no teste do aluno e avaliar a sua relevância na matéria. Isto pode ser conseguido com um grau elevado de concordância através de técnicas de PLN, sendo necessário comparar os conceitos relevantes com os conceitos do aluno. O entendimento pode ser separado em quatro categorias: compreensão, análise, aplicação e síntese as quais são divididas (como separar ou identificar componentes e reorganizar elementos). A avaliação pode ser definida através da contagem de adjectivos e advérbios. No entanto pode ser refinada tendo em conta três pontos: análise de padrões sintácticos, tendo em

consideração expressões do tipo “eu acho que” ou “é óbvio que”; contexto da avaliação, medindo a aproximação de um determinado subconjunto da frase com um conceito relevante e por fim, ponderar a correcção de cada avaliação, pois por vezes a opinião do aluno é valorizada pelo professor.

Este sistema tem sido utilizado na correcção de exames NCA e GCSE mas até ao momento não foram publicados resultados relativamente ao seu desempenho.

3.4. Análise comparativa das ferramentas

Apesar das ferramentas analisadas terem na sua essência o mesmo objectivo – correcção automática de respostas de texto livre - não é possível compará-las directamente pois estas usam métricas de avaliação diferentes. Para efectuar a análise de desempenho de cada uma das ferramentas foi necessário utilizar as métricas referenciadas pelos autores.

É importante referir que das oito ferramentas analisadas quatro baseiam-se em PLN, pelo que se pode concluir que esta técnica é utilizada com alguma frequência neste tipo de ferramentas.

Tabela 2 - Comparação dos sistemas de correcção automática analisados

Sistemas	Técnica (s)	Domínio	Disponibilidade	Desempenho
AutoMark	Extracção de Informação	Ciência	Académica	Correlação: 0.93 a 0.96
Betsy	Redes Bayesianas	Indiferente	Gratuita	Precisão: 0.80
C-Rater	PLN	Compreensão e Álgebra	Académica	Acordo: 0.80
E-Rater	PLN / Estatística	Exame GMAT	Académica	Acordo: 0.87 a 0.94
ETS I	PLN	Indiferente	Comercial	Acordo: 0.93 a 0.96
IEA	ASL	Exame GMAT, Psicologia, Medicina, História	Comercial	Acordo: 0.80 a 0.90
PEG	Características Linguísticas	Disciplinas não-factuais	Académica	Correlação: 0.87
PS-ME	PLN	Exame NCA ou GCSE	Comercial	---

A análise realizada na Tabela 2 tem por base os seguintes parâmetros:

- Técnica (s): apresenta as técnicas utilizadas por cada uma das ferramentas analisadas;
- Domínio: mostra a área em que cada ferramenta foi testada, que como é possível inferir, são as mais variadas;
- Disponibilidade: é interessante referir de que forma é que as aplicações estão disponíveis;
- Desempenho: Foram apresentadas as métricas de análise de performance usadas nas diferentes ferramentas: correlação da regressão múltipla (Correlação), precisão dos resultados (Precisão) e a taxa de acordo entre a avaliação atribuída pelo sistema e pelos professores (Acordo) [Valenti et al., 2003].

3.5. Conclusão

Ao longo deste capítulo foram analisadas várias técnicas e ferramentas de correcção automática de texto livre, mas convém salientar que como em todas as tecnologias existem vantagens e desvantagens. Uma grande vantagem destas ferramentas é sua utilidade pois há uma diminuição considerável do tempo despendido pelos professores na correcção de conteúdos para avaliação. Por outro lado, existem limitações como o nível de confiança depositada por parte dos avaliadores humanos neste tipo de soluções ou a abrangência dos conteúdos que podem ser analisados.

Foram analisadas algumas técnicas utilizadas nas ferramentas de correcção automática de texto livre de forma a permitir uma melhor compreensão da aplicação dessas técnicas. Através deste estudo é possível concluir que existem técnicas com diferentes graus de complexidade e que a utilização das mesmas pode depender dos objectivos que se pretendem atingir com o desenvolvimento das ferramentas.

Tendo em conta a análise efectuada, é possível concluir que as várias ferramentas que foram estudadas na generalidade têm um desempenho elevado. Visto que a comparação não pode ser efectuada de uma forma objectiva, é difícil destacar uma ferramenta como sendo a melhor. No entanto, é possível realçar que existem diversas técnicas que podem ser aplicadas em várias áreas. O ideal seria desenvolver uma ferramenta de correcção de respostas de texto livre que reunisse as vantagens de cada uma das técnicas estudadas.

4. Implementação e Avaliação

4.1. Introdução

Com o avanço das novas tecnologias existe cada vez mais a necessidade intrínseca da utilização de ferramentas de apoio à decisão. Para tal, foi desenvolvido um módulo de correcção de respostas de texto livre que permite auxiliar o professor na correcção de exercícios. Desta forma, a aplicação sugere uma cotação a cada uma das questões diminuindo consideravelmente o tempo despendido pelo professor na correcção.

Um dos principais objectivos a atingir com este módulo é o de ser uma ferramenta útil para os professores, permitindo-lhes uma maior eficácia na correcção. O tempo que seria despendido a corrigir alguns detalhes é bastante mais reduzido quando estas tarefas são efectuadas pela aplicação.

O módulo desenvolvido pretende proporcionar ao professor a possibilidade de utilizar além do corrector ortográfico, o mecanismo de correcção automática que indica a pontuação obtida por cada aluno nas diversas respostas de um exercício.

Nas secções seguintes será descrito de forma detalhada o processo de implementação e de avaliação do módulo desenvolvido. Inicialmente é feita uma contextualização relativamente à aplicação para a qual o módulo de correcção automática será desenvolvido. De seguida é efectuada a especificação dos actores que interagem com o sistema para este módulo em concreto. Posteriormente serão descritas as várias tarefas que constituem todo processo. Por fim, são apresentados alguns exemplos que permitem avaliar o desempenho do módulo desenvolvido.

4.2. Aplicação UNI_NET-Classroom

O projecto nasceu pela constatação da crescente necessidade da introdução de novas tecnologias na educação.

Esta necessidade surge das potencialidades que estas novas tecnologias oferecem em termos de envolvimento dos alunos: tornando as aulas mais interessantes, mais criativas, mais participativas e portanto melhorando o rendimento escolar. Além disso, foi introduzido

o conceito do Livro do Professor Digital cujo principal objectivo é transpor toda a informação relativa aos alunos, disciplinas, turmas e escola para um dispositivo de armazenamento digital.

O UNI_NET-Classroom é um programa que tem como principal objectivo facilitar a gestão de informação e a gestão da sala de aula pelo professor [UNI_NET-Classroom, 2011].

Para além da aplicação do professor (UNI_NET-Classroom) existe uma aplicação para o aluno (UNI_NET-Student) que permite a execução das tarefas solicitadas pelo professor e permite ao aluno seguir a sessão de modo interactivo.

O sistema é constituído por uma aplicação base para a qual foram desenvolvidos vários módulos, tal como exemplifica a Figura 6.

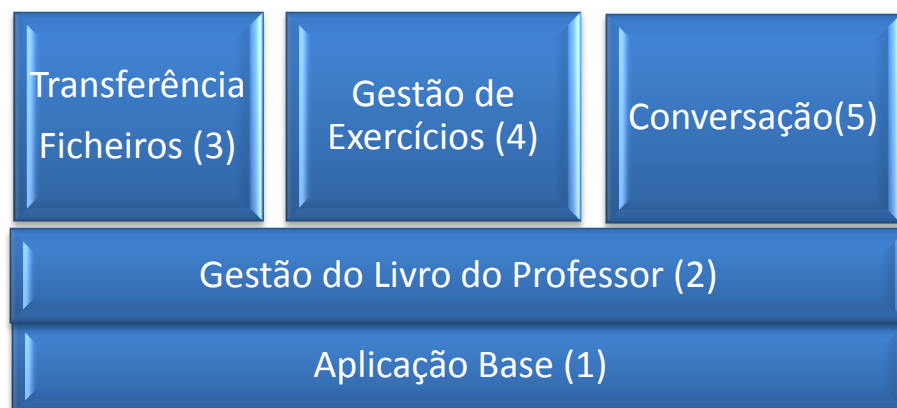


Figura 6 - Arquitectura da aplicação

O módulo de Gestão de Exercícios é aquele que será utilizado como base para a implementação. Este módulo permite criar exercícios que são posteriormente enviados para os alunos. Neste momento, após a recolha, a aplicação corrige as perguntas de escolha múltipla e mostra os resultados.

Para utilizar este módulo o professor tem de criar os exercícios a partir da aplicação. Através de um formulário, o professor adiciona ao exercício um número variável de questões que podem ser de desenvolvimento ou de escolha múltipla.

Este módulo também disponibiliza outras funcionalidades de gestão de exercícios: abrir e visualizar um exercício, remover um exercício; alterar/actualizar um exercício; envio de exercícios para a turma e correcção automática de respostas de escolha múltipla. Outra funcionalidade com bastante relevância é a demonstração de resultados que permite ao

professor analisar a cotação atribuída nas questões de escolha múltipla e corrigir as questões de resposta aberta a partir de um formulário que contém a informação de toda a turma. Além disso, também é possível enviar o exercício com as correcções para os Alunos para estes poderem analisar o seu desempenho.

4.3. Módulo de Correção de Respostas de Texto Livre

O módulo de correção de respostas de texto livre está integrado no módulo de Gestão de Exercícios. Com o desenvolvimento deste módulo, oferece-se ao utilizador uma ferramenta que auxilie no processo de correção de exercícios de texto livre. Assim sendo, o objectivo é potenciar a utilização deste módulo através da produção de resultados considerados interessantes e com utilidade para o professor.

Os resultados são disponibilizados de uma forma consideravelmente mais rápida, em relação à correção efectuada pelo professor, o que permite uma visão geral mais expedita do desempenho dos alunos num determinado exercício e liberta o professor para o esclarecimento/explicação aos alunos do seu desempenho. Em todo o caso, o professor pode rever e alterar a avaliação sugerida pelo sistema.

4.3.1. Actores que interagem com o Módulo

Neste módulo, bem como na aplicação em geral, cada actor tem as suas próprias funções e objectivos. Esta secção tem por objectivo especificar a interacção de cada um dos actores com este módulo específico do sistema.

4.3.1.1. Professor

O professor é o actor que mais interage com o sistema. A sua principal função é gerir todo o processo de gestão de exercícios e por isso é um elemento fundamental.

Por análise do Caso de Uso da Figura 7 é possível identificar as seguintes funcionalidades, que fazem parte das tarefas do professor:

- Criar Exercício: é a primeira tarefa do módulo de Gestão de Exercícios, e permite ao professor criar um exercício que pode conter respostas de escolha múltipla e respostas de texto livre;
- Adicionar correcção do Exercício: esta tarefa é essencial para o sistema fazer a correcção automática das questões e envolve o preenchimento das respostas correctas e a respectiva cotação;
- Enviar Exercício: quando o professor tem uma sessão de aula interactiva activa pode enviar um exercício para um ou vários Alunos que estejam ligados à mesma sessão, bastando para tal seleccionar o exercício pretendido;
- Recolher Exercício: após o envio de um exercício, e assim que o professor pretender executa a acção de recolher os exercícios e através desta acção o sistema inicia a correcção;
- Analisar/Corrigir Resultados: a aplicação abre automaticamente uma grelha com os resultados dos alunos que resolveram o exercício. A partir deste momento o professor pode analisar e rectificar os resultados sugeridos pelo sistema;
- Enviar Resultados: o professor pode enviar os resultados dos exercícios para os Alunos analisarem o seu desempenho;
- Exportar Resultados: é possível exportar os resultados da turma para um ficheiro de texto, de forma a permitir ao professor a gestão dos dados sem ser a partir da aplicação.

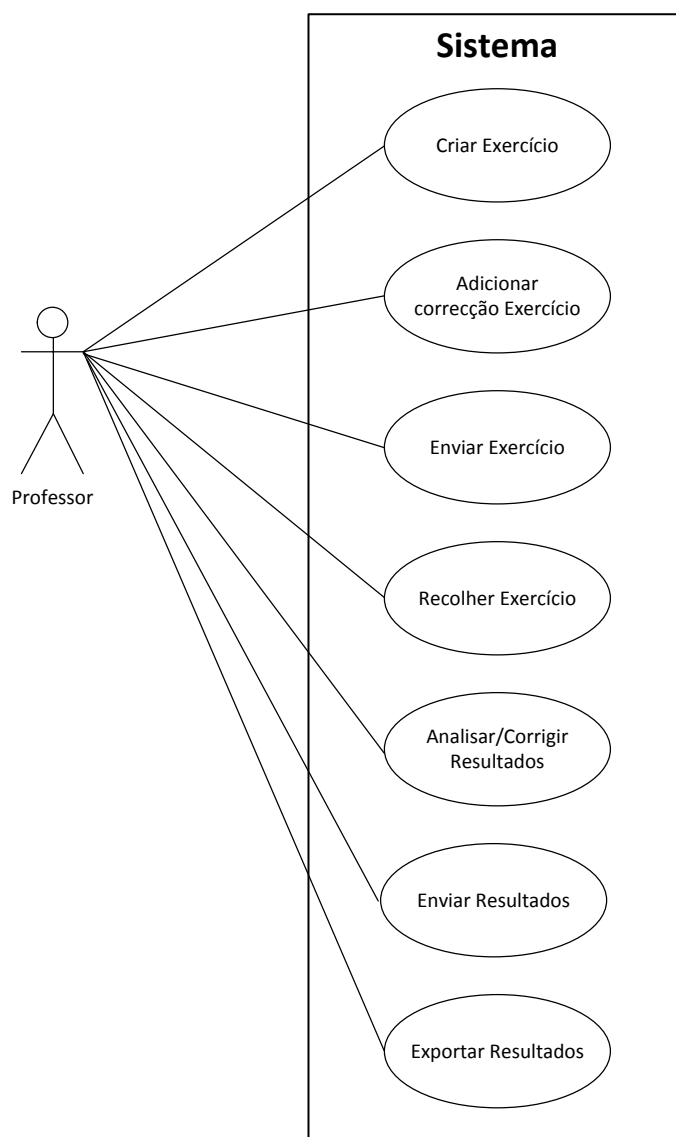


Figura 7 - Diagrama de casos de uso realizados pelo professor

4.3.1.2. Aluno

O aluno é um actor menos interventivo no sistema. Apenas reage às acções do sistema, sendo estas normalmente desencadeadas pelo professor.

