Universidade
Estadual de Londrina

LUCAS BUSATTA GALHARDI

AUTOMATIC GRADING OF PORTUGUESE SHORT ANSWERS USING A MACHINE LEARNING APPROACH

LONDRINA
2019

LUCAS BUSATTA GALHARDI

AUTOMATIC GRADING OF PORTUGUESE SHORT ANSWERS USING A MACHINE LEARNING APPROACH

Dissertation presented to the Master's Program in Computer Science at Londrina State University to obtain the degree of Master in Computer Science.

Advisor:	Prof. Dr. Jacques Duílio Brancher
Co-advisor:	Prof. Dr. Rodrigo Clemente Thom de Souza

LONDRINA
2019

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Galhardi, Lucas Busatta.
Automatic Grading of Portuguese Short Answers Using a Machine Learning Approach / Lucas Busatta Galhardi. - Londrina, 2019.
136 f.

Orientador: Jacques Duílio Brancher.
Coorientador: Rodrigo Clemente Thom de Souza.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2019.
Inclui bibliografia.

1. Automatic Grading - Tese. 2. Short Answer - Tese. 3. Machine Learning - Tese. 4. Natural Language Processing - Tese. I. Duílio Brancher, Jacques. II. Thom de Souza, Rodrigo Clemente. III. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

LUCAS BUSATTA GALHARDI

AUTOMATIC GRADING OF PORTUGUESE SHORT ANSWERS USING A MACHINE LEARNING APPROACH

Dissertation presented to the Master's Program in Computer Science at Londrina State University to obtain the degree of Master in Computer Science.

EXAMINATION BOARD

Advisor: Prof. Dr. Jacques Duílio Brancher
Londrina State University

Prof. Dr. Pedro Paulo da Silva Ayrosa
Londrina State University

Profa. Dra. Gislaine Camila Lapasini Leal
Maringá State University

Londrina, March 29, 2019.

Este trabalho é dedicado aos meus pais que,
ao me apoiarem das mais diversas formas,
tornaram a realização deste trabalho possível.

ACKNOWLEDGEMENTS

I am thankful to God for giving me everything that I needed to get this far and for keeping me able in order to finish this work.

To my parents, Luis Claudio Galhardi and Nilva Busatta, that spared no resources and time to give me all I needed in order to work on this project. Also to my brother, for helping me in his own way.

To all the teachers that I had throughout my life that inspired me in the search for knowledge and in the dream of becoming a teacher myself.

To my childhood friends, Nahan, João Antônio and Thiago for keeping this friendship for so long and for being there when I needed a friendly face.

To my friends from college for sharing with me the enthusiasm for computer science and so many moments of study and fun.

To the friends I made during the master program period. To Yago for keeping me company in the lab and helping me when I needed. To Luiz for his wise advices and productive conversations. To all the other friends for the happy moments shared together and their support.

To all my other friends that are part of my life somehow and supported me.

To my advisor, Jacques Duílio Brancher, for his life's wisdom and for suggesting such a gratifying dissertation thematic. Also to my co-advisor Rodrigo Clemente Thom de Souza, for his experienced advices.

To the Londrina State University, for providing such a beautiful and great study/work environment for the last six years. Also to the Computer Science Department and its staff for the support.

To the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)* - Finance Code 001 for the financial support.

To teacher Ana Cristina Rodrigues from Jaguarão for her support at elaborating this work's questions.

To teacher Silmara Sartoreto Oliveira for her availability on collaborating in this work. Also to all the 14 biology students that helped with their grading and make all this work possible.

To all the teachers, principals and educational supervisors for opening their schools for collaborating with this project. Specially to my childhood school and its teachers and principal. Also to all of their students who answered to the proposed questions.

"We do not need magic to change the world,
we carry all the power we need
inside ourselves already:
we have the power to imagine better."
(Very Good Lives, J.K. Rowling)

GALHARDI, L. B.. **Automatic Grading of Portuguese Short Answers Using a Machine Learning Approach**. 2019. 136p. Master's Thesis (Master in Computer Science) – Londrina State University, Londrina, 2019.

ABSTRACT

Assessments are routinely used in learning environments in order to estimate a percentage of the retained knowledge from students. Despite its importance, teachers usually find the task of assessing lots of discursive answers very time-consuming. Teachers work's conditions and their own human subjectivity have a great impact on grading, as humans make mistakes for some reasons like fatigue, bias or the simple ordering of student's tests. These problems become more intense in tools like Virtual Learning Environments and Massive Open Online Courses that have recently improved their popularity and are used by way more students than physical classes. Aiming at assisting in those difficulties, this dissertation explores the Automatic Short Answer Grading (ASAG) field using a machine learning approach, with three main goals: (1) to perform a systematic review on the subject in order to get an overview of the state of the art and future trends; (2) collect real-world Portuguese ASAG data; and (3) build, evaluate and compare different approaches when automatically grading short answers. For the first goal, we systematically reviewed 44 papers using different techniques when tackling ASAG, analyzing many of their aspects, from the data to model evaluation. For the second, 7473 short answers were collected from 659 students and 9558 grades were gathered for the answers from 14 human evaluators (some answers had more than one grade). For the last goal, six different approaches were experimented and a final model was created with their combination. The model's effectiveness showed to be satisfactory, with kappa scores indicating between moderate to substantial agreement between the model and human grading. Results showed that a machine learning approach can be efficiently used on short answers grading, even for the Portuguese language.

Keywords: Automatic grading. Short answers. Machine learning. Natural language processing.

GALHARDI, L. B.. **Avaliação Automática de Questões Discursivas em Português Usando uma Abordagem de Aprendizado de Máquina**. 2019. 136f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2019.

RESUMO

Avaliações são rotineiramente utilizadas em contextos de aprendizado a fim de estimar o conhecimento retido pelos estudantes. Apesar de sua importância, professores geralmente consideram a tarefa de avaliar respostas discursivas como muito trabalhosa. As condições de trabalho do professor e a sua própria subjetividade podem influenciar nas suas avaliações, pois humanos estão sujeitos ao cansaço, à outras influências e a nota de um aluno pode depender até mesmo da ordem de correção. Esses problemas se apresentam de forma ainda mais intensa em ferramentas como Ambientes Virtuais de Aprendizagem e Cursos Onlines Abertos e Massivos, que recentemente aumentaram sua popularidade e são usados por muito mais estudantes de uma vez que salas de aula físicas. Visando auxiliar nesses problemas, essa dissertação explora a área de pesquisa da avaliação automática de respostas discursivas usando uma abordagem de aprendizado de máquina, com três principais objetivos: (1) realizar uma revisão sistemática da literatura sobre o assunto a fim de se obter uma visão geral do estado da arte e de suas principais técnicas; (2) coletar dados reais de exercícios discursivos escritos na Língua Portuguesa por estudantes; e (3) implementar, avaliar e comparar diferentes abordagens para o sistema de avaliação automática das respostas. Para o primeiro objetivo, 44 artigos foram sistematicamente revisados, analisando vários de seus aspectos, desde os dados utilizados até a avaliação do modelo. Para o segundo, foram coletadas 7473 respostas de 659 estudantes, além de 9558 avaliações feitas por 14 avaliadores humanos (algumas respostas receberam mais de uma avaliação). Para o último objetivo, seis abordagens diferentes foram experimentadas e um modelo final foi criado com a combinação das abordagens. A efetividade mostrada pelo modelo foi satisfatória, com os valores de kappa indicando uma concordância de moderada a substancial entre o modelo e a avaliação humana. Os resultados mostraram que uma abordagem de aprendizado de máquina pode ser eficientemente utilizada na avaliação automática de respostas curtas, incluindo respostas na Língua Portuguesa.

Palavras-chave: Avaliação automática. Questões discursivas. Aprendizado de máquina. Processamento de linguagem natural.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACC	Accuracy
ASAG	Automatic Short Answer Grading
ASAP-SAS	Automated Student Assessment Prize - Short Answer Scoring
BoW	Bag-of-Words
CBoW	Continuous Bag-of-Words
CIR	*Colaborativo e Inteligente de Respostas*
CS	Computer Science
DE	Distance Education
DISCO	Extracting DIStributionally similar words using CO-occurrences
DT	Decision Tree
DTM	Document-term Matrix
EAD.BR	Census of *Ensino à Distância* in Brazil
ELMo	Embeddings from Language Models
ESA	Explicit Semantic Analysis
FS	Feature Selection
GBM	Gradient Boosting Machine
HHA	Human-Human Agreement
LDA	Latent Dirichlet Allocation
LK	Linear Kappa
LSA	Latent Semantic Analysis
ML	Machine Learning
MOOC	Massive Open Online Course
NL	Natural Language
NLP	Natural Language Processing

NLTK	Natural Language Toolkit
POS	Part-of-Speech
QD	Question Demoting
QK	Quadratic Kappa
RBF	Radial Basis Function
RF	Random Forests
RQ	Research Question
SHA	System-Human Agreement
SLR	Systematic Literature Review
SMS	Systematic Mapping Study
SRL	Semantic Role Labeling
SUS	System Usability Scale
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TE	Textual Entailment
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
VLE	Virtual Learning Environments
VSM	Vector Space Model
XGB	Extreme Gradient Boosting

CONTENTS

1 INTRODUCTION

Assessments are routinely used in learning environments in order to estimate a percentage of the retained knowledge from students. Despite its importance, teachers usually find the task of assessing lots of discursive answers very time-consuming. The evaluation work is frequently done at home, compromising the teacher's life quality [7]. Researches indicates that about 75% of some Brazilian teachers claims to frequently bring work home, like assessments of student's exams [8]. This situation overloads teachers and reduces their time, that could be spent in other activities like class elaboration [9].

Teachers work's conditions and their own human subjectivity have a great impact on grading. Humans make mistakes and some reasons for that can be from fatigue, bias or the simple ordering of student's tests [10]. Moreover, with human grading, students may have to wait for a long time to receive feedback on their answers [11] and, when they finally get it, grades can be different from another classmate's, who has given a very similar answer [12, 13].

These problems became more intense in tools like VLEs (Virtual Learning Environments) and MOOCs (Massive Open Online Courses), that have recently improved their popularity and are used by way more students than physical classes [14]. Moreover, these environments can have assessment systems that can support teachers in evaluating many students. However, the assessment of written activities is frequently performed by humans, causing the previously exposed difficulties.

Computer-based assessment came to address these issues and improve other aspects of learning by automating the evaluation process. Some of the benefits of automatic assessments are: criteria is formalized [15], can provide faster feedback to both teacher and student, can save teachers' time so they can use it to work better and allows teachers to easily follow the class performance [13]. Furthermore, automatic grading is becoming highly competitive with human grading, considering short answers [16].

Evaluations are often composed of recall or recognition type of questions, which are in different levels of the learning depth. The recognition kind seeks to test the respondent's ability to organize or identify some specific information. As for the recall ones, respondents need to remember external knowledge and write their own answers. Automatic grading is a solved problem for recognition questions, but it is an open problem and research subject for the recall kind [1].

Within the recall kind, there are questions concerning speaking, structured text (math and source code) or natural language questions. The natural language type can be classified in three groups: fill-the-gap, short answer and essay. Fill-the-gap expects

responses to be only from one to a few words, with fixed openness and focus on words. Short answers varies from one sentence to one paragraph, the focus is on the content and it has closed openness. At last, essays can have from two paragraphs to several pages, focus is on the writing style and it has a more open scope [1].

Research on the assessment of natural language questions using computers started long ago, as soon as computers started demonstrating their potential, in 1966, when concerned by the amount of manual work involved in grading essays, [17] predicted and speculated about essays being graded by computers, demonstrating its imminence. However, since then, the assessment of written activities has come a long way, and the research field has been subdivided.

Considering this division, this work exclusively focus on short answers. In addition to the length, focus and openness, short answers must be written in some natural language and recalls to external knowledge outside the question statement. This research field is defined in [1] as Automatic Short Answer Grading (ASAG). It consists in automatically assessing short natural language responses using computational methods.

Indeed, there are several ways to grade answers through computers. Some of them are not even fully automatic, using clustering algorithms to group answers so teachers can assign one grade to fit several answers. However, semi-automatic approaches are out of the scope of this work. Regarding only fully automated techniques, there are still several methods that can be employed to grade answers. This work will focus on using a machine learning approach to handle ASAG.

Several researches have been recently developed in the ASAG field. However, most of them uses English datasets, concerning the language of the questions and short answers. In its turn, ASAG research using Portuguese data is somewhat scarce. One of the reasons for this situation is the lack of public available Portuguese datasets, something that is not an issue for English research [1].

1.1 Objectives and General Methodology

Considering the presented scenario, this work has as its main objective the exploration of the Portuguese ASAG field using a machine learning approach. The specific goals are the following:

- Perform a systematic literature review of ASAG works that uses a machine learning approach;
- Develop a web system to be used in ASAG context;

- Put the system in operation to be used by several users and collect ASAG data from it. Then, publish the dataset on the web, to be publicly available;
- Build an ASAG model to evaluate on the collected data;
- Compare different approaches when performing the previous goal.

The presented goals are achieved through this work's general methodology, illustrated in Figure 1.

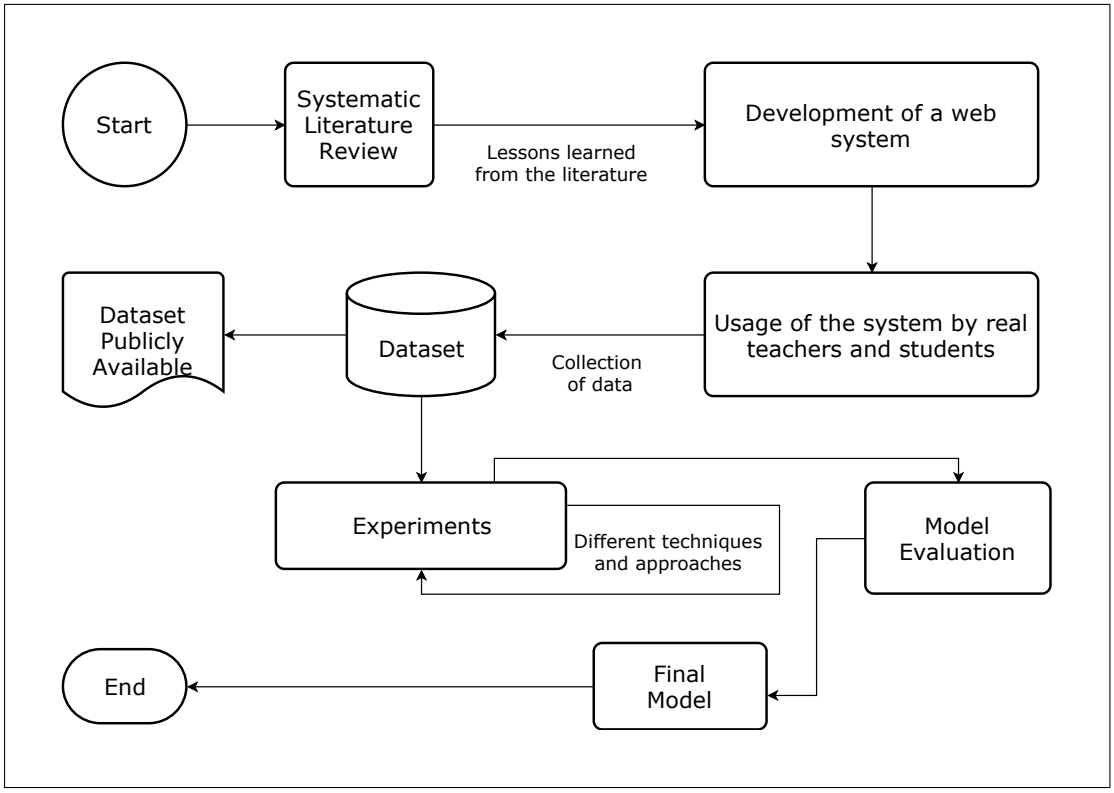


Figure 1 – General Methodology

The first step was to perform a systematic literature review in order to get a overview of already existing works on the field. With that done, a web system was modeled and developed in order to be operated by real world users.

From the system's usage from teachers and students, a dataset was collected and packaged in order to be made publicly available on the web.

From the dataset, several experiments were performed to test the performance of different approaches and techniques.

Then, the different approaches were evaluated and compared. From them, a final model was created for grading new answers, composed of different groups of approaches.

1.2 Outline

This chapter presented an introduction and overview of this dissertation. The remainder of this document is organized as follows:

- Chapter 2 presents the fundamental concepts that this work is based on;
- Chapter 3 reports a systematic literature review performed to give an overview of the ASAG field and its works;
- Chapter 4 goes over the process of data collection and its analysis;
- Chapter 5 presents the experiments performed on the collected data, their methodological procedures, results and discussion;
- Chapter 6 presents the conclusions for this work and possible future directions.

2 FUNDAMENTAL CONCEPTS

This chapter introduces fundamental concepts involved in the development of this work. It begins by presenting where the addressed problem of this work fits regarding similar areas of Computer-based Assessment research and states some definitions. Then, the following two sections gives an overview of commonly employed tools for solving problems like ASAG: Natural Language Processing and Machine Learning. Finally, the final section of the chapter discusses a new NLP/ML technique: Word Embeddings.

2.1 Computer-based Assessment

Computer-based Assessment is the research field interested in using computational methods to help in assessment activities. The use of tests to evaluate the student's retained knowledge is a common process in a wide variety of educational settings. There are many activities that can get a computer graded feedback. Different terms are used in the literature when referring to the differences between questions. In [1], following terminology used in [18], a hierarchical view of common types of questions is presented. Specifically, these are questions in which computers can contribute or even replace humans in the grading task. These question's categories are shown in Figure 2.

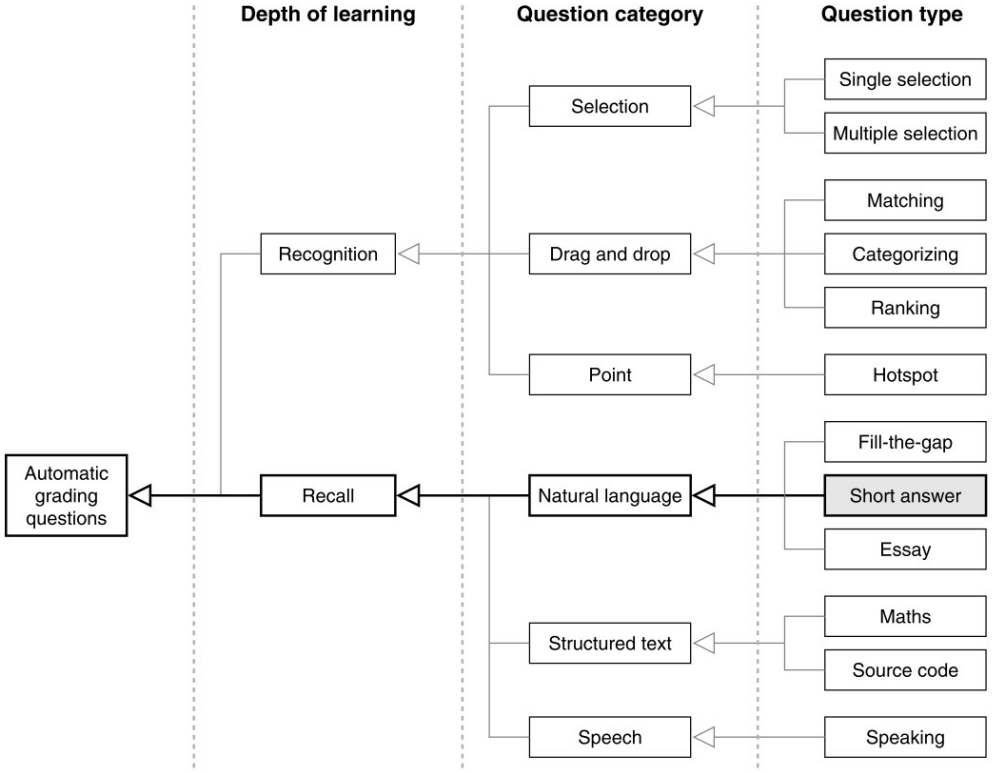


Figure 2 – Question's categories [1]

Firstly, questions are divided between recognition or recall categories, depending on the learning depth. The recognition kind seeks to test the respondent's ability to organize or identify some specific information. As for recall questions, respondents needs to remember external knowledge and write their own answers [1].

Also, as recall questions gives no options to choose from, neither provides more information beyond the question statement itself, it is harder for respondents to use test taking strategies or just guessing the correct answer [19, 20].

Secondly, Figure 2 divides question's categories and types by following the recognition and recall branches. For the recognition type, questions are divided in selection, drag-and-drop and point. These categories are subdivided in single selection, multiple selection, matching, categorizing, ranking and "hotspot". From those, one of the most commonly used for general purpose tests is multiple choice. It consists in analyzing some options and deciding which one is correct.

Regarding the recall branch in Figure 2, the division is made in three categories: natural language, structured text and speech. Speech questions concerns questions involving audio information, whilst structured text questions expects answers in the form of math or code. Natural language questions are also sub-divided and can be of three types: fill-the-gap, short answer or essay.

In [1], the distinction between the three types of natural language questions is performed considering three properties: length, focus and openness. For fill-the-gap questions, the length of expected answers is just from one to few words, the focus for assessment is in specific words and the openness is fixed. In its turn, short answers expects text being from one phrase to one paragraph, the evaluation is focused on the content and it has a closed openness. Finally, essays can range from two paragraphs to several pages, the assessment focus is on the style and it has a broader openness.

As the focus of this work, the formal definition of a short answer question is given considering [1] five criteria, as follows:

1. The question must require an answer that recalls to external knowledge, outside of the question statement;
2. The question must require a natural language response;
3. The answer's length must be roughly between one phrase and one paragraph;
4. The assessment focus must be on the content instead of writing style;
5. The level of openness should be restricted to an objective question design.