Por análise do Caso de Uso da Figura 8 é possível identificar as seguintes funcionalidades, que fazem parte das tarefas do Aluno para este módulo:

- Resolver Exercício: o sistema recebe o exercício e disponibiliza-o através de um formulário para o Aluno o resolver;

- Guardar Exercício: se o Aluno terminar a resolução do exercício antes do professor recolher pode submetê-lo;
- Analisar Resultados: o Aluno recebe o exercício corrigido para analisar o seu desempenho na execução do mesmo.

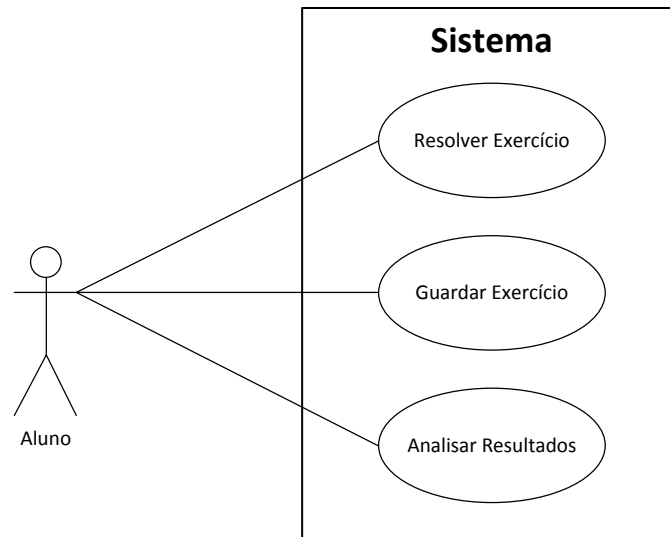


Figura 8 - Diagrama de casos de uso realizados pelo aluno

4.4. Interação entre as aplicações UNI_NET-Classroom e UNI_NET-Student

Sendo o módulo de gestão de exercícios o mais relevante para o trabalho desenvolvido, de seguida será exemplificado de que forma é que as aplicações UNI_NET-Classroom e UNI_NET-Student interagem quando é realizado um exercício.

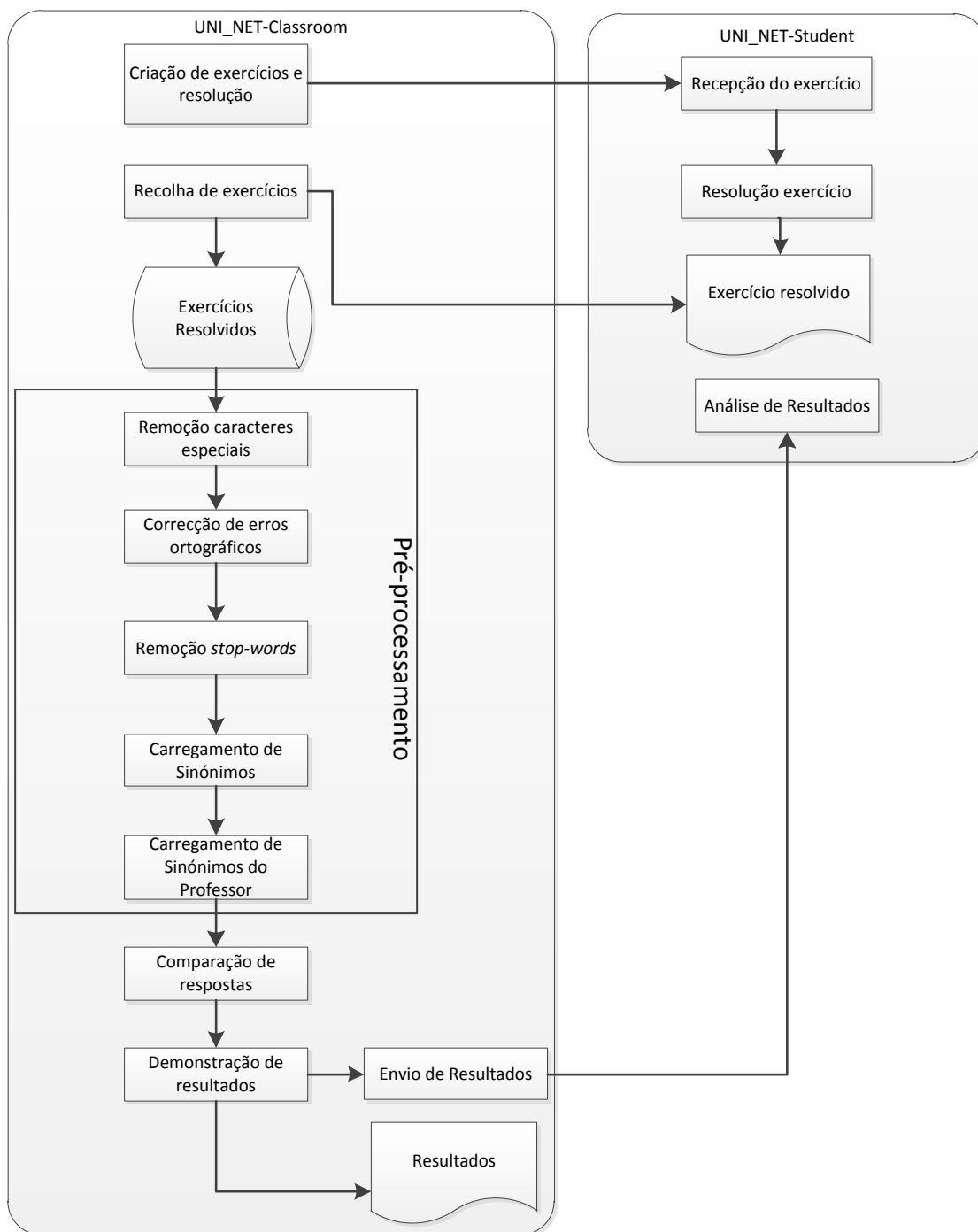


Figura 9 - Processo de elaboração e correcção de exercícios

Tal como ilustra a Figura 9, o processo inicia quando o professor elabora um exercício e respectiva correcção para posteriormente enviar para os Alunos. Quando o exercício é enviado a aplicação do Aluno recebe-o e visualiza-o para que este possa ser resolvido. Assim que terminado o tempo concedido para a resolução do exercício o professor recolhe os exercícios e a partir deste momento o sistema automaticamente começa a correcção. É

criada uma base de dados de respostas para tornar o processamento da informação mais rápido. Assim que toda a informação está disponível começam a ser executadas as tarefas de pré-processamento tanto para as respostas do professor como para as dos alunos. A primeira tarefa é a remoção de caracteres especiais que contempla remover a pontuação de todas as respostas. De seguida é feita a correcção ortográfica, utilizando a ferramenta JSpell e removidas as *stop-words*. Posteriormente são criadas várias respostas referência com base na resposta do professor, ou seja, é efectuado o carregamento de todos os sinónimos relativos à categoria das palavras usadas na resposta do professor. A necessidade de criar várias respostas referência prende-se com o facto da probabilidade do aluno utilizar as mesmas palavras do professor ser relativamente baixa e o aluno não deve ser penalizado pelo facto de não responder exactamente da mesma forma que o professor. Note-se que poderíamos também fazer o mesmo para as respostas dos alunos, mas tal iria tornar o processo de comparação de respostas computacionalmente muito intensivo, pelo que optamos só por criar variantes das respostas do professor, reduzindo-se assim a área de procura. Depois de efectuadas todas as tarefas de pré-processamento já é possível comparar as respostas. O processo de comparação de respostas consiste em contabilizar as palavras iguais e/ou os sinónimos existentes nas respostas dos alunos e do professor. No final, o professor pode visualizar a correcção das respostas e os resultados, e caso pretenda, editar os resultados. Assim que toda a informação estiver validada o professor pode enviar o exercício e respectiva correcção para os Alunos.

4.5. Implementação

4.5.1. Tecnologias utilizadas

O módulo foi desenvolvido na linguagem C++ [C++, 2011] e o ambiente de desenvolvimento foi o *Qt Creator* versão 4.6.2. [Qt Creator, 2011].

A linguagem C++ foi escolhida por ser independente do Sistema Operativo, orientada a objectos e pela existência de numerosas bibliotecas que facilitam o desenvolvimento de aplicações. Para além disso, esta linguagem foi utilizada no desenvolvimento da aplicação UNI_NET-Classroom e por uma questão de coerência decidiu-se manter.

A escolha desta ferramenta deveu-se ao facto desta ser intuitiva, gratuita e multiplataforma. Além disso, existe uma extensa comunidade de programadores que contribuem com exemplos bastante elucidativos e fáceis de integrar na aplicação.

Foram estudadas várias ferramentas de correcção ortográfica, das quais foi seleccionado o JSpell que foi integrado no módulo desenvolvido. A escolha desta ferramenta deveu-se a várias características da mesma, tal como explicado na secção 2.10.

4.5.2. Integração do JSpell na aplicação UNI_NET-Classroom

Para integrar o JSpell na aplicação UNI_NET-Classroom foi necessário utilizá-lo como uma biblioteca externa acedendo aos seus métodos através da interface. No entanto, esta ferramenta foi desenvolvida na linguagem C enquanto que o UNI_NET-Classroom foi desenvolvido na linguagem C++. Apesar da linguagem C++ ser a evolução da linguagem C, existem diferentes modos de interpretação e compilação pelo que foi necessário fazer algumas alterações na interface, e consequentemente no código fonte, para que esta fosse reconhecida pelo compilador. As duas linguagens diferem principalmente na assinatura dos métodos e nos tipos de dados que implicaram fazer as devidas alterações para permitir a integração.

Depois de ter o JSpell disponível a partir do UNI_NET-Classroom surgiram algumas dificuldades em detectar a razão pela qual a aplicação terminava sempre que invocava a biblioteca. Esta ferramenta verifica a existência do dicionário numa localização específica, e caso esta situação não se verifique o JSpell termina em tempo de execução. A solução foi enviar um sinal de aviso à aplicação, para que o utilizador seja alertado da ocorrência de um problema e assim manter a aplicação em execução.

Todos os erros detectados no JSpell foram corrigidos e a resolução foi enviada para os autores da ferramenta. Assim, deu-se uma contribuição para que as futuras versões possam ser integradas na aplicação UNI_NET-Classroom sem ter que alterar novamente o código.

Para a ferramenta JSpell estar disponível para vários idiomas a partir do UNI_NET-Classroom foi necessário criar uma interface que permite ao professor seleccionar o idioma utilizado na elaboração do exercício (tal como ilustra a Figura 10). O professor tem assim ao seu dispor os seguintes idiomas: Português, Português com acordo ortográfico, Inglês, Espanhol, Francês e Alemão.

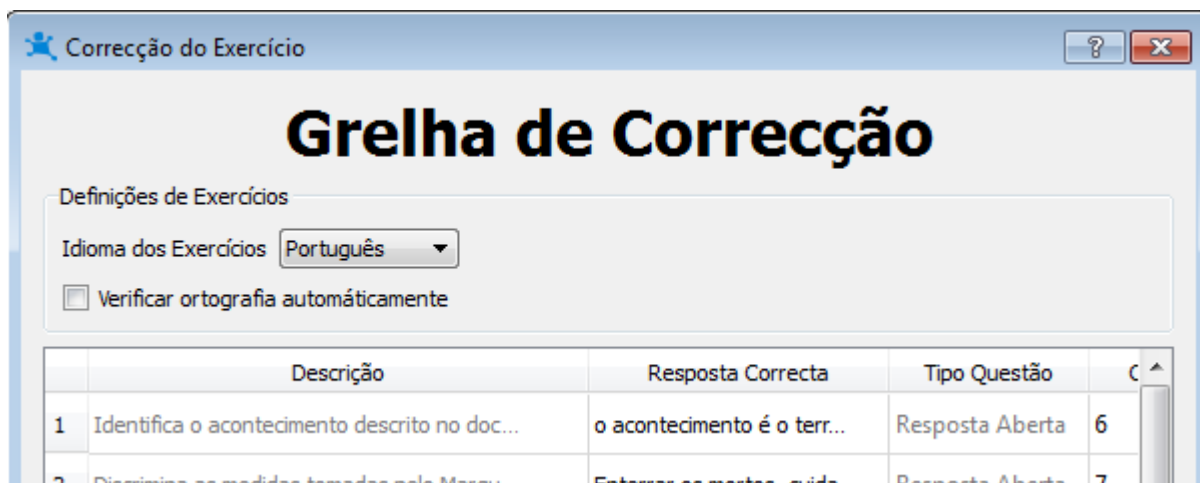


Figura 10 – Formulário de correcção de exercícios

4.5.3. Tarefas de pré-processamento

Para tornar as respostas passíveis de serem comparadas e consequentemente atribuir-lhes uma cotação foi necessário efectuar as seguintes tarefas de pré-processamento: remoção de caracteres especiais, correcção de erros ortográficos, remoção de *stop-words* e categorização das palavras. Estas tarefas têm como intuito transformar a resposta inicial num conjunto de termos que serão sujeitos a avaliação.