Despite the preceding definition, [1] authors considered reading comprehension questions as fitting ASAG research field, because of their very similar way of grading (even not following the "external knowledge" first definition, as the answer is within a piece of text given for the respondent to interpret).

With short answers properly defined, the other part of ASAG consists in the automatic grading. In [1], a pipeline representation was created comprising six artifacts and five processes involved in an ASAG system, as seen in Figure 3. First, a test or exam is performed in some learning setting. Then, datasets are created using the questions, answers, expected answers and grades from the tests.

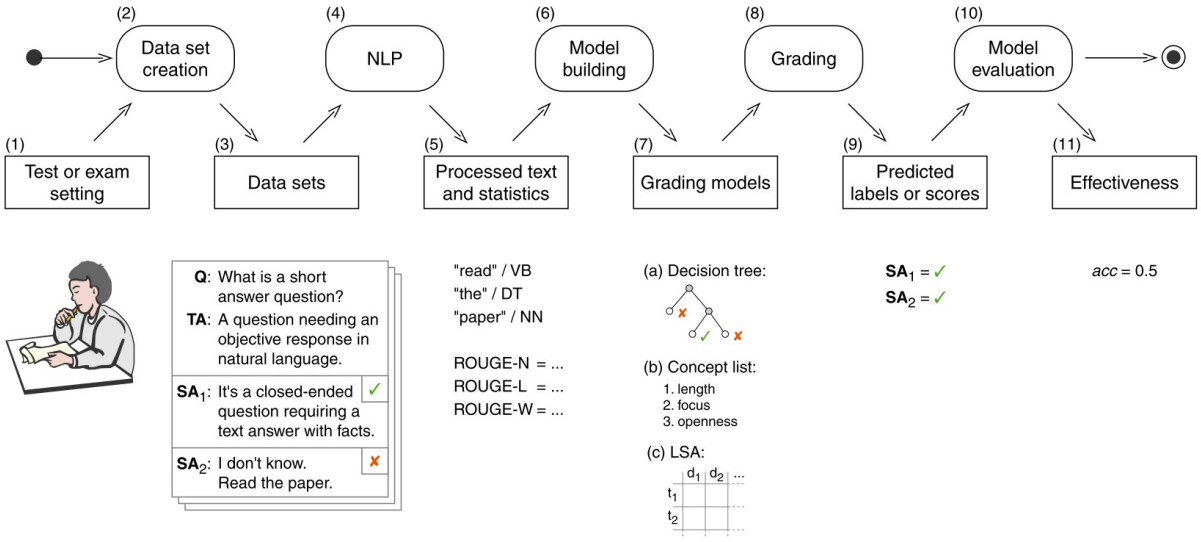


Figure 3 – ASAG pipeline [1]

In possession of the dataset, natural language processing is used to represent text in someway to fit computer requirements. Using the available processed text and statistics, a model can be built and then be used to grade new answers. This generates a new set of graded answers, that goes through an validation process, that outputs the system's performance.

Each component of Figure 3 is better explored and put to practice in the remaining of this work. In Chapter 3, a systematic review reporting all these components is presented. The remainder of this chapter addresses the components of NLP (Section 2.2) and ML (Section 2.3) fundamentals.

Considering numbers 1 to 11 from Figure 3: the exam setting and dataset creation of this work is detailed in Chapter 4 (1 to 3). Our proposed model, using NLP and ML, along with the model building and evaluation is presented in Chapter 5 (4 to 11).

2.2 Natural Language Processing

If the goal of this work is to automatically grade short answers (written in natural language), how can be text represented in numbers to be further processed by computers and machine learning algorithms? This section covers the basic understand of Natural Language Processing (NLP), a subfield of Artificial Intelligence.

NLP is the computer science field that is concerned with every aspect of the use of text by computers, written in any natural language (English, Portuguese, Spanish, etc.). Unlike formal languages (as programming), natural languages (NL) are way harder for the computer to interpret. NLs are full of ambiguities, that requires disambiguation techniques in order to be interpreted by computers. However, disambiguation and NLP are not easy tasks [21].

NLP can be categorized considering two aspects. The first is if the language in study is spoken or written. The second is if the task consists in processing the NL or generate text as output [22]. This work is focused on **written** language and the **processing** aspect.

Despite these aspects, the most representative division made for NLP is considering the level of text analysis. The next four subsections discusses each level of analysis in details. However, the specific NLP features used in ML algorithms for ASAG (also considering the four-way division) are covered by Subsection 3.4.4 in the next chapter. Also, the last subsection presents one of the most commonly used ways to represent text computationally: document-term matrices (along with bag-of-words or ngrams).

2.2.1 Lexical-Morphological

The first level of analyses is the lexical-morphological. It is concerned in analyzing each word individually, without considering anything else from a sentence. Thus, a first important task is to properly divided words. In many languages this process can be considered an easy task (because of the commonly used space separator). However, languages as Chinese have this as an extra challenge.

Words are composed of letters and syllables. Sometimes these letters are all part of the word itself, however, sometimes words have morphological variations, in result of derivations. Morphology is concerned in analyzing a word by its component parts coming from derivations. Words can vary in gender, number and be in diminutive or augmentative, among others variations [21]. Consider an example of a Portuguese root word: *gato* (cat). *Gato* is the male version of the word, the female is *gata*. If you need to refer to more than one cat you say *gatos* but if there are only females cats, it's said *gatas*. The word can also refers to the cat size, it can be a little kitty (*gatinho* or *gatinha*) or a giant cat (*gatão* or *gatona*). Joining all together, if you are talking about a group of large cats, you use *gatões*

for male and *gatonas* for females.

The cat example shows how the same word can be found in many different ways in text (10 variants, and there are more). However, the main idea of them all is to talk about a cat. As humans, is easy to capture the similarity between those words and put them all on the same "package". Nevertheless, computers need words to be exactly the same to interpret that they are referring to the same thing. This can be accomplished with the use of morphological reduction techniques such as **stemming** and **lemmatization** [23].

The differences between the two is based mainly in the resources needed, execution time and output. **Stemming** follows a crude heuristic that just chop off words' endings but without guaranteeing if it is correct. It has a fast execution time but the output is often not a real word but a cut version (example: reduction and reduce would end up being "reduc"). In its turn, **Lemmatization** has a slower execution speed, because it needs to analyze the word's morphemes and make dictionary look ups. However, the advantage of using lemmatization is that the returned word is a real dictionary word, only in its base form, also known as lemma (using the same example, reduction, reduce and reducing would all be turned into reduce) [23].

Other techniques that can act considering just words are:

- **Case normalization:** the simple normalization of upper and lowercases. This is done to enhance matches between words (e.g. Library is the same as library, it is something obvious to humans, but it needs to be informed to computers);
- **Numbers, punctuation and other symbols removal:** sometimes numbers or other symbols are dispensable to a specific application and can be removed from the text;
- **Spelling correction:** as case normalization, it is a procedure performed to enhance match between words. Even though a word is misspelled, would be of interest of many applications to consider it as the correct spelling, in order to match a search query or increase similarity between texts;
- **Stopwords removal:** a technique applied to not account for too common words. Applications using term frequency (or its variations) can be deceived to think that words like "the" are important in its context, which is usually not the case.

2.2.2 Syntactic

This level of analysis is interested in the grammatical constituents of sentences. It consists in analyzing how words are related to each other within sentences [22].

Sentences can be divided in groups, and these groups can also be subdivided in constituents. An example of a constituent analysis can be seen in Figure 4. There, the

sentence is first categorized by the part-of-speech of each word (Det: determinant, Adj: adjective, N: noun, V: verb, P: preposition) [2].

Det	Adj	N	V	Det	Adj	Adj	N	P	Det	N
the	little	bear	saw	the	fine	fat	trout	in	the	brook
Det	Nom		V	Det	Nom			P	NP	
the	bear		saw	the	trout			in	it	
NP			V	NP				PP		
He			saw	it				there		
NP			VP					PP		
He			ran					there		
NP			VP							
He			ran							

Figure 4 – Syntactical constituents [2]

Then, in the next line, *little bear* is replaced by *bear* (nominal), *fine fat trout* by *trout* (nominal) and *the brook* by *it* (noun phrase). In the third line, *the bear* and *the trout* are replaced by *He* and *it*, forming each one a noun phrase. *In it* is also replaced by *there*, creating a prepositional phrase. The last two lines follows the same process.

This is what a syntactical analysis performs, it parses a sentence and find its intermediate and final constituents, creating phrase structure trees as seen in Figure 5.

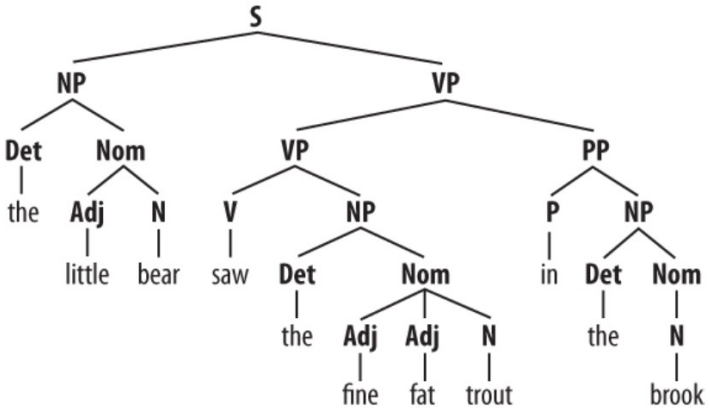


Figure 5 – Parse tree [2]

The way computers parses such trees, without getting into details, is by the use of **formal grammars** and **statistics** obtained from corpus already manually annotated (parsed) [22].

2.2.3 Semantic

Semantic analysis concerns the study of the words' meaning. It is a step further from lexical and syntactical analysis, that considers just words or their relationship within each sentence. Semantics are about how words relate to each other according to their meaning [21].

In [21], some ambiguity challenges originated from natural languages regarding word's meanings are presented as follows:

- **Homonyms:** are words that are written the same way and often pronounced the same way (in this case is also a homophony) but they have totally different meanings. Examples: *duck* (as a noun: the animal OR as a verb: to lower the head or body to prevent being hit by something) and *pile* (a mass of things grouped together OR the head of a spear or arrow);
- **Polysemy:** it is a word that have more than one meaning. It is similar to homonyms, but the difference is that in polysemy the same word have different meanings (but possibly related) and in homonyms, there are different words (with different concepts) that share the same shape. An example of polysemy is *man*, that can refer to the human species, the male equivalent of woman or even an adult human (as opposed to a boy);
- **Compositionality:** it is a principle in which the meaning of the whole can be strictly predicted from the meaning of its parts. This is a huge problem because natural languages often do not obey to this principle. An example is from the word *white*, that represents different colors in different expressions. In *white paper*, it is the actual white. In *white hair*, is usually grey. In *white skin*, it is really a rosy color. And in *white wine*, it is actually yellow;
- **Idiom:** it is a sentence where the meaning of its components are not related to the meaning of the phrase. Example: *best of two worlds* (the meaning usually do not refers to anything about any worlds, it is usually meaning advantages from two different things);

Other relationships between words' meanings that must be considered are [21]:

- **Hypernym and Hyponym:** a hypernym is a word with a broader sense, for instance, a animal, which is a hypernym of cat, dog, etc. In opposite, a hyponym has a more specialized meaning. In general, if a word $w1$ is a hypernym of $w2$, then $w2$ is a hyponym of $w1$;
- **Synonyms and Antonyms:** synonyms refers to two or more words that share the same meaning. Examples: intelligent and smart or rich and wealthy. In its turn, antonyms are words that have opposite meanings. Examples: hot and cold or long and short;
- **Meronym and Holonym:** when two words have a part-whole relationship they are meronym and holonym of each other. Examples: the word *tire* is a meronym of *car*, whilst *tree* is a holonym of *bark*.