Os exercícios criados a partir da aplicação são guardados em ficheiros XML que são enviados para o aluno com o enunciado e quando são recolhidos já contêm as respostas. Para acelerar o acesso à informação relativa aos exercícios, estes são carregados a partir dos ficheiros XML do sistema, para uma estrutura de dados onde são guardados temporariamente em memória. Na Figura 11, exibe-se a abstracção UML da relação de classes que suportam a dinâmica de informação do exercício. Foram criadas quatro classes de forma a permitir o acesso aos dados de uma forma rápida. A primeira classe designada como “Exercício”, guarda a informação geral do exercício e o conjunto de respostas dos alunos e do professor. A classe “Resposta Aluno” tem como atributos chave o nome do aluno e a identificação da questão, que são utilizados na pesquisa das respostas. Cada elemento desta classe corresponde a uma resposta de um determinado aluno. Para guardar as respostas do professor foi criada a classe “Resposta Professor” que contém as respostas indicadas pelo professor. A classe “Palavra” que é utilizada tanto na classe “Resposta Aluno” como na classe “Resposta Professor” tem como intuito guardar informação relativa a uma palavra e representa cada elemento das listas de palavras das referidas classes.

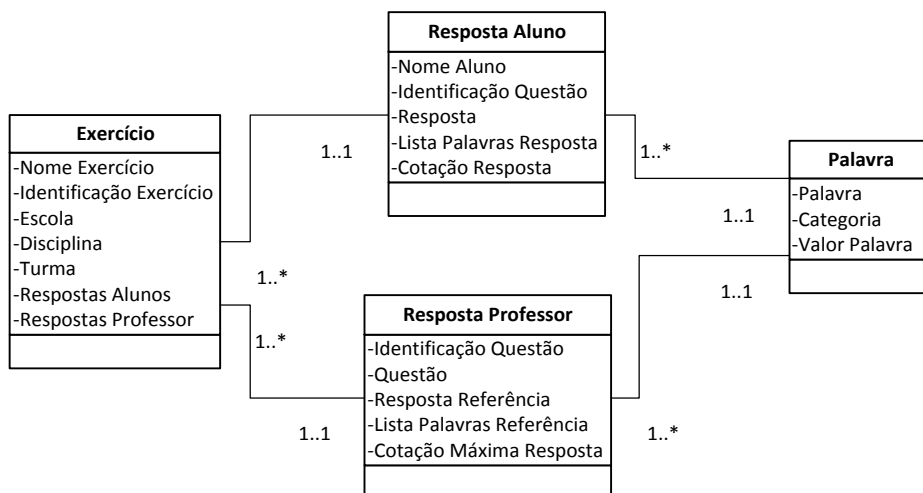


Figura 11 – Estrutura dinâmica com informação do exercício

Todas as tarefas de pré-processamento foram desenvolvidas de forma independente para tornar mais fácil a sua reorganização e combinação. Assim sendo, foi possível testar as diversas combinações e seleccionar a mais eficiente. Considerou-se que a melhor abordagem é aquela que no final confere ao sistema melhores resultados. Um caso prático que ilustra a relevância da ordem pela qual as tarefas são executadas, poderá ser o seguinte: se existir uma proposição com um erro ortográfico quando forem removidas as *stop-words* essa palavra não será considerada. Logo, é necessário corrigir primeiro os erros ortográficos e posteriormente retirar as *stop-words*.

4.5.3.1. Remoção de caracteres especiais

A primeira tarefa de processamento realizada consiste na remoção de caracteres especiais. Apenas foram considerados caracteres especiais aqueles que não estão integrados numa palavra, ou seja, a pontuação. A acentuação das palavras teve de ser mantida para que estas não sejam consideradas como erros pelo corrector ortográfico.

```

1: //Remove caracteres especiais - Este algoritmo remove toda a pontuação existente
na resposta
2: Entrada:
3: Saída:
4: //Remove os caracteres especiais das respostas de cada Aluno
5: Para cada Alunoi E ListaRespostas.ListaParametros Faz
6:     Resposta <- Alunoi.Resposta;
7:     //Esta função processa a Frase eliminando a pontuação
8:     //e passando todos os caracteres para minúsculas
9:     Resposta <- processaFrase(Resposta);
10:    //Separa a frase em palavras, utilizando os espaços entre as mesmas,
11:    //e agrega-as numa lista
12:    ListaPalavras <- separaPalavras(Resposta);
13:    Para cada Palavrai E ListaPalavras.ListaParametros Faz
14:        Alunoi.ListaPalavras <- Palavrai;
15:
16: //Remove os caracteres especiais das respostas do Professor
17: Para cada RespostaCorrectai E ListaRespostasProfessor.ListaParametros Faz
18:     RespostaCorrecta <- RespostaCorrectai.Resposta;
19:     //Esta função processa a Frase eliminando a pontuação
20:     //e passando todos os caracteres para minúsculas
21:     RespostaCorrecta <- processaFrase(RespostaCorrectai);
22:     //Separa a frase em palavras, utilizando os espaços entre as mesmas,
23:     //e agrega-as numa lista
24:     ListaPalavrasCorrectas <- separaPalavras(RespostaCorrectai);
25:     Para cada PalavraCorrectai E ListaPalavrasCorrectas.ListaParametros Faz
26:         RespostaCorrectai.ListaPalavrasCorrectas <- PalavraCorrectai;

```

Figura 12 - Algoritmo de remoção de caracteres especiais

Tal como é possível visualizar na Figura 12, o algoritmo de remoção de caracteres especiais tem como intuito a remoção de caracteres especiais das respostas dos alunos e do professor. Por conseguinte, é necessário percorrer todas as respostas e sempre que for detectada pontuação, esta é removida. Em simultâneo, todos os caracteres são alterados para minúsculas de forma a facilitar a comparação. Sendo esta a primeira tarefa de pré-processamento, é neste momento que é criada para cada resposta uma lista de palavras para facilitar o processamento.

4.5.3.2. Correção de erros ortográficos

Tal como foi referido anteriormente, o corrector utilizado para a correção de erros ortográficos foi o JSpell. Todas as palavras das respostas do professor e dos alunos são verificadas e tendo erros são corrigidas. Além de detectar os erros ortográficos a ferramenta JSpell sugere a correção. Assim sendo as palavras com erros são substituídas pela correção que é sugerida.

```

1: //Corrigir erros das respostas - Este algoritmo detecta e corrige os erros
2: //ortograficos das palavras, utilizando para tal o corrector ortográfico JSpell.
3: Entrada:
4: Saída:
5: //Percorre todas as respostas de cada Aluno e corrige os erros ortograficos
6: Para cada Alunoi E ListaRespostas.ListaParametros Faz
7:     //Necessita de voltar a construir a frase para enviar ao corrector
8:     //ortografico como parametro
9:     Resposta <- agregaPalavras(Alunoi.ListaPalavras) ;
10:    Resposta = CorrectorOrtográfico.detectarErrosECorrigir(Resposta);
11:    Alunoi.Erros = CorrectorOrtográfico.contagemErros();
12:    //Separa a frase em palavras, utilizando os espaços entre as mesmas,
13:    //e agrega-as numas lista
14:    ListaPalavras <- separaPalavras(Resposta);
15:    Para cada Palavrai E ListaPalavras.ListaParametros Faz
16:        Alunoi.ListaPalavras <- Palavrai;
17:
18: //Percorre todas as respostas do Professor e corrige os erros ortograficos
19: Para cada RespostaCorrectai E ListaRespostasProfessor.ListaParametros Faz
20:     //Necessita de voltar a construir a frase para enviar ao corrector
21:     //ortografico como parametro
22:     Resposta <- agregaPalavras(RespostaCorrectai.ListaPalavras) ;
23:     Resposta = CorrectorOrtográfico.detectarErrosECorrigir(Resposta);
24:     //Separa a frase em palavras, utilizando os espaços entre as mesmas,
25:     //e agrega-as numas lista
26:     ListaPalavras <- separaPalavras(Resposta);
27:     Para cada Palavrai E ListaPalavras.ListaParametros Faz
28:         RespostaCorrectai.ListaPalavras <- Palavrai;

```

Figura 13 - Algoritmo de correcção de erros ortográficos

Através da Figura 13, pretende-se exemplificar de que forma é que é efectuada a correcção de erros ortográficos. Para cada uma das respostas, tanto dos alunos como do professor, é necessário agregar numa *string* as palavras que constituem a lista de palavras das respostas já sem pontuação de forma a poder ser enviada para o corrector ortográfico como parâmetro de entrada. O parâmetro de saída do corrector ortográfico é a resposta com os erros substituídos pela possível correcção. Além disso, também são contabilizados os erros encontrados, que poderá servir como penalização caso o professor assim tenha definido. No final, a lista de palavras é actualizada, de forma a poder ser utilizada na próxima tarefa.

4.5.3.3. Remoção de *stop-words*

Uma *stop-word* é uma palavra que é considerada irrelevante, porque não afecta o significado semântico da resposta. O objectivo desta tarefa é a remoção de todas as *stop-words* que não têm qualquer significado para a cotação da resposta. Por exemplo, preposições e artigos que fazem a ligação entre os termos das frases.

Convém no entanto referir que nem todas as *stop-words* são usadas, por exemplo, palavras como “Sim” ou “Não” são também consideradas *stop-words*, mas podem alterar totalmente o significado de uma resposta, e portanto, mudar a cotação que lhe é dada, por essa razão estas palavras têm um tratamento especial e não são removidas nesta fase.

Resumindo, esta tarefa tem como intuito remover *stop-words* para apenas serem contabilizadas as palavras que têm realmente valor semântico na resposta. Para tal, as *stop-words* são carregadas a partir de um ficheiro de texto [Stop-words, 2010] para uma estrutura dinâmica de forma a facilitar a comparação com todas as palavras das respostas do professor e dos alunos.

```
1: //Remove "stop words" - Este algoritmo retira as "stop-words" das respostas
2: //dos Alunos e do Professor
3: Entrada:
4: Saída:
5: //Carrega as "stop words" de um ficheiro de texto para uma lista
6: ListaStopWords <- carregaStopWordsFicheiro();
7: //Remove as "stop words" das respostas de cada Aluno
8: Para cada Alunoi E ListaRespostas.ListaParametros Faz
9:     Para cada Palavrai E ListaPalavras.ListaParametros Faz
10:         Para cada StopWordi E ListaStopWords.ListaParametros Faz
11:             Se Palavrai = StopWordi Faz
12:                 ListaPalavras <- removeStopWord(ListaPalavras);
13:
14: //Remove as "stop words" das respostas do Professor
15: Para cada RespostaCorrectai E ListaRespostasProfessor.ListaParametros Faz
16:     Para cada PalavraCorrectai E ListaPalavrasCorrectas.ListaParametros Faz
17:         Para cada StopWordi E ListaStopWords.ListaParametros Faz
18:             Se PalavraCorrectai = StopWordi Faz
19:                 ListaPalavrasCorrectas <- removeStopWord(ListaPalavrasCorrectas);
```

Figura 14 - Algoritmo de remoção de *stop-words*

O algoritmo de remoção de *stop-words*, tal como ilustra a Figura 14, percorre todas as respostas tanto dos alunos como do professor, e compara cada uma das palavras com a lista de *stop-words*, que foram carregadas a partir do ficheiro. Sempre que a comparação for verdadeira, a *stop-word* é removida da lista de palavras.

4.5.3.4. Categorização das palavras

Esta tarefa tem como objectivo uniformizar as palavras e categorizá-las quanto ao seu género. Assim será possível comparar apenas as palavras que pertencem à mesma categoria e detectar os respectivos sinónimos. Foi utilizado o corrector ortográfico nesta

tarefa, pois este devolve a palavra na sua forma canónica. Como restrição detectou-se o facto de JSpell retornar várias hipóteses para a mesma palavra optando-se por utilizar a primeira que é sugerida. Esta decisão baseou-se em testes efectuados, em que na maioria dos casos a primeira palavra era a mais correcta.

```
1: //Adicionar categoria as palavras - Este algoritmo faz a etiquetagem da categoria
2: //a todas as palavras
3: Entrada:
4: Saída:
5: //Percorre todas as respostas de cada Aluno
6: Para cada Alunoi E ListaRespostas.ListaParametros Faz
7:     Para cada Palavrai E ListaPalavras.ListaParametros Faz
8:         Resultado <- CorrectorOrtográfico.categorizarPalavra(Palavrai);
9:         //O corrector ortografico retorna uma string com a informação relativa
10:        //a palavra. Essa string tem de ser tratada para se retirar a categoria
11:        PalavraComCategoria <- trataResultado(Resultado);
12:        //Acrescenta a palavra com a categoria na lista de palavras, removendo
13:        //a palavra que existia anteriormente
14:        substituiPalavraListaPalavras (PalavraComCategoria);
15:
16: //Percorre todas as respostas do Professor
17: Para cada RespostaCorrectai E ListaRespostasCorrectas.ListaParametros Faz
18:     Para cada PalavraCorrectai E ListaPalavrasCorrectas.ListaParametros Faz
19:         Resultado <- CorrectorOrtográfico.categorizarPalavra(PalavraCorrectai);
20:         //O corrector ortografico retorna uma string com a informação relativa
21:        //a palavra. Essa string tem de ser tratada para se retirar a categoria
22:        PalavraCorrectaComCategoria <- trataResultado(Resultado);
23:        //Acrescenta a palavra com a categoria na lista de palavras, removendo
24:        //a palavra que existia anteriormente
25:        substituiPalavraListaPalavrasCorrectas (PalavraCorrectaComCategoria);
```

Figura 15 - Algoritmo de categorização das palavras

Através da Figura 15, pretende-se demonstrar de que forma é que é efectuada a categorização das palavras. Nesta tarefa, é enviada como parâmetro de entrada para o corrector ortográfico a palavra que se pretende categorizar e este por sua vez retorna informação relativamente à palavra da qual é necessário extrair a sua forma canónica e a categoria. No final, a palavra é actualizada na lista e para além disso é acrescentada a sua categoria.

4.5.3.5. Carregamento de sinónimos

O dicionário de sinónimos é essencial para fazer uma comparação mais coerente e abrangente das respostas visto que a probabilidade do aluno utilizar as mesmas palavras do professor é relativamente baixa. Um aluno não deve ser penalizado pelo facto de não

responder exactamente da mesma forma que o professor e por isso é necessário ter em consideração se o aluno utilizou um sinónimo da palavra da resposta de referência.

O projecto OpenThesaurusPT [OpenThesaurus, 2010] disponibiliza um ficheiro de sinónimos para a Língua Portuguesa que será utilizado no processo de correcção desenvolvido.

Como já se referiu, são criadas várias respostas referência relativas a cada resposta do professor. Isto é essencial para fazer uma comparação mais coerente e abrangente das respostas, visto que a probabilidade do aluno utilizar as mesmas palavras do professor é relativamente baixa. As variações das respostas referência são criadas associando-se a cada palavra da resposta do professor e respectiva categoria a lista de sinónimos associados a essa categoria de palavra.

O dicionário de sinónimos é fundamental para esta operação tendo-se adoptado o dicionário do projecto OpenThesaurusPT [OpenThesaurus, 2010] que disponibiliza um ficheiro de sinónimos para a Língua Portuguesa e será utilizado no processo de correcção desenvolvido.

Todos os sinónimos são carregados para uma estrutura de dados a partir de um ficheiro de texto. Cada elemento da estrutura contém palavras que são sinónimas entre si.

A partir da estrutura de dados que contém todos os sinónimos, é criada uma nova estrutura que contém apenas as palavras sinónimas da lista de palavras da resposta do professor. Desta forma reduz-se bastante a área de procura, no momento da comparação das respostas tornando o processo de correcção mais rápido.

```
1: //Carregar sinonimos do Professor - Este algoritmo carrega os sinonimos a partir
2: //de um ficheiro de texto para uma estrutura, filtrando apenas os que constam
3: //nas respostas do Professor
4: Entrada:
5: Saída:
6: //Carrega os sinonimos do ficheiro de texto para uma estrutura em que cada
7: //posição contém uma lista de palavras sinonimas
8: EstruturaSinonimos <- carregaSinonimosFicheiro();
9: //Percorre a estrutura de sinonimos geral para criar uma estrutura mais pequena
10: //so com as palavras existentes nas respostas do Professor.
11: Para cada ListaSinonimosi E EstruturaSinonimos.ListaParametros Faz
12:     Para cada RespostaCorrectai E ListaRespostasProfessor.ListaParametros Faz
13:         Para cada PalavraCorrectai E ListaPalavrasCorrectas.ListaParametros Faz
14:             Se ListaSinonimosi.contem(PalavraCorrectai) Faz
15:                 //Adiciona as listas que contem palavras utilizadas pelo Professor
16:                 //a uma estrutura mais pequena
17:                 adicionaListaAEstruturaSinonimosProfessor(ListaSinonimosi);
```

Figura 16 - Algoritmo de carregamento de sinónimos

O algoritmo de carregamento de sinónimos que está representado na Figura 16 tem como intuito carregar todos os sinónimos de um ficheiro de texto e filtrá-los com base nas palavras das respostas do professor. Assim, é criada uma lista com as palavras que poderão ser utilizadas na pesquisa de sinónimos nas respostas dos alunos.

4.5.3.6. Exemplo de pré-processamento de uma resposta

Para tornar mais perceptível o processo de correcção automática, será ilustrada a evolução de uma resposta ao longo do processo de correcção.

No exemplo da Figura 17 pretende-se demonstrar com um exemplo prático a transformação de uma resposta em cada uma das tarefas de pré-processamento.

Foi utilizada uma resposta dada pelo professor pelo que a tarefa de correcção de erros ortográficos não acrescentou nenhuma alteração relativamente à tarefa de remoção de caracteres especiais.

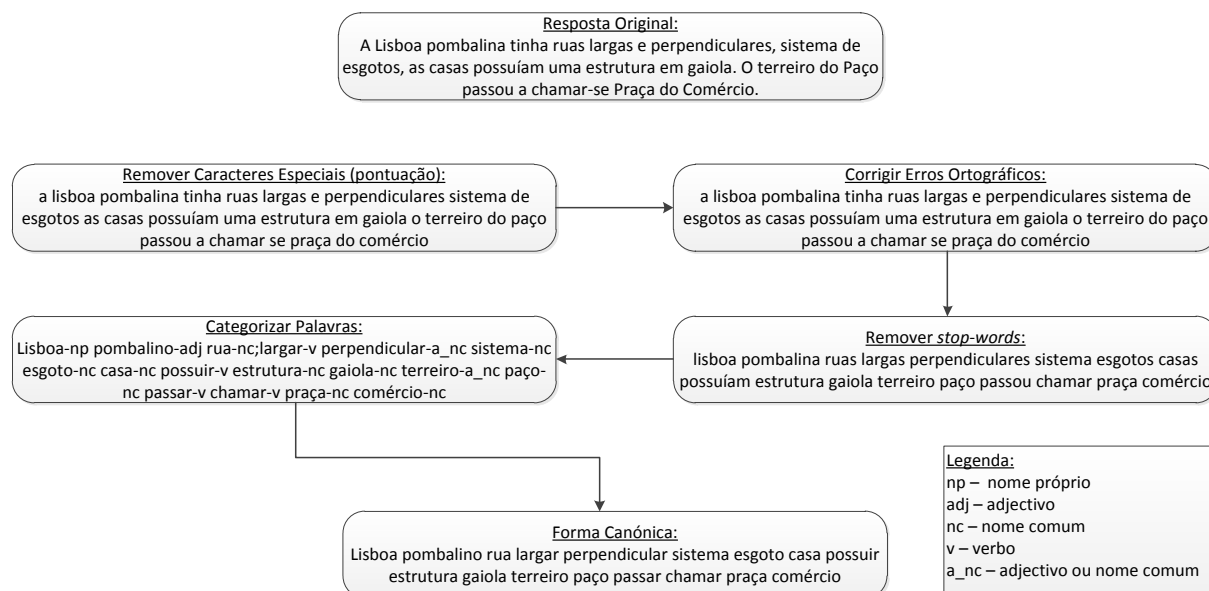


Figura 17 - Evolução da resposta à pergunta “Caracteriza o Urbanismo Pombalino.”

Tal como está representado na Figura 17, o processo inicia com a resposta no seu estado original, e a primeira tarefa é remover os caracteres especiais mais especificamente a pontuação. De seguida a resposta é submetida ao corrector ortográfico para que sejam corrigidos todos os erros e para além disso são sugeridas sugestões para a sua correcção. A próxima tarefa consiste na remoção das *stop-words*, que remove todas as palavras que não devem ter significado na correcção da resposta. Por fim, é efectuada a categorização das palavras que atribui a categoria às palavras e substituí-as pela sua forma canónica.

4.5.4. Comparação de respostas e atribuição da cotação

Após o pré-processamento, já é possível comparar as respostas dos alunos com as respostas de referência. Em ambos os casos é efectuada todo o processo de pré-processamento das respostas e por isso mesmo, neste momento, é necessário calcular o peso mais adequado para cada uma das palavras-chave das respostas do professor e só depois então proceder à comparação.

A primeira abordagem para a obtenção de resultados foi a aplicação da técnica VSM (Vector Space Model), mas visto que os resultados obtidos divergiram substancialmente da avaliação atribuída pelo professor foi necessário efectuar alguns ajustes ao processo de forma a adaptá-lo a este contexto. O objectivo de VSM é calcular a similaridade entre textos através da representação dos textos em vectores. Para aplicar este algoritmo não é necessário efectuar tarefas de pré-processamento, pois por si só, esta abordagem não atribui qualquer valor às palavras que aparecem em todas as respostas. De seguida, serão exemplificados todos os cálculos necessários na implementação desta técnica.

O peso de cada palavra é obtido a partir da seguinte fórmula:

$$PesoPalavra = W_i = tf_i * \log\left(\frac{D}{df_i}\right)$$

Onde, tf_i corresponde ao número de vezes que o termo aparece na resposta, D é o número total de respostas e df_i representa o número de respostas em que o termo aparece.

Todas as respostas, incluindo as do professor, são representadas por vectores. De seguida, é necessário calcular o tamanho desses vectores.

$$TamanhoVector = \sqrt{(termo_1^2 + termo_2^2 + .. + termo_n^2)}$$

Assim que forem obtidos os tamanhos de todos os vectores, procede-se ao cálculo do produto vectorial entre o vector de cada uma das respostas dos alunos com a resposta referência.

$$ProductoVectorial = \sum_{i=1}^n Vector_{Aluno}^i * Vector_{Professor}^i$$

Para calcular a cotação da resposta, é necessário calcular a similaridade entre a resposta do aluno e a do professor. A similaridade é calculada com base na seguinte fórmula:

$$Similaridade = \frac{ProductoVectorial}{|TamanhoVector_{Professor}| * |TamanhoVector_{Aluno}|}$$

A aplicação do algoritmo VSM sugere que não seja efectuada qualquer tarefa de pré-processamento, pois assim as palavras que aparecem menos vezes são consideradas mais relevantes no cálculo do peso das palavras. No caso específico das respostas dos alunos, uma palavra que aparece com menor frequência pode estar simplesmente errada e por isso não deve ser mais valorizada. De forma a contornar esta situação, é efectuado em primeiro lugar todo o processo de pré-processamento, que remove todas as palavras que não devem ser contabilizadas na avaliação. Assim, pode-se considerar que as palavras que aparecem com maior frequência devam ter um peso superior.

Tendo em consideração estas alterações, o peso da palavra do professor é calculado, agora, a partir da seguinte fórmula:

$$PesoPalavra = \frac{TotalPalavraRespostasProfessor}{TotalQuestões * TotalPalavraRespostas} * \log(TotalQuestões * TotalPalavraRespostas)$$

Onde, *PesoPalavra* é o valor de contribuição de cada palavra para a cotação final da resposta, *TotalPalavraRespostasProfessor* é o número de vezes que a palavra aparece em todas as respostas do professor, *TotalQuestões* é o número de questões do teste e por fim o *TotalPalavraRespostas* que indica o número de vezes que a palavra aparece em todas as respostas (frequência da palavra).

De seguida é necessário aplicar as fórmulas do algoritmo VSM, que foram anteriormente descritas, para calcular a similaridade entre a resposta do professor e do aluno. A partir do valor de similaridade já é possível calcular a cotação da resposta:

$$CotaçãoResposta = SimilaridadeResposta * CotaçãoMáxima$$

Por fim, procede-se ao cálculo da cotação final da resposta que será atribuída ao aluno para a resposta em questão:

$$CotaçãoFinalResposta = CotaçãoResposta - \left(\frac{TotalErrosResposta}{TotalPalavrasResposta} * \alpha \right)$$

A cotação final da resposta é calculada através da subtração da penalização dos erros ortográficos do aluno na resposta ao valor da cotação obtido anteriormente. Essa penalização é influenciada pelo valor de α que for definido pelo professor na grelha de correcção.

De forma a demonstrar todo o processo de atribuição da cotação a uma resposta, é apresentado de seguida, um exemplo da aplicação das fórmulas a uma resposta dada por um aluno. Este exemplo tem por base as seguintes premissas:

- Questão: Identifica o acontecimento descrito no documento número um;
- Resposta professor: acontecimento terramoto Lisboa;
- Resposta aluno: acontecimento descrito documento terramoto;
- Cotação Máxima Questão (*CotaçãoMáxima*): 6 pontos;
- Penalização erros ortográficos (α): 0 pontos.

1. Calcular o peso das palavras da resposta do professor:

$$PesoPalavra_{acontecimento} = 1 * \log(17 * 21) = 5.88$$

$$PesoPalavra_{terramoto} = 1 * \log(17 * 25) = 6.05$$

$$PesoPalavra_{lisboa} = 1 * \log(17 * 14) = 5.47$$

2. Calcular o peso das palavras do aluno:

$$PesoPalavra_{acontecimento} = 1 * \log(17 * 21) = 5.88$$

$$PesoPalavra_{terramoto} = 1 * \log(17 * 25) = 6.05$$

3. Calcular tamanho dos vectores:

$$TamanhoVector_{Professor} = \sqrt{(5.88^2 + 6.05^2 + 5.47^2)} = 10.05$$

$$TamanhoVector_{Aluno} = \sqrt{(5.88^2 + 6.05^2)} = 8.44$$

4. Calcular o produto vectorial:

$$ProductoVectorial = (5.88 * 5.88 + 6.05 * 6.05 + 5.47 * 0) = 71.18$$

5. Calcular a similaridade:

$$Similaridade = \frac{71.18}{10.05 * 8.44} = 0.84$$

6. Calcular a cotação resposta:

$$CotaçãoResposta = 0.84 * 6 = 5.04$$

7. Calcular a cotação final:

$$CotaçãoFinalResposta = 5.04$$

Neste exemplo o alfa é zero, pois o professor não atribuiu qualquer penalização aos erros logo a cotação final da resposta é igual à cotação da resposta.