All the preceding relationships between words demonstrates the complexity of natural languages, specially when dealing with the associated meaning of words, in a sentence context. In order to model all these relationships and help computers to better process human languages, WordNet was created [24].

WordNet is a large database of the English language. It is modeled using synsets, which are sets of cognitive synonyms expressing a specific concept. Its strength lies in the interlinked structure linking synsets through their conceptual-semantic and lexical relations (as the ones just listed above). Its structure forms a semantic network, that can be used by many computational linguistics and natural language processing applications [24].

2.2.4 Discourse and Pragmatic

This subsection only gives a note on pragmatics as it is related to longer texts, out of the scope of this work (that deals with short answers).

A **discourse** is a sequence of linked sentences. In a logical sequence of sentences, it is common that the current sentence depends on the preceding one. One of the main challenges in discourse analysis is to resolve anaphoric pronouns such as he, she and it. Considering the following discourse: *Angus used to have a dog. But he recently disappeared.* It is more possible that the missing agent is the dog, but it could also be the man (Angus). It is harder for computers to decide in such ambiguities.

The discourse analysis is part of **pragmatics**, a study field concerned with how knowledge about the real world interacts with literal meanings. Considering the example about the dog and its owner, the reader only interpret the dog as missing because he knows that people lose dogs, not the opposite. The same goes for the example: *Angus used to have a dog. He took him for walks in New Town.* In this case, him is referring to the dog, as a reader knows that humans take dogs for walks and not the other way around.

2.2.5 Document-term Matrices, Bag-of-Words and Ngrams

One of the most used and intuitive ways of representing a set of documents computationally is by the use of a document-term (or term-document) matrix (DTM). In it, each row represents a document and each column a term. Documents can be any piece of text such as a sentence, an article, a chapter, a short answer and etc. In its turn, the terms are the words present in all the documents [23]. The following example will be used to illustrate this subsection:

Document1 = "the man wrote a letter to the lady"
Document2 = "the letter was written for the lady"

A DTM can be weighted in many ways [23]. Three of the most used are: **1)** by binary presence (1) or absence (0) of words **2)** using term frequency to weight the matrix (like in Table 1) and **3)** using the term frequency-inverse document frequency (tf-idf) to weight the matrix (this weighting scheme is used to not give too much importance to terms frequently used in a set of documents).

Table 1 – Term-frequency weighted matrix

X	**the**	**man**	**wrote**	**a**	**letter**	**to**	**lady**	**was**	**written**	**for**
D1	2	1	1	1	1	1	1	0	0	0
D2	2	0	0	0	1	0	1	1	1	1

If both documents went through a lemmatization and stopwords removal procedure, the correspondent output and matrix would be much smaller, as seen in Table 2.

Document1' = "man write letter lady"
Document2' = "letter be write lady"

Table 2 – Reduced matrix

X	**man**	**write**	**letter**	**lady**	**be**
D1'	1	1	1	1	0
D2'	0	1	1	1	1

Both the last two matrices, beyond being document-term matrices, are also a **Bag-of-Words (BoW)** representation from text. It is characterized by the representation of words disregarding order and grammar, but keeping their frequency. However, BoW is only a specific case of the **ngrams** modeling. Ngrams parses the input text as a sequence of m n-grams together. Following with the example, 2-grams would transform D1' and D2' into:

Document1" = ["man write", "write letter", "letter lady"]
Document2" = ["letter be", "be write", "write lady"]

Table 3 – Ngrams matrix

X	**man write**	**write letter**	**letter lady**	**letter be**	**be write**	**write lady**
D1"	1	1	1	0	0	0
D2"	0	0	0	1	1	1

Moreover, using ngrams with words as in the previous example, there are also character or syllable ngrams. Producing character 5-grams for D1' and D2' results in:

Document1"' = ["man w", "an wr", "n wri", " writ", "write", ..., "r lad", " lady"]
Document2"' = ["lette", "etter", "tter ", "ter b", "er be", ..., "e lad", " lady"]

2.3 Machine Learning

Machine learning (ML) is a research field within Artificial Intelligence from Computer Science. It can be defined by the use of computers to make predictions considering some existing data [25]. The goal of ML algorithms is to discover patterns in data, analyzing the available data and its relationships in order to predict for future situations [3].

As the center of machine learning, data can present itself in different shapes and characteristics. In order to make sense from them, it needs to be modeled as a logic and consistent set of **features** that can better describe its essence for computers. Once the features that represents the nature of data are identified, they can be used in learning algorithms that will attempt to find relationships between features from the **samples** (each instance of the data) and possibly **labels**.

However, in order to find patterns on data, it needs to be available in a large amount and be very representative of the task setting. In other words, for creating a data set, a large amount of samples (or data points) are required, where each instance is represented by a feature set, forming a matrix structure.

As a vast research field, machine learning can be divided considering some aspects. Its algorithms can be divided in three categories depending on the type of its learning [3]:

- **Supervised Learning:** it is when the learning is driven by the given output labels. Knowing the desired output label for each instance, learning algorithms tries to find patterns in features that can lead to these specific outputs;
- **Unsupervised Learning:** used when there is no labels in the data, in other words, when data is not categorized or there is not a specific value as output. In this case, the goal is to find hidden patterns, leaving for the learning algorithm to make sense from the data;
- **Reinforcement Leaning:** in this category, the learning process is conducted in a different manner. It consists of training a algorithm by offering it a "reward" or a "punishment" for the actions performed by the current state of learning.

ML can also be divided in three categories by considering the desired output for a task, being [26]:

- **Classification:** consists of a supervised learning task when the given labels are made of discrete categories, often called **labels** or **classes**. The goal of a classification algorithm is to categorize new samples into one of the possible classes;

- **Regression:** another supervised learning category, it is used when the desired output is a continuous variable. That is, values assumed by the output are not discrete classes, they are inside a range and can assume any value;
- **Clustering:** being an unsupervised learning, it is performed when no labels are available (neither discrete nor continuous). In this case, the goal of the learning algorithm is to cluster samples into groups. In other words, its output consists of assigning each new sample as belonging to a specific group, not previously known before the training phase.

From this point on, in this section, the same data example will be used to illustrate the addressed concepts. It consists in a dataset of 150 iris flowers, comprising their petal and sepal lengths and widths (totalizing 4 attributes for each flower). Moreover, each flower is classified in a specific category, according to its species: Setosa, Virginica and Versicolor.

An image with common terminology used in ML datasets can be seen in Figure 6 (using the addressed example). Sepal and petal are exemplified in a iris picture. Then, the 150 instances (or observations, samples) are disposed in rows, with each row comprising of four features (or attributes) from the flowers. Finally, in the last column of each row, there is a label representing each iris' class (its correspondent specie).

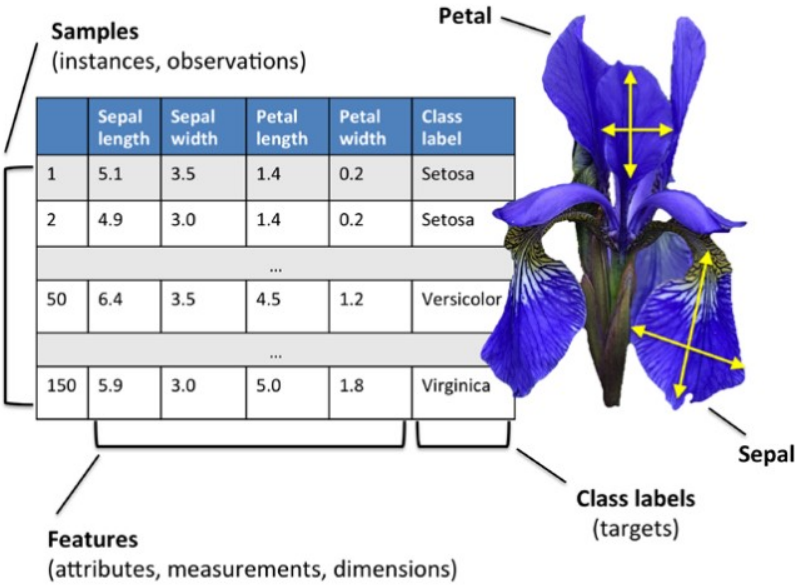


Figure 6 – Dataset terminology with iris example [3]

One goal of this iris dataset can be to discover if: with the available features and labels, can an algorithm make correct predictions about new data? Considering the labels, it is possible to tackle this challenge using **supervised learning**. Considering that the desired output consists in three categories, this can be handled by a **classification**

algorithm. An example of how to create and use a machine learning model with supervised learning can be seen in Figure 7.

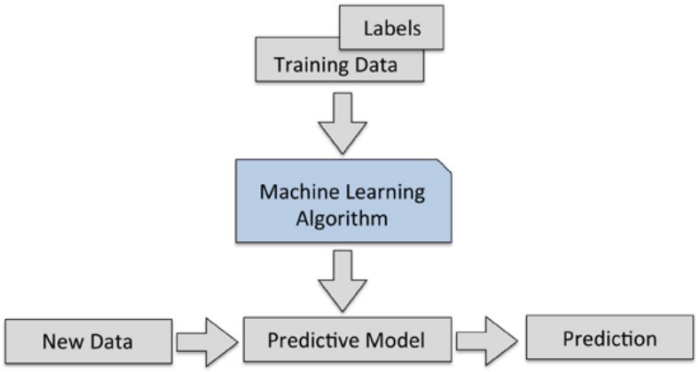


Figure 7 – Training and using a ML model to make new predictions [3]

As this work also deals with a classification task using supervised machine learning, despite all machine learning algorithms, **Decision Tree** (DT) [27] was chosen due to its specific characteristics, performance and usage in others algorithms.

Considering different machine learning algorithms, DT is one of the most used and easy to understand [28]. Decision tree is similar to a flowchart, in some ways. It is composed by nodes and lines (called branches). Nodes can be of two types: internals and finals. Internal nodes represents a decision and, each branch leaving a node represents one made decision. Final nodes stands for a final decision, representing the possible classes. An example of a built decision tree using iris data can be seen in Figure 8.