De seguida será apresentado o algoritmo que descreve o processo de comparação de respostas.

```

1: //Comparação de respostas - Este algoritmo compara todas respostas dos alunos
2: //com as respostas de referência indicadas pelo professor, atribuindo-lhes
3: //uma cotação
4: Entrada:
5: Saída:
6: //Calculo do peso das palavras das respostas do professor
7: Para cada RespostaCorrectai E ListaRespostasProfessor.ListaParametros Faz
8:   ValorAuxiliar = 0
9:   PesoPalavra = 0
10:  Para cada PalavraCorrectai E ListaPalavras.ListaParametros Faz
11:    TotalPalavraRespostasProfessor = contarPalavra(RespostaCorrectai,
12:      PalavraCorrectai)
13:    PesoPalavra = TotalPalavraRespostasProfessor *
14:      log(NúmeroQuestões * FrequênciaPalavra(PalavraCorrectai)
15:    PalavraCorrectai.PesoPalavra = PesoPalavra
16:    ValorAuxiliar = PesoPalavra * PesoPalavra
17:    RespostaCorrectai.tamanhoVector = raizQuadrada(ValorAuxiliar)
18:
19: //Calculo do peso das palavras das respostas dos alunos
20: Para cada RespostaAlunoi E ListaRespostas.ListaParametros Faz
21:   ValorAuxiliar = 0
22:   PesoPalavra = 0
23:   Para cada Palavrai E ListaPalavras.ListaParametros Faz
24:     TotalPalavraRespostasProfessor = contarPalavra(RespostaAlunoi, Palavrai)
25:     PesoPalavra = TotalPalavraRespostasProfessor *
26:       log(NúmeroQuestões * FrequênciaPalavra(Palavrai)
27:     Palavrai.PesoPalavra = PesoPalavra
28:     ValorAuxiliar = PesoPalavra * PesoPalavra
29:     RespostaAlunoi.tamanhoVector = raizQuadrada(ValorAuxiliar)
30:
31: //Análise Similaridade
32: Para cada RespostaAlunoi E ListaRespostas.ListaParametros Faz
33:   Para cada RespostaCorrectai E ListaRespostasProfessor.ListaParametros Faz
34:     //Produto Vectorial - Palavras iguais e sinónimos
35:     Produto = 0
36:     Se RespostaAlunoi.IdQuestão = RespostaCorrectai.IdQuestão
37:       Para cada Palavrai E ListaPalavras.ListaParametros Faz
38:         Para cada PalavraCorrectai E ListaPalavrasProfessor.ListaParametros Faz
39:           Se Palavrai.Palavra = PalavraCorrectai.Palavra
40:             Produto = Produto + (Palavrai.PesoPalavra *
41:               PalavraCorrectai.PesoPalavra)
42:       Senão
43:         Para cada Sinónimosi E ListaSinónimos.ListaParametros Faz
44:           Se Sinónimosi.contém(Palavrai.Palavra)
45:             Produto = Produto + (Palavrai.PesoPalavra *
46:               PalavraCorrectai.PesoPalavra)
47:
48:     //Calculo Similaridade
49:     Similaridade = Produto / (RespostaAlunoi.tamanhoVector *
50:       RespostaCorrectai.tamanhoVector)
51:
52:     //Calculo cotação aluno
53:     CotaçãoAluno = (Similaridade * RespostaCorrectai.CotaçãoMáxima) -
54:       (RespostaAlunoi.erros / RespostaAlunoi.totalPalavras * alfa)

```

Figura 18 – Algoritmo de comparação das respostas

Tal como ilustra a Figura 18, o algoritmo de comparação de respostas é composto por várias fases:

1. Em primeiro lugar, é necessário atribuir um peso a cada uma das palavras-chave da resposta. Desta forma, cada palavra passa a ter um peso adequado à sua importância na resposta;
2. Após a atribuição dos pesos, é necessário adequá-los ao valor máximo da questão, para que estes tenham significado quando forem acrescentados ao valor da resposta;
3. De seguida, são percorridas todas as respostas dos alunos, e para cada uma delas é verificado se a palavra ou um sinónimo desta existe na resposta do professor. Se existir, o valor da palavra é incrementado à cotação da resposta;
4. Por fim, é calculado o valor final que será atribuído à questão. À cotação da resposta obtida no ponto anterior é subtraída uma penalização dos erros ortográficos, caso o professor pretenda penalizar os erros ortográficos.

Esta tarefa, de comparação da resposta do aluno/professor é a mais relevante de todo o processo, pois todas as tarefas de pré-processamento efectuadas têm como intuito melhorar o desempenho da comparação de respostas.

4.5.5. Disponibilização dos resultados

Para o professor visualizar os resultados obtidos pela correcção automática, assim que a aplicação termina a correcção, é aberto um formulário com uma tabela que contém os resultados de todos os alunos que participaram no exercício (tal como ilustra a Figura 19).

O formulário disponibilizado contém, para além da cotação total do exercício, o valor atribuído a cada uma das questões, dando também informação relativamente à cotação máxima, cotação mínima, média, mediana e moda de todas as respostas.

Resultados da Turma

Seleccionar Exercício: **Teste_8A**

Cotação Máxima: **100**

	Resultado	Cotação Obtida	Questão 1.1	Questão 1.2	Questão 1.3
Alexandre Sá	Não Satisfaz	49	6	3	0
André Simões de Sousa	Satisfaz	56	5	7	0
Andreia Resende	Não Satisfaz	44	3	6	0
Catarina Costa Ferreira	Satisfaz	61	6	5	0
Cristiana Gomes	Satisfaz	55	5	6	0
Edgar Rocha da Costa	Não Satisfaz	39	6	2	2
Fábio Miguel Dias Marques	Não Satisfaz	42	6	5	0
Filipe Vieira da Silva	Satisfaz	50	5	2	0
Heliodoro Ilha da Costa	Satisfaz	62	6	4	4
Maria da Paz Sousa Meneses	Não Satisfaz	46	3	5	0
Marlene Barreto	Satisfaz	71	6	0	0
Matilde Soares Ribeiro	Bom	75	6	0	7
Noémia Costa	Satisfaz	56	6	2	0
Ofélia Rego Pereira	Satisfaz	74	5	0	6
Paulo Alexandre de Sá Peixoto	Satisfaz	56	5	6	0
Pedro Gonçalves	Satisfaz	54	6	4	2
Valdemar Sousa	Não Satisfaz	48	5	6	0
Cotação Máxima	0	75	6	7	7
Cotação Mínima	0	39	3	0	0
Média	0	55.18	5.29	3.71	1.24
Mediana	0	55	6	4	0
Moda	0	48	5	6	0

Figura 19 - Formulário de demonstração de resultados da aplicação UNI_NET-Classroom

Para cada aluno, o professor poderá editar qualquer uma das cotações, excepto as respostas de escolha múltipla que não são editáveis pois só existe uma resposta correcta indicada pelo professor. Para as respostas de texto livre, o professor pode alterar directamente no formulário, ou então, abrir o exercício do aluno, que contém todas as

perguntas e respectivas respostas, e verificar se a cotação sugerida corresponde às suas expectativas. Através da

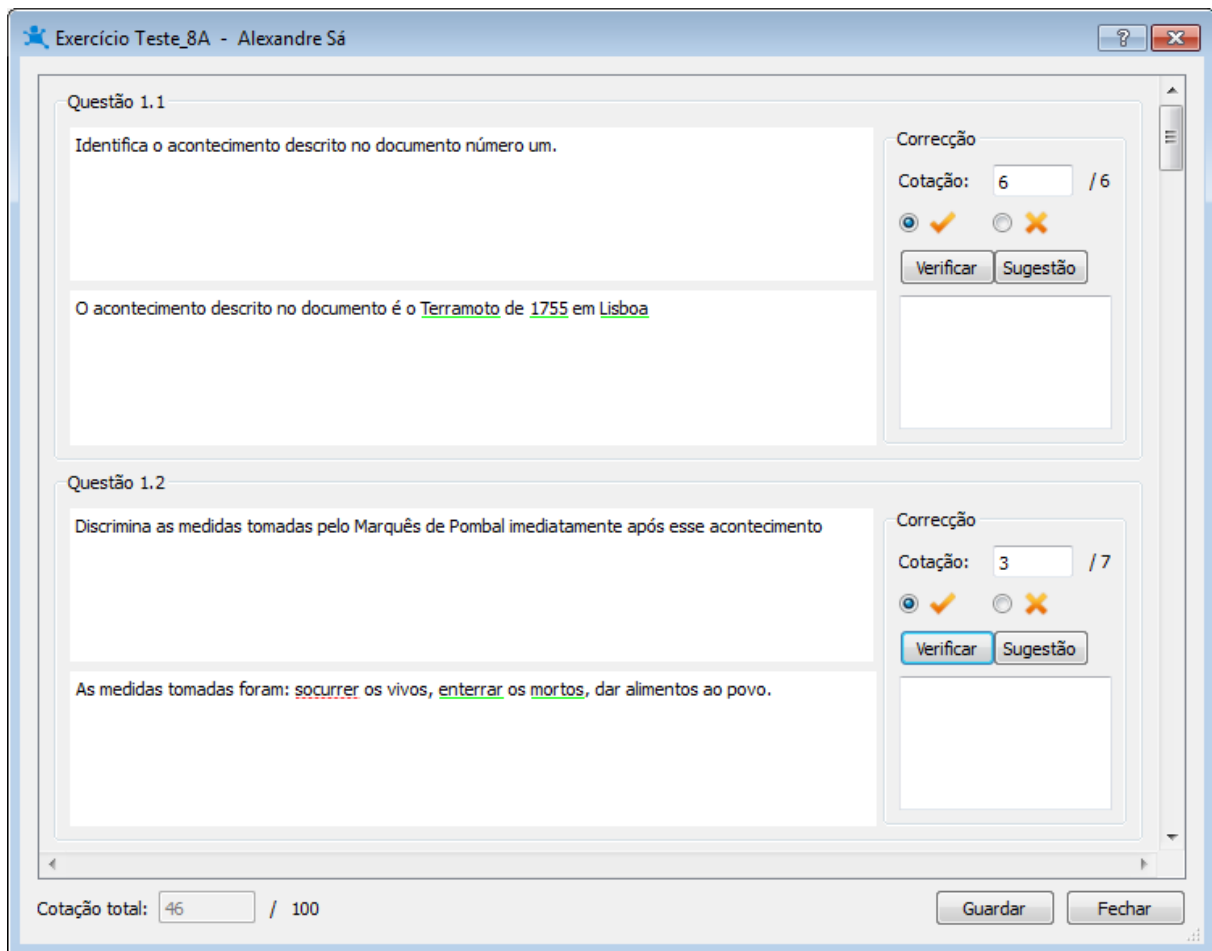


Figura 20 é possível visualizar o exercício de um aluno. Para cada questão o professor pode verificar a questão, cujo objectivo é assinalar a verde os termos correctos e a vermelho os erros ortográficos; ler a sua sugestão de resposta; adicionar um comentário relativamente à forma como o aluno respondeu e alterar a cotação sugerida pelo sistema.

Exercício Teste_8A - Alexandre Sá

Questão 1.1

Identifica o acontecimento descrito no documento número um.

O acontecimento descrito no documento é o Terramoto de 1755 em Lisboa

Correcção

Cotação: 6 / 6

✓ ✗

Verificar Sugestão

Questão 1.2

Discrimina as medidas tomadas pelo Marquês de Pombal imediatamente após esse acontecimento

As medidas tomadas foram: socorrer os vivos, enterrar os mortos, dar alimentos ao povo.

Correcção

Cotação: 3 / 7

✓ ✗

Verificar Sugestão

Cotação total: 46 / 100

Guardar Fechar

Figura 20 – Formulário de visualização do exercício de um aluno

O professor também poderá analisar os resultados, de uma forma externa à aplicação, tendo apenas que exportar os exercícios. O ficheiro que é disponibilizado tem o formato CSV para que possa ser importado para várias ferramentas.

Após a verificação dos resultados, o sistema disponibiliza o envio dos resultados para os alunos para que estes possam analisar a correcção do professor, incluindo os comentários registados pelo mesmo.

4.6. Caso de estudo

Nesta secção será apresentado e analisado um caso de estudo que tem como intuito demonstrar o funcionamento do módulo de correcção automática e de que forma este pode ser útil para auxiliar o professor na correcção de exercícios. O caso de estudo tem como

base uma turma de dezassete alunos, do oitavo ano de escolaridade que foram submetidos a um teste da disciplina de História (o enunciado do exercício pode ser consultado no Anexo I).

Todas as respostas dos alunos foram avaliadas pelo professor para determinar o nível de acordo entre a avaliação do professor e do sistema desenvolvido. O professor foi convidado a dar pontuação às respostas usando os pontos da resposta, isto é, se a resposta tem uma pontuação máxima de 7 pontos, o professor pode atribuir à questão um valor entre 0 e 7.

4.6.1. Definição do Caso de Estudo

Na análise efectuada pretende-se interpretar os resultados de forma a retirar conclusões relativamente à diferença de cotações entre o sistema e o professor. O objectivo é analisar a variância de cotações para cada uma das questões relativamente a todos os Alunos que participaram no exercício.

A cotação máxima por exercício é de cem pontos, sendo considerada uma avaliação positiva aquela que seja igual ou superior a cinquenta pontos. A cotação total de cada exercício é a medida quantitativa que permite ao professor definir posteriormente a medida qualitativa a atribuir.

Neste caso de estudo o professor baseou-se na Tabela 3.

Tabela 3 - Correspondência das medidas quantitativa e qualitativa

Medida Quantitativa	Medida Qualitativa
0 - 24	Fraco
25 - 49	Não Satisfaz
50 - 74	Satisfaz
75 - 89	Bom
90 - 100	Muito Bom

Os resultados obtidos pelo sistema estão arredondados a duas casas decimais, sendo os resultados do professor na generalidade números inteiros. Por essa razão foi necessário colocar os resultados do sistema em números inteiros pois de outra forma os valores atribuídos nunca coincidem, excepto nas cotações máxima e mínima.

Para analisar os dados foram utilizadas as medidas estatísticas seguintes: a média, a moda e mediana.

A média é obtida a partir da seguinte fórmula:

$$Média = \frac{1}{n} \sum_{i=1}^n ai$$

Esta fórmula permite calcular a média de cotações de cada uma das questões bem como a média de cada um dos exercícios.

A moda é o valor mais frequente num determinado conjunto de dados. Através deste valor é possível analisar as cotações atribuídas mais frequentemente em cada uma das questões.

Para determinar a mediana é necessário que o grupo de dados esteja ordenado. Se o número de elementos for par, a mediana é o elemento central; caso seja impar é necessário fazer a média dos dois números centrais.

4.6.2. Análise do Caso de Estudo

De forma a analisar o desempenho do sistema, pretende-se comparar os resultados obtidos relativamente à avaliação do professor e assim obter conclusões relativamente à sua eficácia.

Todas as respostas dos alunos foram avaliadas pelo professor e pelo sistema. A Tabela 4 resume os resultados obtidos nas experiências.

Tabela 4 – Resultados gerais por questão

Questão	Tamanho Resposta Prof. (palavra)	Tamanho Resposta Aluno (palavra)	Cotação Máxima	Média Avaliação Professor	Média Avaliação Sistema	Diferença Avaliações	Acordo	Média Acordo Professor-Sistema
1.1	3	3,81	6	5,35	5,53	-0,18	0,03	0,59
1.2	7	11,29	7	4,88	3,88	1,00	0,14	0,35
1.3	11	10,62	8	2,18	1,47	0,71	0,09	0,41
1.4	21	16,62	7	3,82	2,94	0,88	0,13	0,06
2.1	13	13,48	7	6	5,94	0,06	0,01	0,06
4.1	1	1	7	2,24	3,18	-0,94	0,13	0,24
4.2	5	1,52	6	4,59	4,59	0,00	0,00	0,88
5.1	4	3,48	7	4,47	3,71	0,76	0,11	0,12
5.2	13	12,86	7	3,53	4,47	-0,94	0,13	0,35
5.3	9	1,38	5	0,82	2,12	-1,30	0,26	0,41
6.1	30	7,76	6	1,41	0,53	0,88	0,15	0,71
7.1	22	19,86	6	3,94	2,24	1,70	0,28	0,29
7.2	13	6,34	7	5,47	4,29	1,18	0,17	0,12
7.3	9	5,1	7	3,47	2,88	0,59	0,08	0,12

Tal como é possível verificar através da Tabela 4, a diferença máxima entre a avaliação do sistema e do professor é de 1,70 pontos na questão 7.1 cujo a cotação máxima é de 6 pontos, com um acordo de 0,28.

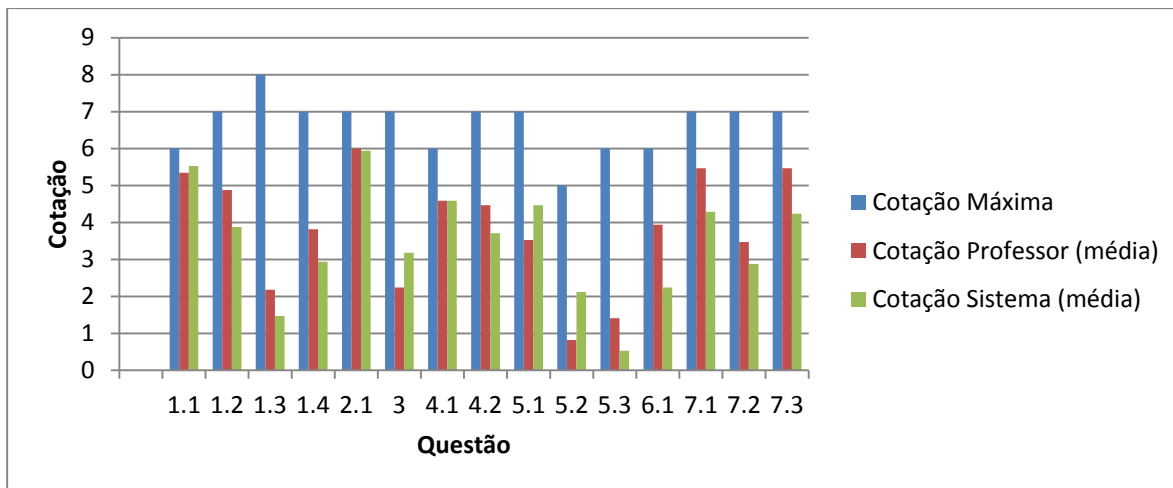


Figura 21 – Comparação das cotações atribuídas por questão

Tendo em consideração a Figura 21 é importante salientar que na generalidade as cotações sugeridas pelo professor são superiores em relação às cotações atribuídas pelo sistema, o que significa que o sistema na maioria dos casos é pessimista.

De todas as questões existem dois casos em que existe uma divergência mais significativa entre o resultado do sistema e do professor. Nas questões 5.2 e 6.1, apesar de divergirem uma positivamente e outra negativamente é onde é notada uma maior diferença. Em ambos os casos, existe uma diferença superior a um ponto. Visto que a resposta à questão 6.1 (“Qual foi a importância da Enciclopédia?”) poderia ser bastante abrangente, ou seja, existem várias possibilidades de resposta, o professor considerou mais conceitos do que aqueles que foram referidos na sua resposta de referência.

Foram consideradas várias medidas para avaliar o desempenho do sistema. A média de acordo entre o professor e o sistema para uma questão é o número de vezes que o professor e sistema concordam a dividir pelo número total de respostas dos alunos. Esta medida tende a diminuir à medida que a complexidade/comprimentos da resposta de referência aumenta. Nos casos em que não há acordo entre o professor e o sistema de pontuação, o professor normalmente atribui uma pontuação mais elevada do que a do sistema. Isso acontece porque a resposta do aluno é muito abstracta, ou é muito menor do que a resposta de referência. Nestes casos, a pontuação do sistema, cuja precisão é baseada na correspondência de palavras, é reduzida. Além disso, algumas respostas dos alunos estão num formato sem correspondência com a resposta de referência, mas a avaliação do professor indica que os alunos compreendem os conceitos de aprendizagem.

Por isso, o especialista avalia essas respostas com uma pontuação mais elevada, aumentando assim as diferenças entre as avaliações.

Para avaliar os resultados obtidos neste estudo, foi utilizada a correlação de *Pearson* que mede o grau de correlação, ou seja, o quanto a pontuação dos professores (X) está associada com a pontuação do sistema (Y). Obteve-se um resultado de correlação promissor de 0,78 o que indica que a correlação entre a avaliação do sistema e avaliação dos professores pode ser melhorada.

Tabela 5 – Análise de cotações coincidentes

Questão	1.1	1.2	1.3	1.4	2.1	3	4.1	4.2	5.1	5.2	5.3	6.1	7.1	7.2	7.3
Cotação	6	7	8	7	7	7	6	7	7	5	6	6	7	7	7
Alexandre	1	1	1	0	0	1	1	0	0	0	1	0	0	1	0
André	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0
Andreia	0	0	1	0	1	0	1	0	0	1	1	0	0	1	0
Catarina	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0
Cristiana	0	1	1	1	0	0	1	0	1	1	1	1	0	0	1
Edgar	1	0	0	0	1	0	1	0	1	1	1	0	1	1	0
Fábio	1	0	1	1	0	0	1	1	0	0	1	0	0	1	1
Filipe	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0
Hélio	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0
Maria	0	0	1	0	0	1	1	0	0	1	1	0	0	0	0
Marlene	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1
Matilde	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0
Melvin	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
Noémia	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1
Ofélia	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0
Paulo	1	1	0	0	1	0	1	0	0	0	0	0	0	0	1
Pedro	1	0	0	1	1	0	1	1	0	0	1	0	0	1	0
Média	0,65	0,35	0,35	0,29	0,71	0,24	0,88	0,29	0,29	0,35	0,59	0,06	0,29	0,41	0,29
Acordo															

A Tabela 5 regista para cada aluno e questão informação se a cotação atribuída pelo sistema coincide com a do professor. Caso isso aconteça é atribuído o valor um, caso contrário é atribuído o valor zero. Através da análise dos resultados é possível concluir que na resposta 4.1 há uma concordância entre o sistema e o professor em 88% das respostas

sendo este um valor bastante positivo. Em contrapartida na resposta 6.1 o sistema só atribuiu a mesma cotação em 6% dos casos. Esta divergência pode ser justificada com o facto de o professor poder ter interpretado as respostas de forma diferente da resposta de referência enquanto o sistema fez a verificação de todas as palavras da resposta indicada pelo professor.

Tabela 6 – Análise de Falsos Positivos e Falsos Negativos

Questão	1.1	1.2	1.3	1.4	2.1	3	4.1	4.2	5.1	5.2	5.3	6.1	7.1	7.2	7.3
Cotação	6	7	8	7	7	7	6	7	7	5	6	6	7	7	7
Alexandre	0	0	0	-1	1	0	0	-1	-1	1	0	-1	1	0	-1
André	0	0	-1	0	0	0	0	-1	0	0	0	-1	0	0	-1
Andreia	-1	-1	0	-1	0	1	0	-1	-1	0	0	-1	-1	0	-1
Catarina	0	-1	-1	-1	0	1	-1	1	0	1	0	-1	0	-1	-1
Cristiana	1	0	0	0	1	1	0	-1	0	0	0	0	-1	1	0
Edgar	0	-1	1	-1	0	1	0	1	0	0	0	-1	0	0	-1
Fábio	0	-1	0	0	1	1	0	0	1	1	0	1	-1	0	0
Filipe	1	-1	0	-1	0	0	0	-1	1	1	1	1	-1	1	1
Helio	0	-1	1	-1	0	1	0	0	-1	1	-1	-1	1	1	-1
Maria da Paz	-1	-1	0	1	-1	0	0	-1	1	0	0	-1	-1	-1	-1
Marlene	0	0	1	-1	0	1	0	0	0	1	-1	-1	0	-1	0
Matilde	0	0	1	-1	0	1	0	-1	-1	0	-1	-1	-1	-1	-1
Melvin	1	1	-1	1	-1	1	1	1	1	1	0	1	-1	0	-1
Noémia	0	-1	-1	0	0	1	0	-1	1	1	-1	-1	-1	-1	0
Ofélia	1	-1	1	-1	0	1	0	0	1	1	-1	-1	0	-1	-1
Paulo	0	0	-1	1	0	1	0	1	1	-1	-1	-1	-1	-1	0
Pedro	0	-1	1	0	0	1	0	0	1	1	0	-1	-1	0	1
FN (-1)	2	10	5	9	2	0	1	8	4	1	6	13	10	7	10
(0)	11	6	6	5	12	4	15	5	5	6	10	1	5	7	5
TP (1)	4	1	6	3	3	13	1	4	8	10	1	3	2	3	2

Tal como ilustra a Tabela 6, foi efectuada outra análise dos erros do sistema que divide os erros em duas categorias: falsos negativos (FN) e falsos positivos (FP). O FN ocorre quando o sistema dá uma pontuação inferior do que aquela que devia ser atribuída. O FP ocorre quando o sistema atribui mais pontos do que a resposta vale. No caso da avaliação do

sistema, o número de FN foi muito maior que o número de FP, 0,35 de todos os erros foram FN enquanto apenas 0,25 foram FP. A proporção relativamente a FN é devido ao facto de que é muito difícil antecipar todas as paráfrases possíveis para uma resposta. Se alguma possibilidade correcta é perdida, em seguida, a resposta do aluno vai ser FN. A razão para FP do sistema ocorre quando um aluno não sabe a resposta a uma pergunta, mas escreve algo que é claramente errado.

De forma a analisar o desempenho do sistema foi efectuada uma comparação da avaliação qualitativa dada pelo sistema e pelo professor. Assim é possível analisar os efeitos práticos da correcção sugerida pelo sistema nas classificações dos alunos. Normalmente, para determinar a nota de um aluno no fim de um determinado período o professor analisa as avaliações qualitativas atribuídas ao longo dos exercícios.

Tabela 7 – Comparação das avaliações do sistema/professora/professor

Alunos	Avaliação Quantitativa		Avaliação Qualitativa	
	Sistema	Professor	Sistema	Professor
Alexandre	48	47	Não Satisfaz	Não Satisfaz
André	59	67	Satisfaz	Satisfaz
Andreia	37	54	Não Satisfaz	Satisfaz
Catarina	57	72	Satisfaz	Satisfaz
Cristiana	55	53	Satisfaz	Satisfaz
Edgar	44	50	Não Satisfaz	Satisfaz
Filipe	34	31	Não Satisfaz	Não Satisfaz
Fábio	45	38	Não Satisfaz	Não Satisfaz
Hélio	58	64	Satisfaz	Satisfaz
Maria da Paz	38	48	Não Satisfaz	Não Satisfaz
Marlene	62	65	Satisfaz	Satisfaz
Matilde	61	84	Satisfaz	Bom
Melvin	48	38	Não Satisfaz	Não Satisfaz
Noémia	54	77	Satisfaz	Bom
Ofélia	71	73	Satisfaz	Satisfaz
Paulo	63	68	Satisfaz	Satisfaz
Pedro	55	51	Satisfaz	Satisfaz

Tal como é possível verificar através da Tabela 7, há quatro casos em que o sistema sugere uma avaliação qualitativa inferior à atribuída pelo professor, o que significa que em 76,5% dos casos foram atribuídas avaliações iguais. No caso das avaliações diferentes, em todas elas o sistema foi penalizador em relação ao professor. Por outro lado, nas restantes avaliações o sistema obteve cotações com um grau de similaridade elevado.

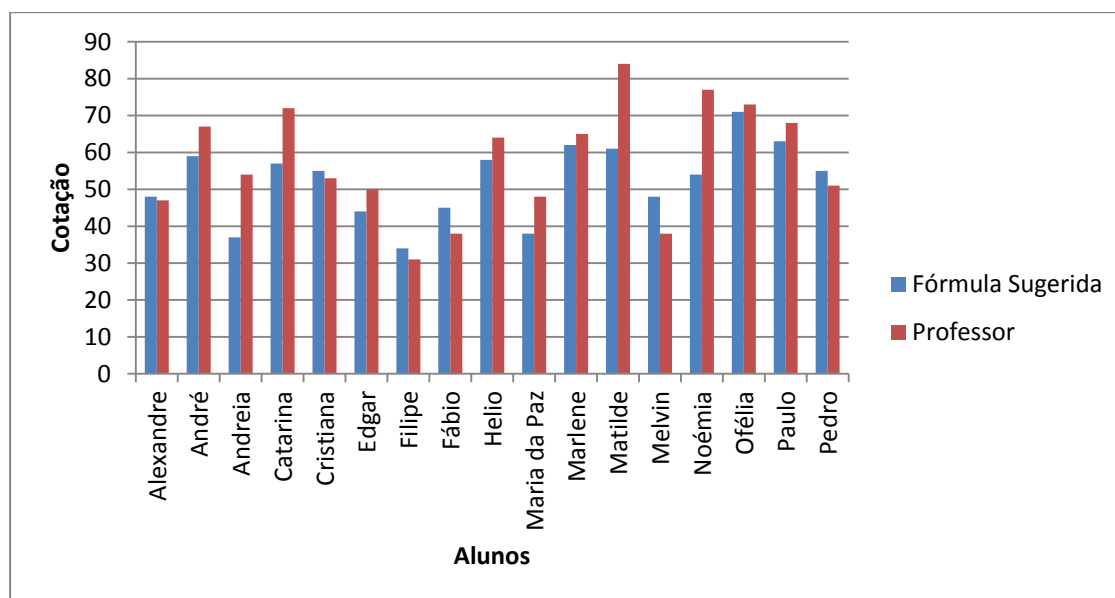


Figura 22 - Comparação das cotações atribuídas por aluno

Através da Figura 22, é possível concluir que na maioria dos casos, apesar das avaliações qualitativas serem iguais, ao nível das avaliações quantitativas o sistema sugere cotações inferiores. Isso deve-se ao facto do professor aceitar respostas que apesar de não terem semelhança com a resposta de referência, têm conceitos que são considerados correctos por este. Quando acontece o contrário significa que o sistema avalia todas as respostas de uma forma imparcial, e que por isso é coerente em todas as cotações atribuídas. O sistema não é influenciado por avaliações anteriores e por isso consegue um desempenho semelhante ao longo da correcção.