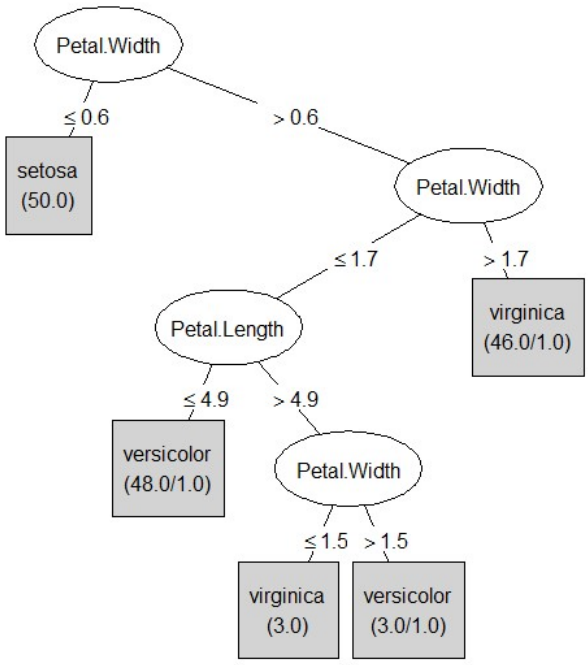


Figure 8 – Decision tree with iris example [4]

Despite DT's good performance and easy to understand models, it has a major disadvantage: it is prone to **overfit**. This is a problem in machine learning algorithms when the built model performs good, but is too fitted to the given data. Thus, when used on new data, the model performs poorly [28].

In order to aid with the overfitting problem, another algorithm was created, called **Random Forests** (RF) [29]. The idea is to use a great amount of decision trees, each one built with a different subset from the original data, take predictions from each tree and then defining a final prediction based on the majority vote from all the trees.

Random Forests is an example of an **ensemble learning** algorithm. The concept behind ensemble learning techniques is to combine results from several "weak" learners to build a more robust model, a "stronger" learner that can generalizes better and be less susceptible to overfit [25].

Another ensemble learning algorithm, also usually based on decision trees, is **Gradient Boosting Machine** (GBM) [30]. A great difference between RF and GBM is that the former is a bagging algorithm, whereas the other is a boosting algorithm. Bagging uses several independently models and combine them after each one has finished. On the other hand, boosting algorithms are sequential, that is, models are created sequentially, one after the other, and the error from the former is handled by the next. This is the case for GBM, that in each iteration, using a single model, tries to improve the current model based on the error from the previous one.

In order to summarize and present an overview of the whole process, Figure 9 shows a commonly used pipeline when creating and using machine learning models.

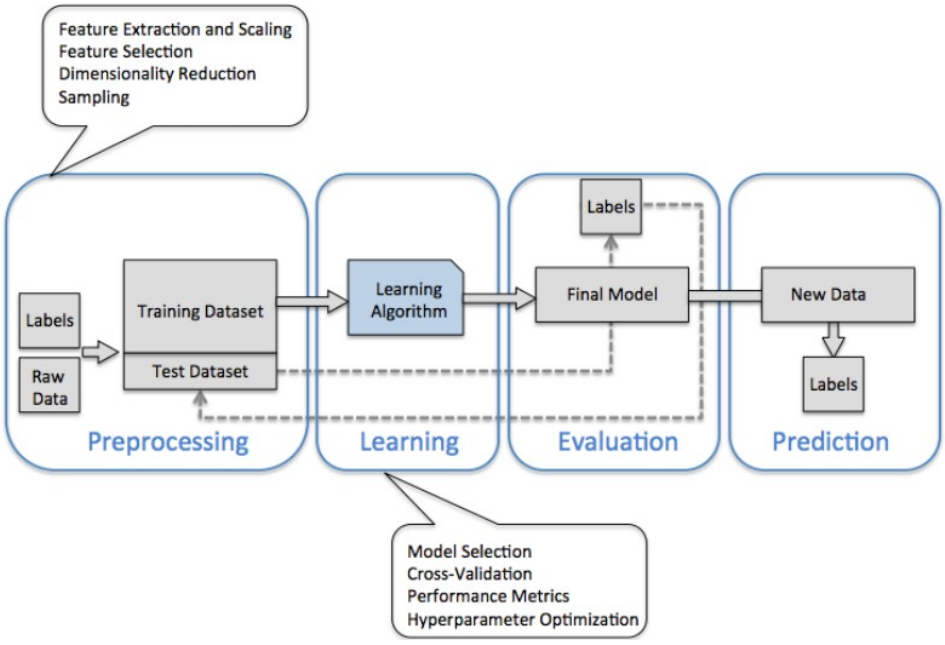


Figure 9 – A commonly used pipeline for creating and using ML models [3]

Each process from Figure 9 is detailed below:

- **Preprocessing:** in this first step, the data is processed and organized in order to be used as a dataset. As ML algorithms works with numbers, data consisting of categorical variables, text, image and etc. need to be somehow converted to numbers. This is the work of feature extraction, the process of transforming the data for being used by further algorithms. Other preprocessing techniques consists in handling missing values, outliers and scaling different features to be on the same range;
- **Dimensionality Reduction:** it is used when the original features are highly correlated and therefore redundant. As this situation can impair the learning process, dimensionality reduction techniques extracts features from a higher dimension to a lower one with less correlation between features;
- **Feature Selection:** another procedure to reduce the number of features. This is done when some of the used features are not helping or even impairing the learning process. In this case, they are completely removed;
- **Learning:** after the previous procedures were performed, the data is ready to be used as input to a machine learning algorithm. Which will, in its turn, return a trained model, ready to be used;
- **Prediction:** in possession of a trained model, it is possible to use it to make predictions on data that has never been seen before.

By performing the exposed procedures, one can successfully create and use a machine learning model. However, in order to assess its quality, it needs to be somehow evaluated. This can be achieved with the use of **performance metrics** and the **model evaluation**. Some of the most used performance metrics are [3]:

- **Confusion matrix:** it is a matrix of size $n \times n$ in which the predicted samples classes are reported in contrast to their actual classes. In Figure 10 it is possible to see a 2×2 confusion matrix. In this particular case, each matrix quadrant is named as shown in the figure. However, generalizing for the n value, the main diagonal contains the correctly predicted samples and the other cells contains incorrect predictions. In the example of the iris confusion matrix (Figure 11) the correspondent interpretation is the following: the model has predicted 50 setosas as being setosas, 48 versicolors as being versicolors and 49 virginicas as being virginicas. However, it also predicted 1 virginica as being versicolor and 2 versicolors as being virginicas;

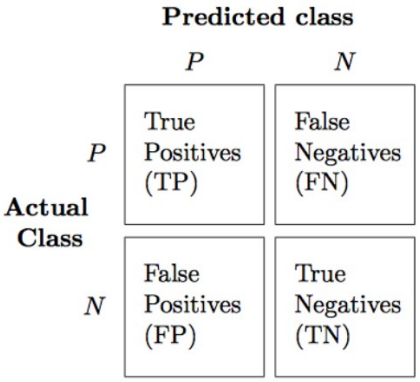


Figure 10 – A confusion matrix [3]

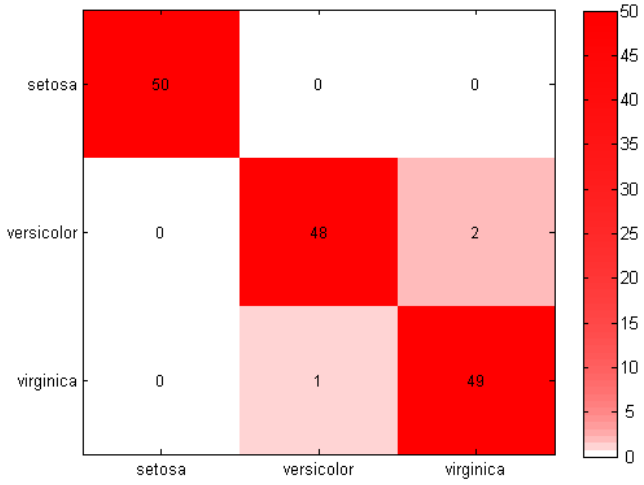


Figure 11 – Confusion matrix - iris example [5]

- **Accuracy:** is defined by the number of correctly predicted samples over the total of instances:

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.1}$$

In the example from Figure 11, the accuracy is defined by:

$$\frac{50 + 48 + 49}{150} = 98\% \tag{2.2}$$

- **Precision:** considering a specific class x, the precision of x is calculated by the number of samples that were correctly classified over all the samples classified with the same class. In other words, from the confusion matrix:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

- **Recall:** it measures the proportion of correctly predicted samples from a specific class x over all the samples that actually have the same class x. As equation:

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

- **F1:** it is the harmonic mean between precision and recall scores. It is defined by:

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall} \tag{2.5}$$

- **Cohen's Kappa Coefficient:** it measures agreement between two different raters (or the actual and predicted values) from a set of n items (or samples) classified into k categories [31]. The interpretation of the obtained value is defined by [32] as: < 0 - having no agreement, 0 to 0.2 - slight, 0.21 to 0.4 - fair, 0.41 to 0.6 - moderate, 0.61 to 0.8 - substantial and 0.81 to 1 - almost perfect;

- **Pearson's Correlation Coefficient:** it measures the linear correlation between two variables. It is calculated dividing the covariance of the two variables by the product of their standard deviations [33];
- **Spearman's Rank Correlation Coefficient:** it is similar to the Pearson's correlation coefficient, but in the Spearman's the relationship between variables can be linear or not. It is calculated in a very similar way to Pearson's, but the rank of the variables are used instead of themselves [34].

Despite all the preceding performance metrics, a machine learning model is still not ready to be evaluated. Assessing the model on the same data that it was trained for is not the correct way of validation. Different aspects of the training set used to build the model can influence in its ability at predicting future samples.

In order to properly validate machine learning models, some techniques can be used to avoid overfitting or underfitting and correctly show the model's performance:

- **Holdout:** it splits the original dataset into a training and test set with a specified percentage proportion (examples: 70/30 or 80/20) [26];
- **K-fold cross-validation:** it uses the splitting idea from holdout, but it repeats the process k iterations. At each iteration, a kth piece of the dataset is used for testing and the rest is used for training. In the end, the average of each iteration is considered to be the model's performance [25];
- **Leave-one-out cross validation:** with this technique, only one instance is used as test set and all the other samples are use as training set. Like k-fold, the process is repeated. However, in this case, each of the n instances will be the test set, and n iterations will be performed, one for each [25].

Finally, counting on a reliable way to test the model's performance, it is possible to improve it a little further. Some machine learning algorithms like Random Forests and Gradient Boosting have what is called **hyperparameters**. These are parameters from the internal structure of the algorithms, that possesses a default value. However, as a parameter, it can be changed, and by changing, it is possible to significantly improve or decrease the model's performance. This process is usually referred to as **parameter tuning**.

2.4 Word Embeddings

Subsection 2.2.5 expatiate in one of the most common ways of language modeling for computer interpretability. Using a bag-of-words, ngrams and document-term matrix approach can be very useful, despite its simplicity. However, more powerful and sophisticated methods are available. In order to understand them, it is necessary to go through the history of representing words in vectors and the issues with simple representations.