4.7. Conclusão

No início deste capítulo foi apresentado o sistema desenvolvido, desde as tecnologias utilizadas até à implementação prática.

O módulo desenvolvido teve como principal objectivo disponibilizar a correcção de exercícios de uma forma mais rápida do que quando é efectuada manualmente pelo professor. Assim sendo, o professor despende menos tempo a executar esta tarefa apesar de poder rever e se necessário alterar as cotações sugeridas pelo sistema.

Em seguida foram apresentadas de uma forma detalhada todas as tarefas necessárias para a implementação do módulo.

Foi apresentado um caso de estudo que teve como objectivo permitir a análise dos resultados obtidos pelo sistema comparativamente com os resultados disponibilizados pelo professor.

No que diz respeito aos resultados, podem ser considerados satisfatórios. Na generalidade dos casos, o sistema atribuiu notas qualitativas semelhantes às do professor.

5. Conclusões

5.1. Resumo

O principal objectivo do trabalho desenvolvido foi a criação de um módulo de correcção automática de respostas de texto livre para a aplicação UNI_NET-Classroom. O intuito do trabalho proposto é tornar possível a redução do tempo despendido pelo professor na correcção de exercícios.

A área de correcção automática de texto livre tem despertado interesse desde há algumas décadas. O facto do ser humano despende um tempo bastante superior ao de um computador no processamento de informação levou a que investigadores tivessem interesse em colocar um computador a fazer tarefas como análise e classificação de textos. Ao longo dos tempos foram desenvolvidas várias técnicas de forma a melhorar o desempenho e consequentemente os resultados. Estes sistemas têm diversas vantagens e em oposição ao ser humano não são influenciados por factores externos, como prazos ou fadiga. Assim, é possível obter uma maior coerência e objectividade relativamente à qualidade dos testes.

Neste sentido foi realizado um levantamento do estado da arte dos temas envolvidos neste trabalho, nomeadamente: correctores ortográficos que pudessem ser posteriormente integrados na aplicação, técnicas usadas em sistemas de correcção automática e os sistemas de correcção automática mais referenciados na literatura.

Foram analisados vários correctores ortográficos de forma a seleccionar um que se adequasse às necessidades do módulo que iria ser desenvolvido. Depois da comparação das diversas características de cada uma das ferramentas estudadas, o corrector ortográfico seleccionado foi o JSpell. Esta ferramenta adequa-se às necessidades do módulo de correcção automática por diversas razões. A licença é de utilização livre e o código fonte está disponível. Para além disso, contém todos os dicionários que a aplicação necessita permitindo assim a utilização desta funcionalidade em diversas disciplinas do sistema de ensino.

De forma a aumentar o conhecimento sobre a área de correcção automática foi efectuada uma análise de algumas das técnicas que frequentemente são aplicadas e de algumas ferramentas mais relevantes. É importante referir que estas aplicações têm utilidade prática em diversas áreas de aprendizagem, e algumas delas disponibilizam resultados para provar a qualidade dos seus resultados. Na maioria dos casos estudados, foi possível depreender

que a maioria destas ferramentas necessita de uma grande quantidade de informação para ter resultados satisfatórios, o que dificultou o enquadramento com o objectivo deste trabalho. Dado este requisito, em alguns casos também não é possível obter resultados em tempo real que seria outro foco do módulo a desenvolver. Nesta pesquisa o objectivo foi adquirir informação sobre a área tendo sempre como referência que neste trabalho o que se pretendia era um sistema que fizesse a correcção automática de texto livre baseando-se na resposta de referência do professor e a disponibilização de resultados da forma mais rápida quanto possível.

Tal como foi referido anteriormente, o módulo foi desenvolvido para ser integrado na ferramenta de gestão de sala de aula UNI_NET-Classroom. Esta ferramenta tem como principal objectivo facilitar a gestão de informação e a gestão da sala de aula pelo professor disponibilizando diversas funcionalidades de monitorização dos alunos durante uma aula interactiva. Visto que um dos componentes desta aplicação é a gestão de exercícios, pretendeu-se então acrescentar a possibilidade de além do professor poder criar e enviar exercícios para os alunos, poder ter também ao seu dispor uma sugestão de correcção dos mesmos. Para tal, o professor tem de adicionar uma resposta referência do exercício, que anteriormente apenas era obrigatória para as respostas de escolha múltipla, de forma ao sistema poder proceder à correcção.

Depois da aquisição do conhecimento considerado necessário no âmbito deste trabalho, iniciou-se a implementação. A partir deste momento, foi necessário iniciar o desenvolvimento das tarefas de pré-processamento para colocar as respostas do professor e alunos numa forma canónica passível de serem comparadas. Estas etapas foram desenvolvidas de forma independente permitindo assim analisar e reorganizar, sempre que necessário, a sua ordem de execução. No final, definiu-se que as tarefas deviam ter a ordem que será descrita de seguida. Em primeiro lugar são removidos os caracteres especiais, mais especificamente a pontuação. Não podem ser removidos caracteres que constituem a palavra pois assim a palavra passaria a conter erros. Na tarefa seguinte, de correcção de erros ortográficos, é utilizado o JSpell que além de detectar os erros sugere uma correcção. De seguida são removidas as *stop-words*, pois não têm valor para a atribuição da cotação. Depois da resposta apenas conter as palavras relevantes para a avaliação, procede-se à categorização das palavras que as coloca na sua forma canónica e adiciona-lhes a sua categoria. A última tarefa, e com maior relevância, é a comparação das respostas. É neste momento que as respostas são avaliadas e lhes é atribuída uma cotação. Depois do processo de correcção automática estar concluído os resultados são disponibilizados ao professor para que este os possa analisar.

De modo a apresentar uma avaliação rigorosa do funcionamento do módulo desenvolvido, foram analisadas as respostas de uma turma a um teste de História e apresentados os resultados obtidos. O objectivo foi avaliar o desempenho do sistema em relação à avaliação do professor.

5.2. Objectivos Alcançados

Na fase inicial deste trabalho foi realizado um estudo relativamente às ferramentas de correcção ortográfica com o intuito de seleccionar uma para integrar na aplicação UNI_NET-Classroom. Relativamente às ferramentas analisadas, foi dada principal atenção a características como o tipo de licença, a disponibilidade do código fonte e os dicionários disponíveis.

Também foi realizada uma análise do estado da arte da Correcção Automática de respostas de texto livre. Inicialmente foi definido o conceito de Correcção Automática, seguido pela análise das técnicas mais frequentemente utilizadas, assim como dos principais sistemas existentes.

Posteriormente foi efectuada uma breve descrição da aplicação UNI_NET-Classroom pois o módulo desenvolvido foi integrado nesta ferramenta. A descrição das funcionalidades mais relevantes foi importante para fazer um enquadramento do trabalho desenvolvido.

O principal objectivo deste trabalho, que se prendeu com o desenvolvimento de um módulo de correcção automática, foi também alcançado. O foco deste trabalho foi disponibilizar uma ferramenta de apoio ao professor, que não pretende substituí-lo, mas sim facilitar-lhe o processo de correcção de exercícios.

Apesar de não ser um objectivo inicial deste trabalho, foi necessário corrigir alguns erros da ferramenta JSpell, o que permitiu contribuir para o melhoramento da mesma.

Por último, foi analisado um caso de estudo para avaliar a solução implementada.

5.3. Limitações e Trabalho Futuro

Apesar de já há algumas décadas existir investigação na área de correcção automática de texto livre é sempre possível progredir, logo este trabalho não é excepção. Apesar da dificuldade, é possível obter resultados semelhantes ao avaliador humano através do

processamento por computador. Como foi possível verificar ao longo do estudo sobre esta área, é necessário existir, na maioria dos casos, referências a correcções efectuadas por humanos. Caso isso não aconteça tem de ser considerada uma elevada quantidade de informação, que torna o processamento mais lento pois a área de pesquisa é substancialmente maior.

Um dos pontos que poderiam ser melhorados seria a elaboração da correcção do mesmo teste por diferentes professores e usar as respostas como referência, em vez de as gerar automaticamente através de sinónimos por categoria. O facto de na análise dos resultados apenas se ter considerado a correcção de um professor, condiciona a análise de desempenho da aplicação. Como foi possível verificar na análise dos casos de estudo, o professor pode considerar que uma resposta está correcta sem que esta contenha qualquer termo da resposta de referência. Para o sistema, que se baseia nos conteúdos inseridos pelo professor, não é possível prever e corresponder correctamente a estas situações.

Em relação à implementação, uma das tarefas de pré-processamento que poderia ser melhorada seria a categorização das palavras. Neste momento o sistema assume a primeira sugestão indicada pelo corrector ortográfico JSpell o que não quer dizer que em todos os casos seja a escolha mais acertada. Mais processamento terá de ser investido nesta fase crucial para a obtenção de bons resultados.

Por fim, seria interessante dar ao professor a possibilidade de seleccionar uma das várias metodologias de atribuição de cotação estudadas e que não foram consideradas na versão final do módulo desenvolvido. Desta forma, este poderia seleccionar que critérios seriam mais relevantes para a avaliação já que neste momento a única interacção com o sistema é na definição do idioma que deve ser utilizado e no valor da penalização dos erros ortográficos.

5.4. Considerações Finais

Para finalizar, gostaria de referir que todo o trabalho desenvolvido envolveu um enorme empenho e dedicação devido à área de investigação em que este se enquadra. A aquisição de conhecimento foi bastante e facilitou muito o desenvolvimento da aplicação.

O trabalho realizado terminou com bons resultados e com a disponibilização de uma ferramenta que apesar de poder ser melhorada, já reduz consideravelmente o trabalho dos professores.

No futuro, pretende-se melhorar ao nível das limitações referidas e evoluir de modo a disponibilizar uma ferramenta aos professores que garanta a confiança dos mesmos na sua utilização.

Referências

- [Burstein et al., 1996] Burstein, J., Kaplan, R., Wolff, S. and Chi Lu, Using Lexical Semantic Techniques to Classify Free-Responses, in Proceedings of SIGLEX 1996 Workshop, Annual Meeting of the Association of Computational Linguistics, University of California, Santa Cruz, 1996
- [Burstein et al., 2001] Burstein J., Leacock C., Swartz, R., Automated evaluation of essay and short answers, M. Danson (Ed.), Proceedings of the Sixth International Computer Assisted Assessment Conference, Loughborough University, UK, 2001
- [Chang & Hsiao, 2008] Te-Min Chang, Wen-Feng Hsiao, A hybrid approach to automatic text summarization, Computer and Information Technology, 2008
- [Cohen, 1996] Jacques Cohen, Communications of the ACM, Volume 39 Issue 1, ACM New York, pp. 80-91, New York, 1996
- [Dikli, 2006] Semire Dikli, Automated Essay Scoring, Turkish Online Journal of Distance Education-TOJDE, Volume: 7, Number: 1, Article: 5, 2006
- [Figueira et al., 2009] Helena Figueira, Afonso Mendes, Pedro Mendes, Cláudia Pinto, “Implicações do acordo ortográfico da língua portuguesa de 1990 em programas informáticos de correção ortográfica”, 2009
- [Foltz et al., 1999] Foltz, D. Laham, and T. Landauer, The Intelligent Essay Assessor: Applications to Educational Technology. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1(2), 1999
- [Grishman, 1997] Grishman, Ralph, Information Extraction A Multidisciplinary Approach to an Emerging Information Technology, Springer Berlin / Heidelberg, pp. 10-27, 1997
- [Jerrams-Smith et al., 2001] Jerrams-Smith J., Soh V., Callear D., Bridging gaps in computerized assessment of texts. In: Proceedings of the International Conference on Advanced Learning Technologies, pp.139-140, 2001
- [Landauer & Dumais, 1997] Landauer, T. K., Dumais, S. T., A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge, Psychological Review, 104, pp. 211-140, 1997

[Landauer et al., 1998] T.K. Landauer, P.W. Foltz, and D. Laham, An introduction to Latent Semantic Analysis, *Discourse Processes*, 25(2), pp. 259-284, 1998

[Li & Hu, 2009] Yaxiong Li, Dan Hu, Study on the Classification of Mixed Text Based on Conceptual Vector Space Model and Bayes, *International Conference on Asian Language Processing*, 2009

[Liddy, 2001] Liddy, E.D., Natural Language Processing, In *Encyclopedia of Library and Information Science*, 2nd Ed. NY. Marcel Decker, Inc., 2001

[Lutticke, 2005] R. Lutticke, Graphic and NLP Based Assessment of Knowledge about Semantic Networks, In *Proceedings of the Artificial Intelligence in Education (AIED) conference*, 2005.