The representation of words as presented in Subsection 2.2.5 has some issues that might be relevant to be addressed in specific applications. Firstly, words that are semantic related have different columns in DTM matrices, that do not represent the meaning relation between these different but related words. Secondly, each new word in the vocabulary constitutes one more dimension to be represented in DTMs, which is a problem for vocabularies of thousands or millions of different words [35].

New approaches for constructing more effective Vector Space Models (VSMs) and addressing the aforementioned issues made use of the co-occurrence of words based on large corpus. The idea that words sharing similar context also share similar meaning was proposed in the 1950s by [36] and [37] in the distributional hypothesis linguistic theory. This idea enabled the development of new approaches such as Latent Semantic Analysis (LSA) [38], a method that applies Singular Value Decomposition on a co-occurrence matrix to reduce its dimensionality.

A great resource used by LSA for tackling the high dimensionality and semantic representation issues was to use a large corpus in order to extract relevant information. By mixing the distributional hypothesis theory, the large amount of available text data in recent years (Google News) and shallow neural networks architectures arises Word2Vec [39]. This work popularized the term *Word Embedding* and presented two novel model architectures for continuous vector representation (CBow and Skip-gram).

The CBoW model trains the neural network to predict a target word based on its contextual words (that is, it uses the words from before and after the target word, based on a window of a specific size w). Oppositely, Skip-gram uses the reverse idea and uses one word to predict the surrounding words in the window [39].

The Word2Vec model solved the dimensionality issue: in this novel method, there is a parameter for defining the desired number of dimensions (usually tens or hundreds of dimensions, in contrast to thousands or millions from sparse models). This low dimensionality reduces the computational power needed to process words. Moreover, the improved efficiency of Word2Vec is verified in many NLP tasks and applications [35].

However, beyond the aforementioned benefits of Word2Vec, it became widely known by its capacity of modeling semantics in a way that it enables to resolve analogies such as: *king* is to *man* as *queen* is to *woman* [39]. These semantic regularities can be

obtained from simple algebraic operations between words' representations. The previous example comes from the equation: $vector(\text{“}king\text{”}) - vector(\text{“}man\text{”}) = vector(\text{“}queen\text{”}) - vector(\text{“}woman\text{”})$. This can be verified by changing any of the four elements by an unknown variable and performing the calculation. The resulting vector is the closest to the vector from the expected word.

Another possible usage from this vector calculation feature is to find the most similar words for a given word. For instance, a Word2Vec model might return the words *emperor, prince, tsar, despot* for the word *king*. It could even return *cat, bird, feline, ferret* for the word *dog*. These similarities and analogies between words can be visualized by projecting the vectors in two dimensions, as in Figure 12.

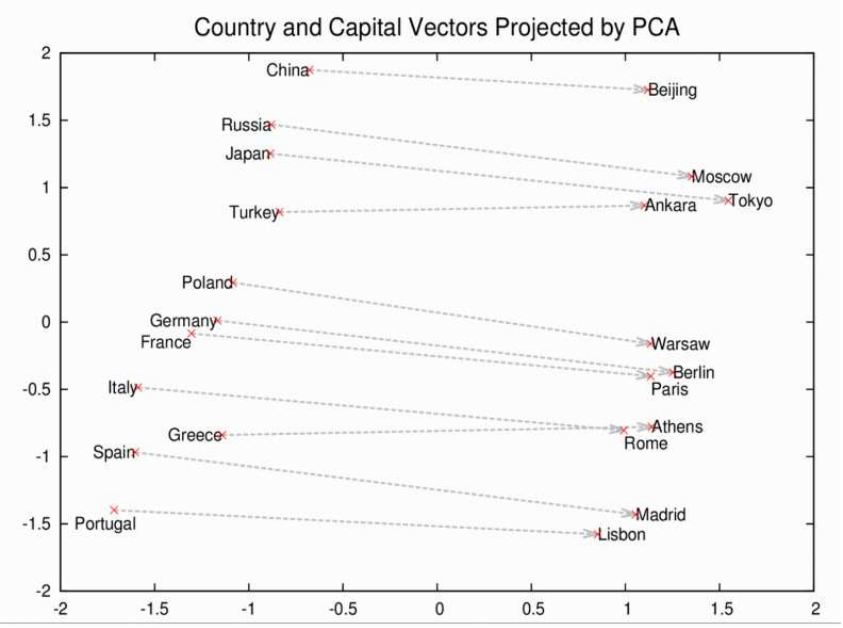


Figure 12 – Vectors projected in 2D: example [6]

One more property of Word2Vec is the Additive Compositionality. If the vector representation of two words is summed, their sum can correspond to a meaningful concept. For instance, adding “Russian” and “river” together can produce a vector very close to “Volga river” (a river located in Russia) [40].

The Word2Vec is a model that can be created in a automatic way, simply by providing it with a large corpus like Google News. It opposes to knowledge-based approaches for semantic modeling, such as WordNet. The benefits of Word2Vec upon WordNet is that it requires no human labor, it uses the context in which words appears on text (while WordNet does not) in their representation and it has a greater adaptability to other domains. The disadvantage is that Word2Vec has less interpretability than WordNet, because its senses are built based on corpora, as opposed to WordNet's manually defined senses [35].

3 SYSTEMATIC LITERATURE REVIEW

This chapter presents a Systematic Literature Review (SLR) in the ASAG study field. When searching for surveys, six literature reviews on the subject were found [41, 34, 42, 43, 1, 44]. However, two of them [34, 43] have essay systems mixed along with the short answer ones. In its turn, [44] reviews only studies that used an Information Extraction approach for ASAG. The other three surveys reviews only automatic short answer grading systems and without restrictions on the approach, especially [1] that has the most recent and comprehensive review. Despite all these surveys, this chapter presents a systematic approach (not found in the others) for conducting the literature review based on [45] guidelines. Inspired by [1], who has already conducted a comprehensive review, and by [44], that reviews focused on only one approach, we limited the scope by reviewing only studies that used a machine learning approach to solve the problem.

This chapter is structured as follows. Section 3.1 presents the methodology used to conduct the research. Section 3.2 goes over details of the planning step. Section 3.3 reports the accomplished conduction process. In Section 3.4 the results of the review are presented. Finally, closing the systematic review, in Section 3.5 a summary from previous sections is presented. In order to complement the systematic review (first conducted in early 2017), Section 3.7 reports an update considering all of 2017 and 2018's works. Additionally, a review of Portuguese works is also performed and reported in Section 3.6.

3.1 Methodology

According to [45], systematic reviews can identify, evaluate and interpret all available research concerning a specific research question, topic area or phenomenon of interest. It can present a fair review of the research topic by using a rigorous, trustworthy and auditable methodology. The process defines a research protocol in which researchers have to follow when conducting the research. The detailed and replicable aspects of systematic reviews are their main advantage since other researchers can follow the conducted process and even repeat the research obtaining the same results (considering same period).

Another kind of review that can be considered to provide an overview of a literature topic is the Systematic Mapping Study (SMS), sharing some common aspects with SLR and differing in others [46, 47, 48, 49]. Some of their differences are:

- SMSs are usually broader than SLRs considering the studied topic area and in the number of considered works [49, 46];

- A SLR generally deepens into the analysis of each work, whilst SMS only provides more surface analysis like work's identification and classification [46, 48];
- For a SMS there is no need of a rigorous quality assessment criteria for the works in analysis [48];
- On a SMS, categories must be identified, whilst a SLR can focus on only one of the identified categories [47, 48].

To sum up, SMSs are broader than SLRs, in the sense that the former analyzes more works in a more general topic and SLRs focus on a more specific topic and lesser works. Also, a systematic literature review performs a deeper analysis of each work individually than a systematic mapping study.

In this work, a kind of mixture of SMS and SLR is performed to review the literature. Most of the research questions defined for the review here presented are SMS's kind, classifying, categorizing and comparing works in tables and graphics. However, one of the questions involves a deep analysis of primary studies to be answered, an aspect typically of a SLR.

Also, the methodology for this work is heavily based on systematic literature review's procedures of [45]. However, two methodological steps are not performed in a rigorous manner required by SLR: the research protocol validation and quality criteria assessment.

For this review, the research protocol is only validated with the advisor of this work, a researcher with expertise in the application of computing in education. No further validations are performed (such as with externals specialists). Also, the quality criteria is presented and applied, but not in a rigorous procedure such as defining and grading conditions, summing grades and defining a threshold value to accept or reject works (as it is done in systematic reviews [46]). The quality criteria is based on whether the work in analysis can answer to most of the defined research questions or not.

All of the aforementioned *design decisions* made for this work are presented in other to call attention for the validity of the review and its possible threats. This is important because any difference in the methodological procedure could lead to a different review's result. All things considered, the following sections presents the steps performed in the review and the results achieved.

3.2 Planning

The planning stage of a systematic review elaborates the review protocol, which specifies the methods that will be used before starting the review. Such early definition

helps researchers avoid a biased process. The protocol (in the following subsections) includes the objective, research questions, keywords and synonyms, the sources definition, data extraction strategy (based on research questions) and studies inclusion, exclusion and quality criteria definition [45].

The protocol of this systematic review was elaborated by the conductor of the process and validated only by one researcher, a specialist on the research field of the application of computers in education.

3.2.1 Objective and Research Questions

This systematic review seeks to study, explore and understand the current state of the art of automatic short answer grading, considering works that used a machine learning approach to handle ASAG. The research questions elaborated to address the review objective are presented in Table 4. Within some of the research questions, sub-questions were also defined, as shown in Table 5.

Table 4 – Research questions

ID	Research Question
RQ1	What is the temporal distribution of the studies?
RQ2	What is the datasets' nature?
RQ3	Which natural language processing techniques are used?
RQ4	What are the selected features?
RQ5	Which machine learning approaches are employed?
RQ6	How are the results obtained and how do they compare with human grading?

Table 5 – Research sub-questions

ID	Research Sub-question
RQ2.1	What is the knowledge area addressed in the questions?
RQ2.2	Which natural language is used in the questions and answers?
RQ2.3	Are the questions applied in school or college?
RQ2.4	How old are the respondent students?
RQ2.5	How many questions are involved in the study?
RQ2.6	How many reference answers (provided by teachers) were used in each question?
RQ2.7	How many answers are there for each question?
RQ2.8	What is the grading scale of the questions?
RQ2.9	What is the average size of the questions?
RQ6.1	Which metric is used to evaluate the system?
RQ6.2	What is the human-human agreement about the given score?
RQ6.3	What is the system-human agreement about the given score?

3.2.2 Sources (Online Databases) and Search String

The sources of this systematic review are the following nine online databases: *LearnTechLib*, *Microsoft Research*, *ScienceDirect*, *IEEE Xplore Digital Library*, *ACM Digital Library*, *Scopus*, *Springer*, *Semantic Scholar* and *Keele University Library*.

As the other literature reviews on the ASAG subject do not present a systematic procedure, there are no previous search strings to base the creation of a new one. Considering this, some preliminary research was made to determine the most used words for the subject matter. Similar words were grouped and a search string using boolean operators was created and refined using one of the online databases until it was considered good despite all the possibilities that those keywords creates. The search string composed by the keywords and synonyms is:

("automatic assessment" OR "automatic scoring" OR "automatic marking" OR "automatic grading") AND (short OR "short answer" OR "free text" OR free OR text) AND (response OR question OR answer)

The above string was used in all sources, but some of them have different ways of representing the boolean operators. In some cases the string was divided in two or four, but keeping the string identity with its respective result set, in order to accomplish the database interface restrictions. English results filter was used where possible to match the exclusion criteria presented in the next subsection.

3.2.3 Inclusion, Exclusion and Quality Criteria

The Inclusion (I), Exclusion (E) and Quality (Q) criteria used when filtering the works is detailed in Table 6.

Table 6 – Inclusion, exclusion and quality criteria

Type	Criteria	Type	Criteria
I	Studies written in English	E	Studies written in another language than English
I	Journal, Conference or Methodology papers	E	Studies that do not match the research questions
I	Studies relevant to the subject matter	E	Papers about the same study or system
E	Secondary studies	Q	Are most of the research questions answered?
E	Semi automatic approaches	Q	Is the research methodology properly exposed?
E	Studies that assess essay length answers	Q	Are all the used techniques properly described?

3.2.4 Data Extraction Strategy

The data extraction strategy consists in the elaboration of a record for each study under review. This record contains bibliographic fields, the research questions and sub-questions fields and some general observations about the paper. All fields will then be filled whenever possible.

3.3 Conduction

3.3.1 Search in the Online Databases

In this stage, the defined search string was used to perform the research in the nine online databases. The results were exported from the databases in some bibliography reference format like bibtex, RIS or CSV. To easily handle these files and their content, a software called StArt (State of the Art through Systematic Reviews [50]) was used. This software was developed by the software engineering research team of the Federal University of São Carlos. The StArt tool helps researchers in the process of planning, executing and summarizing a review.

The sum of the retrieved results from the nine databases was 6789. From those papers, 1562 consisted of duplicated papers due to papers that are in more than one online database. StArt has a tool that automatically detects the duplicates by comparing the important reference fields like the title. In Figure 13 it is possible to see the databases search result numbers without the duplicates. The large number of results is due to the broad range created by the string. The search string fits good for getting wanted results, but it also gets other areas of research like medicine. For instance, one possible form that the search string can assume is "automatic scoring short response" which can refer to analyses about the performance of medical tools for measuring short body responses like stimulus or impulses. This explains the large number of initial results.

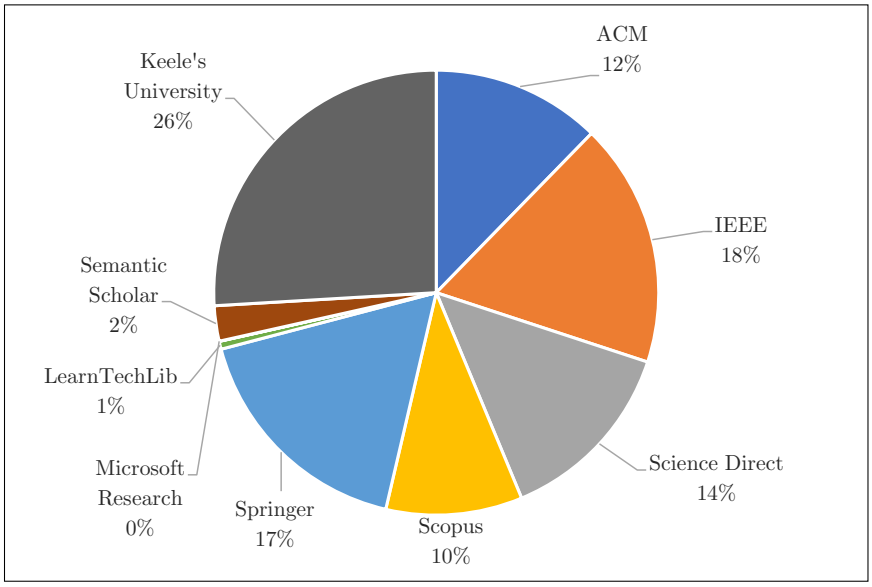


Figure 13 – Papers' sources

3.3.2 Inclusion and Exclusion Criteria Application

In sight of the 5227 remaining papers, the inclusion, exclusion and quality criteria were applied in three levels. First, the inclusion and exclusion criteria were applied only in

the title and keywords (and abstract if necessary). This filter reduced the numbers to 182. These papers were related with the research but not necessarily addressing all criteria. In the second stage, the title, keywords and abstract were carefully read to identify the relevant studies. That stage left 112 remaining papers, which were all downloaded with the exception of papers that we did not have access to (five). The remaining 107 papers were read following the first step defined in [51], which consists of reading the paper from four to ten minutes passing by the title, abstract, introduction and section and sub-section headings. The results, figures and references were also glanced to determine if the paper would pass to the next step. This technique resulted in 75 remaining papers.

Among these 75 studies, only papers that described the use of a machine learning approach to handle ASAG were selected. This was done by looking at each paper's abstract and introduction and searching for keywords like *"machine learning"*, *feature*, *classifier*, *regression* and similar. After this procedure, 18 papers remained. Knowing that due to the nature of the field a variety of keywords could be used in the studies and not being present in those 18 papers, we looked up for studies using the machine learning keywords in the six identified review papers mentioned in the introduction of this chapter. We gathered 26 more papers by looking in their references and 14 more looking at papers that cite those reviews in Google Scholar or in the references of the 14 recently acquired papers (from 2014 to 2016). This left us with 58 papers. The final filtering process can be seen in Figure 14.

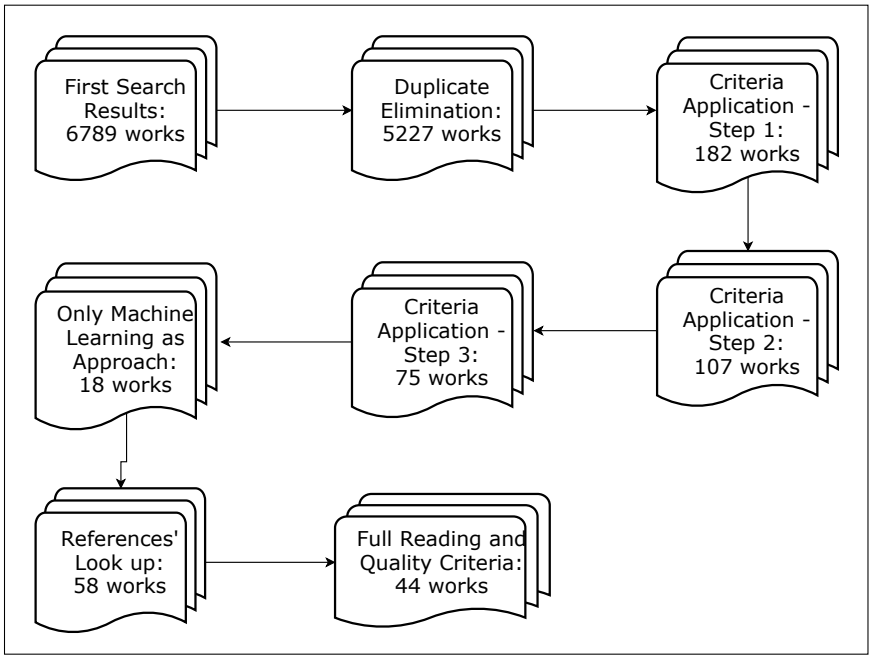


Figure 14 – Work's Filtering

3.3.3 Data Extraction

In possession of the remaining 58 papers, they were fully read in order to apply the quality criteria and at the same time do the data extraction step. A record was created to fill with the information to be extracted of each paper. The fields of the record are bibliographical info, research questions and sub-questions and general annotations. After the quality filter, the number of papers was finally established in 44. From these 44 studies, the answers of the research questions were obtained, the data was summarized and the results created.

3.4 Results

In this section, the results of the review are presented. Firstly, the 44 selected papers are shown in Table 7, with their IDs and references. Then, each subsequent subsection will answer to one of the research questions defined in the planning stage.

Table 7 – Selected papers (IDs and references)

ID	Reference	ID	Reference	ID	Reference
1	[Rosé et al. 2003][52]	**16**	[Peters and Jankiewicz 2012][53]	**31**	[Higgins et al 2014][54]
2	[Pulman and Sukkarieh 2005][55]	**17**	[Sil et al. 2012][56]	**32**	[Aldabe et al. 2015][57]
3	[Makatchev and VanLehn 2007][58]	**18**	[Dzikovska et al. 2012][59]	**33**	[Sakaguchi et al. 2015][9]
4	[Nielsen et al. 2008][60]	**19**	[Madnani et al. 2013][61]	**34**	[Nye et al. 2015][62]
5	[Wang et al. 2008][63]	**20**	[Levy et al. 2013][64]	**35**	[Luo et al. 2015][65]
6	[Lee et al. 2009][66]	**21**	[Heilman and Madnani 2013][67]	**36**	[Sorour et al. 2015][68]
7	[Sukkarieh 2010][69]	**22**	[Jimenez et al. 2013][70]	**37**	[Ramachandran et al. 2015][71]
8	[HOU and TSAO 2011][72]	**23**	[Bicici and van Genabith 2013][73]	**38**	[Zesch and Heilman 2015][74]
9	[Mohler et al. 2011][75]	**24**	[Gleize and Grau 2013][76]	**39**	[Zhang et al. 2016][77]
10	[Meurers et al. 2011a][78]	**25**	[Ott et al. 2013][79]	**40**	[Magooda et al. 2016][80]
11	[Meurers et al. 2011b][78]	**26**	[Kouylekov et al. 2013][81]	**41**	[Sultan et al. 2016b][82]
12	[Zbontar 2012][83]	**27**	[Horbach et al. 2013][84]	**42**	[Roy et al. 2016][85]
13	[Tandalla 2012][86]	**28**	[Leeman-Munk et al. 2014][87]	**43**	[Liu et al. 2016][11]
14	[Conort 2012][88]	**29**	[Gomaa and Fahmy 2014][89]	**44**	[Sultan et al. 2016a][90]
15	[Jesensky 2012][91]	**30**	[Moharreri et al. 2014][92]		

3.4.1 RQ1: Temporal distribution

At conducting the review, some interesting facts about the development in the field were noticed. The beginning of ASAG research focused mainly on rule-based methods, concept mapping, manually written patterns, information extraction and similar approaches, as pointed in [1]. This fact can be noticed in Figure 15 as between 2003 and 2010 there is about only one work per year that uses machine learning and filled our criteria.