[Marin et al., 2004] Perez Marin D., Pascual-Nieto I., Rodriguez P., Computer-assisted assessment of free-text answers. *The Knowledge Engineering Review*, 24(4), pp. 353-374, 2009

[Marin, 2007] Perez Marin D., Adaptive Computer Assisted Assessment of free-text students' answers: an approach to automatically generate students' conceptual models pp. 37-60, 2007

[Mason & Grove-Stephenson, 2002] Mason O., Grove-Stephenson I., Automated free text marking with paperless school. In: M. Danson (Ed.), *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough University, UK, 2002

[Mitchell et al., 2002] Mitchell T., Russel T., Broomhead P., Aldridge N., Towards robust computerized marking of free-text responses. In: Danson (Ed.), *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughboroug University, UK, 2002

[Oliveira, 2009] Fábio Abreu Dias de Oliveira, "Processamento de linguagem natural: princípios básicos e a implementação de um *analisador sintático* de sentenças da língua portuguesa", Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, 2009

[Page, 1994] Page, E.B, New computer grading of student prose, using modern concepts and software. *Journal of Experimental Education*, 62(2), pp. 127-142, 1994.

[Powers et al., 2001] Donald E. Powers, Jill C. Burstein, Martin Chodorow, Mary E. Fowles, Karen Kukich, Stumping E-Rater: Challenging the Validity of Automated Essay Scoring, ETS Research Report 01-03, 2001

[Rudner & Liang, 2002] L.M. Rudner, T. Liang, Automated Essay Scoring Using Bayes' Theorem. In Proceedings of the annual meeting of the National Council on Measurement in Education, 2002

[Simões & Almeida, 2001] Alberto Manuel Simões, José João Almeida, jspell.pm – um módulo de análise morfológica para uso em processamento de linguagem natural, Departamento de Informática, Universidade do Minho, Braga, 2001

[Thomas et al., 2004] Thomas P., Haley D., Roeck A., Petre M., E-Assessment using Latent Semantic Analysis in the Computer Science Domain: A Pilot Study. In: The Proceedings of the Workshop on eLearning for Computational Linguistics and Computational Linguistics for eLearning, pp. 38-44. Association for Computational Linguistics, 2004

[Valenti et al., 2003] Valenti S., Neri F., Cucchiarelli A., An Overview of Current Research on Automated Essay Grading. Journal of Information Technology Education, 2, 2003

[Visnick, 2003] Lorinda Visnick, Clustering Techniques, A Technical Whitepaper, Progress Software Corporation, USA, 2003

[Wang, 2009] Wang L.S, Relevance weighting of multi-term queries for Vector Space Model, Computational Intelligence and Data Mining, 2009

[Wanner, 2004] Leo Wanner, Introduction to Clustering Techniques, 2004

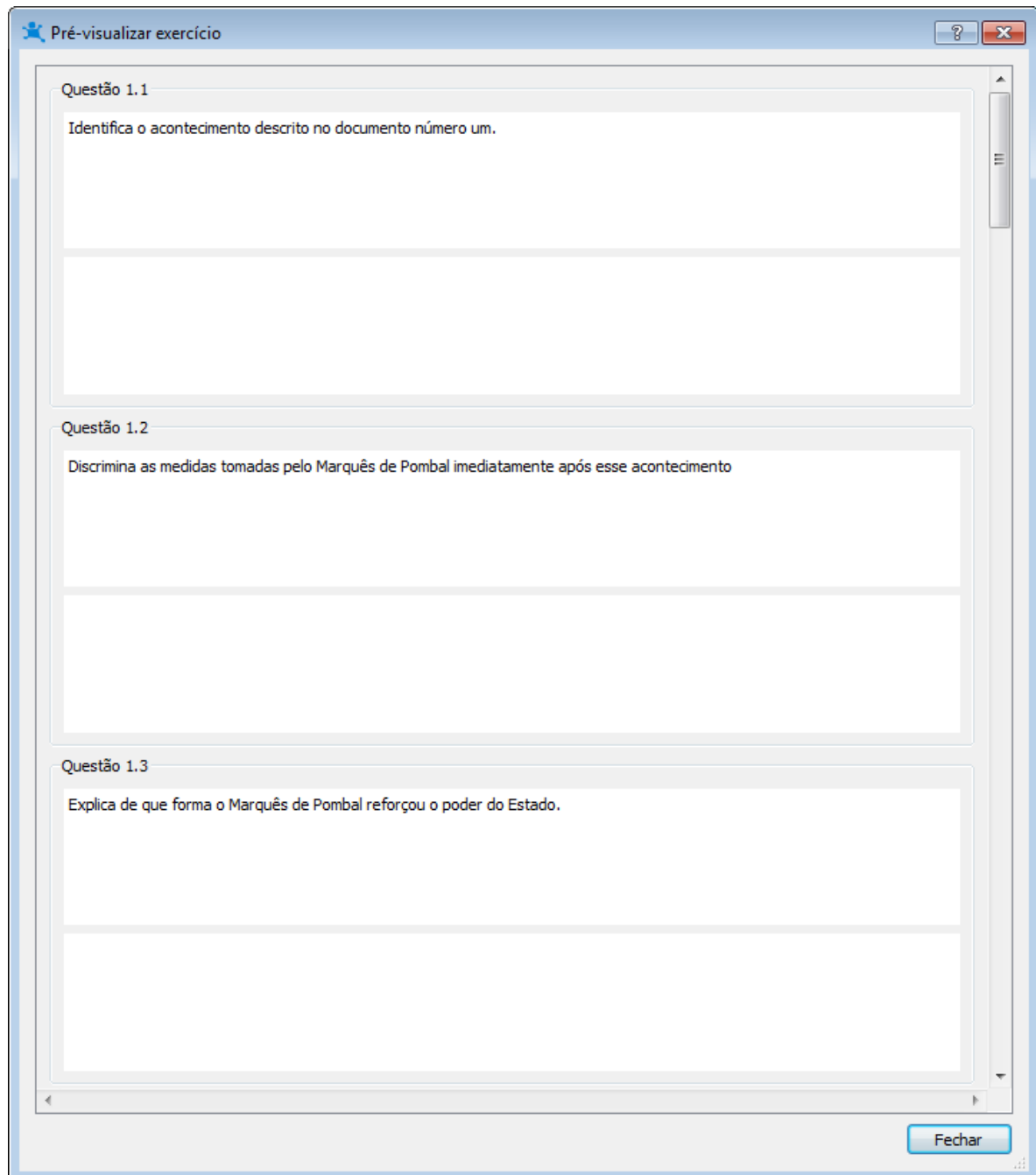
[Whittington & Hunt, 1999] Whittington, D., Hunt, H., Approaches to the computerized assessment of free text responses. In: M. Danson (Ed.), Proceedings of the Sixth International Computer Assisted Assessment Conference, Loughborough University, UK, 1999

Referências URL

- [Betsy, 2011] <http://echo.edres.org:8080/betsy/>, última visita em 2011
- [C++, 2011] <http://www.cplusplus.com/>, última visita em 2011
- [FLiP, 2010] <http://www.flip.pt/Produtos/tabid/203/Default.aspx>, última visita em 2010
- [IEA, 2011] <http://www.knowledge-technologies.com>, última visita em 2011
- [ieSpell, 2010] <http://www.iespell.com/>, última visita em 2010
- [GoogieSpell, 2010] <http://orangoo.com/labs/GoogieSpell/>, última visita em 2010
- [OpenThesaurus, 2010] <http://openthesaurus.caixamagica.pt/>, última visita em 2010
- [Orangoo, 2010] <http://orangoo.com/spellcheck/>, última visita em 2010
- [Qt Creator, 2011] <http://qt.nokia.com/products/developer-tools/>
- [Revisor, 2010] <http://revisor.com.ar/>, última visita em 2010
- [SpellJax, 2010] <http://www.techiequest.com/spelljax-free-online-ajax-spell-checker/>, última visita em 2010
- [*Stop-words*, 2010] <http://miningtext.blogspot.com/2008/11/listas-de-stopwords-stoplist-portugues.html>, última visita em 2010
- [UNI_NET-Classroom, 2011]
http://uninet.pt/index.php?page=shop.product_details&flypage=shop.flypage&product_id=26&category_id=5&manufacturer_id=0&option=com_virtuemart&Itemid=29, última visita em 2011
- [WebJspell, 2010] <http://natura.di.uminho.pt/webjspell/jsolhelp.pl>, última visita em 2010

Anexo I

As seguintes ilustrações contêm o exercício utilizado no caso de estudo da secção 4.6.



The image shows a screenshot of a software window titled "Pré-visualizar exercício". The window contains three questions, each with a text input area. The questions are:

- Questão 1.1**
Identifica o acontecimento descrito no documento número um.
- Questão 1.2**
Discrimina as medidas tomadas pelo Marquês de Pombal imediatamente após esse acontecimento
- Questão 1.3**
Explica de que forma o Marquês de Pombal reforçou o poder do Estado.

At the bottom right of the window, there is a button labeled "Fechar".

Pré-visualizar exercício

Questão 1.4

Analisa de que forma o Clero e a Nobreza foram prejudicados durante o reinado de D.José I.

Questão 2.1

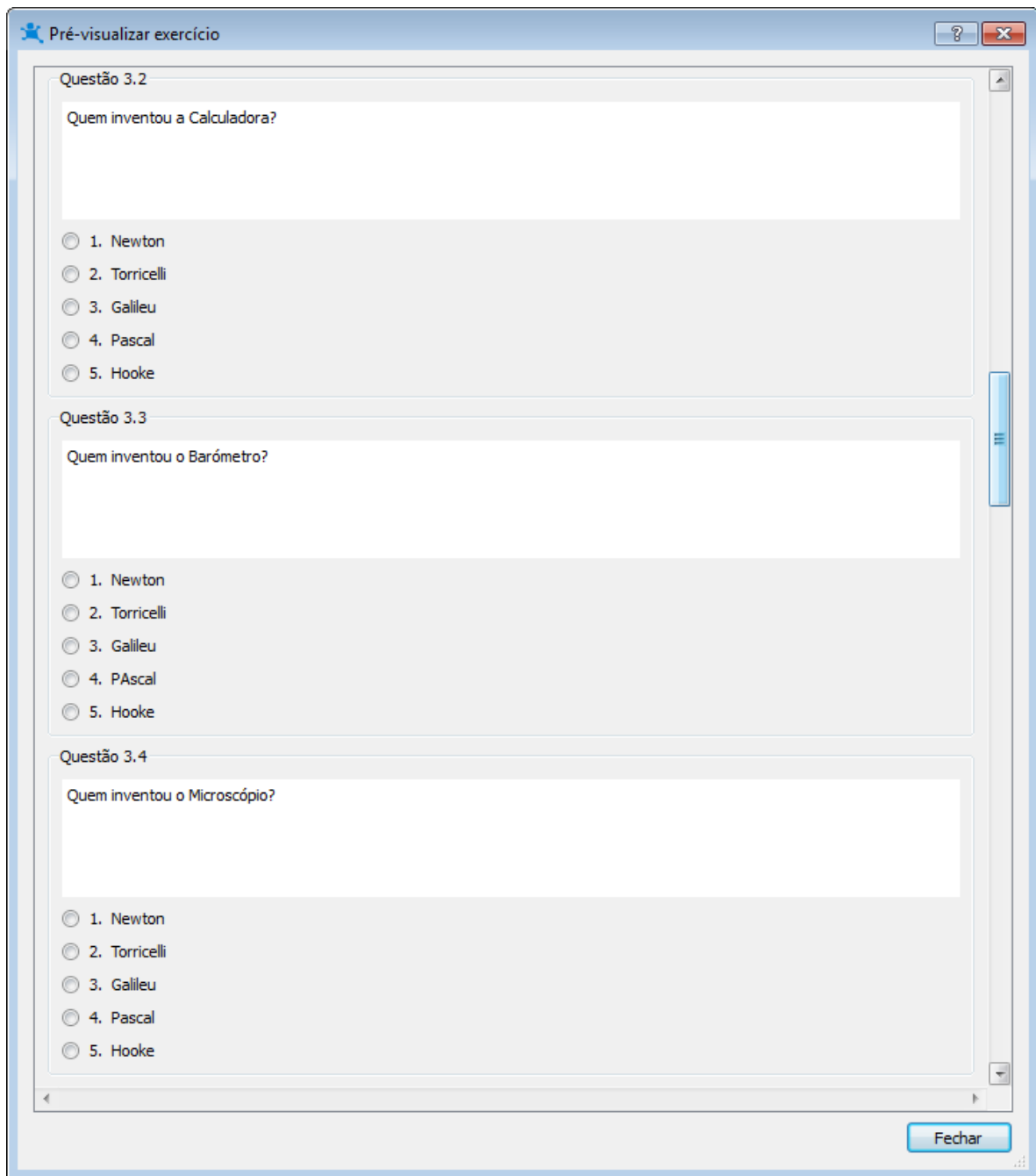
Caracteriza o Urbanismo Pombalino.

Questão 3.1

Quem inventou a Luneta astronómica?

- 1. Newton
- 2. Torricelli
- 3. Galileu
- 4. Pascal
- 5. Hooke

Fechar



Pré-visualizar exercício

Questão 3.5

Quem inventou o Telescópio?

1. Newton

2. Torricelli

3. Galileu

4. Pascal

5. Hooke

Questão 4.1

Identifica o método científico defendido no documento.

Questão 4.2

Justifica o atraso científico de Portugal e Espanha relativamente aos países do Norte da Europa.

Fechar

Pré-visualizar exercício

Questão 5.3

Explica o que entendes por Iluminismo.

Questão 5.4

Refere os deveres dos governantes "iluminados".

Questão 5.5.1

Ideias e doutrinas defendidas por Rousseu.

Fechar

Pré-visualizar exercício

Questão 5.5.2

Ideias e doutrinas defendidas por Voltaire.

Questão 5.5.3

Ideias e doutrinas defendidas por Montesquieu.

Questão 6.1

Qual foi a importância da Enciclopédia?

Fechar

Pré-visualizar exercício

Questão 7.1

Avalia a importância dos "estrangeirados" para a introdução do Iluminismo em Portugal.

Questão 7.2

Explica a importância de Luís António Verney.

Questão 7.3

Descreve as principais reformas do ensino empreendidas pelo Marquês de Pombal.

Fechar