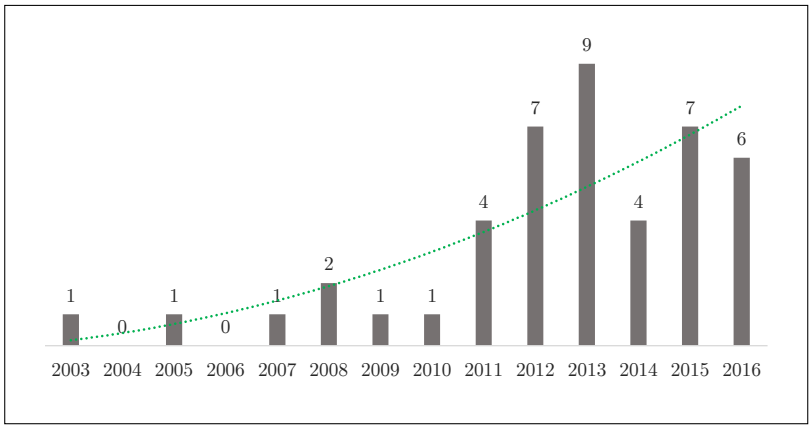


Figure 15 – Work's temporal distribution

In 2011, the ASAG scenario starts to change when the first three publicly available datasets were released (the Texas dataset [75] and the CREE and CREG datasets from the CoMic project [78, 93]). These datasets opened possibilities for future works fair comparisons and began the "Evaluation Era" as stated by [1].

The huge growth of ASAG works in 2012 and 2013 (as seen in Figure 15) can be explained by two competitions that challenged participants to solve the short answer scoring problem. In 2012, the ASAP (Automated Student Assessment Prize) competition took place, organized by the online website Kaggle [1]. The top five winners released their codes and work methodology papers, which are included in this systematic review.

In 2013 another competition occurred, the SemEval '13 Task 7, the Joint Student Response Analysis and Eighth Recognizing Textual Entailment Challenge [94], releasing the Beetle and the SciEntsBank datasets. In total, three more public datasets were released from both competitions, which favored the next years (2013 to 2016) in researching new methods to solve ASAG since now there were six public datasets available that could be (and were, as seen in Section 3.4.6) used to evaluate new works.

As seen in the trend line of Figure 15, the trend is an overall increase in works that uses a machine learning approach on ASAG. The reasons, beyond the data availability, is that the techniques are more consolidated and it is a problem with a real world direct application.

3.4.2 RQ2: Nature of datasets

There is a great variety in the nature of datasets used by each reviewed paper. They vary in many aspects such as in the topic of the questions, language, student characteristics, grading scale, answers average size and the number of questions, answers and reference answers samples as can be seen in Table 8. In there, all of the different datasets

[1] http://www.kaggle.com/c/asap-sas

identified (28 out of the 44 studies) are presented. Some papers use more than one dataset. The ID column reports the id of the work that used this dataset, associated with their references in Table 7. The exceptions are ids from D1 to D6 that comprises the public datasets presented in the previous subsection (by the same order of appearance in the text) and better explained in Subsection 3.4.6. A X represents not reported information. Languages are represented in the table by their two letter ISO 639-1 representation due to available space.

Table 8 – Datasets' attributes

ID	Topic	Lang.	Educ. Level	NoQ	NoApQ	NoRA	Grading Scale	Resp. Length
1	Physics	EN	University	1	126	CPs	1-6 matches	48 W
2	Biology	EN	16 years	9	200	X	0 - n, n: 1-4	*Nominal
3	Physics	EN	X	1	293	CPs	1-16 matches	X
4	Science	EN	Grades 3-6	290	53	1	5 classes	*Nominal
5	Science	ZH	HighSchool	4	226	CPs	0 - n, n: 28-30	X
6	Science	ZH	High School	1	391	X	0 to 10	X
7	R. C.	EN	Grades 7-8	18	76	CPs	0 - n, n: 2-3	X
8	Formal Languages	EN	University	9	38	1	2 classes	X
17	Scientific Inquiry	EN	Middle School	2	152	0	0 to 4	48.5 / 62.4 W
19	R. C.	EN	Grades 6-9	2	1348	1	1 to 5	4 S
28	Science	EN	10 years	20	67	1	3 classes	*Nominal
29	Philosophy	AR	X	50	12	X	0 to 10	2.5 S / 24 W
30	Biology	EN	University	86	2200	1	2 classes	X
33	R. C.	EN	Grades 6-9	4	2000	1+, CPs	0 to 4	1,3 / 6 S
34	Scientific Inquiry	EN	University	33	35	1	1 to 6	X
35	Introductory C. S.	JA	University	15	123	X	5 classes	X
38	US Citizenship Test	EN	X	10	486	X	2 classes	4 T
39	Physics	EN	University	482	34	1+	2 classes	7.6 W
40	Science	EN	University	61	10	X	0 to 5	X
40	Science	AR	University	61	10	X	0 to 5	X
42	R. C.	HI	12 years	14	58	1+	0 to 5	X
43	Scientific Inquiry	EN	Middle School	8	500	1+	0 to 5	X
D1	Introductory C. S.	EN	University	80	28	X	0 to 5	X
D2	R. C.	EN	ESL	75	8	X	2 / 5 classes	1,5 S
D3	R. C.	DE	GSL	177	6	1,3 avg	2 classes	*Nominal
D4	Interdisciplinary	EN	Grade 10	10	2295	0	0 - n, n: 2 - 3	50 W
D5	Electronics	EN	High School	47	109	1	2 / 3 / 5 classes	X
D6	Science	EN	Grades 3-6	135	80	1	2 / 3 / 5 classes	X

Some terms used in the table are contracted due to available space, with their definitions following. **NoQ**: Number of Questions. **NoApQ**: Number of Answers per Question. **NoRA**: Number of Reference Answers. **R. C.**: Reading Comprehension. **C. S.**: Computer Science. **ESL**: English as Second Language. **GSL**: German as Second Language. **CPs**: Concepts. ***Nominal**: Some authors defined the response's length in nominal forms like "from short verb phrases to several sentences", "from a couple of words to several sentences", "up to around five lines" and "one to few sentences". **S**: Sentences. **W**: Words. **T**: Tokens.

Science related questions are the most common (57%) topic in the studies (Figure 16). Some studies only report generic science whereas some specify like Scientific Inquiry, Biology, Physics and Electronics. Another greatly used kind of questions are the reading comprehension type, present in 21% of the datasets. Computer Science related topics are also present (11%) in some works dealing with programming basic concepts, introductory

and formal language content. Other topics comprise of Philosophy, US citizen test and interdisciplinary content.

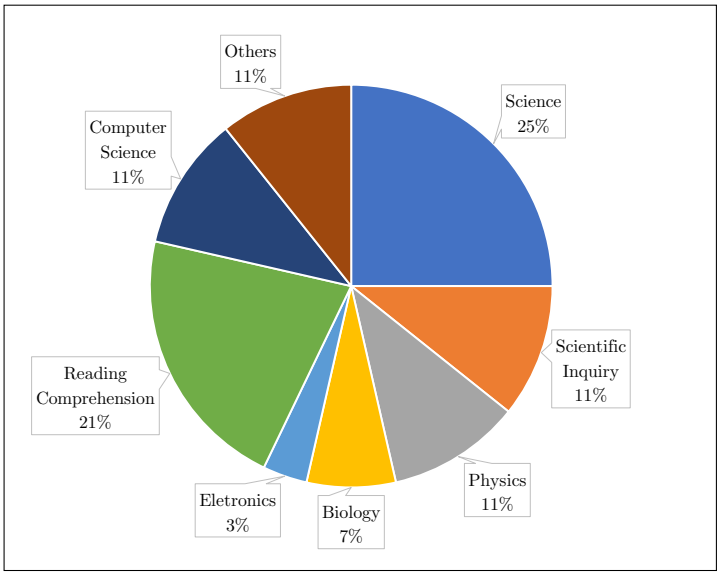


Figure 16 – Topics

Concerning the language of the datasets, 75% of them are in English. The other 25% are distributed between Chinese and Arabic (2 datasets each) and Japanese, Hindi and German (1 dataset each).

The respondents' educational level is reported in 89% of the papers. From those, 56% are in school as some report being in "middle school", "high school", "grade x to y" or the students' age. The other large group (36%) is in college, usually without specified age or year. Two works also deals with Second (Foreign) Language Studies (in English and German).

The graphic in Figure 17 reports the numbers of answers per question used among the 28 datasets.

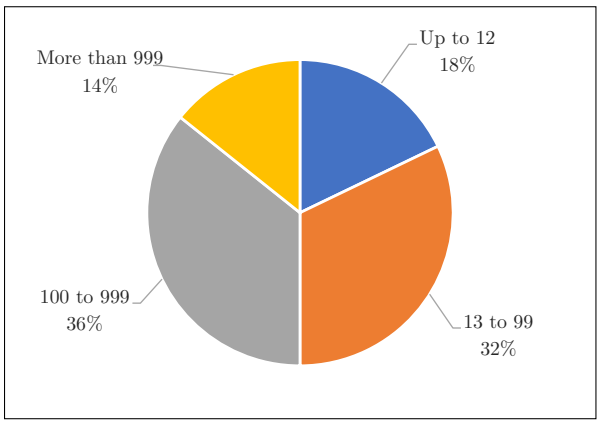


Figure 17 – Number of answers per question

The number of reference answers is not reported in one third of the works. Most studies used 1 (sometimes more) reference answers for comparisons with the students' ones. Few papers describe the use of concepts instead of reference answers itself.

The grading scale is presented in Table 8 in three possible formats: number of matches, a range of points or the number of classes. Some datasets have more than one grading scale for the same questions. Two, three, four and five points or classes correspond to the majority of works. Some have a 10 point scale and one isolated work has a 30 point range scale.

Only half the studies reports the responses' length. They are presented in terms of average number of words, sentences, tokens, lines or a written estimate. The number of sentences varies from 1 to 7 and the number of words from 7 to 63 in average.

Examples of questions, answers and grades from public datasets can be seen in Table 9.

Table 9 – Examples from public datasets

Id Ref	Question/Prompt	Answer	Scale	Grade
D1	What is a variable?	*A variable is a location in the computer's memory, in which a value can be stored and later can retrieve that value.*	0 to 5	3
D2	How is violence portrayed in cartoons according to the article?	*The bad guys do not usually win. Harmful or threatening characters usually tend to lose in the end.*	6 cl.	MC
D3	Ein Freund von dir möchte sich die alte Kamelienpflanze ansehen. Wann sollte er nach Pillnitz gehen und warum gerade in dieser Zeit?	*Er sollte Mitte Februar bis April gehen, weil die alte Kamelienpflanze zehntausende karminrote Blüten trägt.*	5 cl.	correct
D4	List and describe three processes used by cells to control the movement of substances across the cell membrane.	*1. Cells can use Passive Transports, which is where there is no energy used. 2. Cells can use Active Transports, which is where there is energy used.*	0 to 3	2
D5	What is voltage?	*the differences in the electrical states of the positive and negative terminals of a battery*	5 cl.	correct
D6	Carrie wanted to find out which was harder, a penny or a nickel, so she did a scratch test. How would this tell her which is harder?	*You could scratch the penny against the nickel and the nickel against the penny. If one scratches the other, it is harder.*	5 cl.	correct

Scales with "cl" stand for categorical classes. MC stands for "Missing Content".

3.4.3 RQ3: Natural Language Processing/Preprocessing Techniques

In order to model answers in easier ways for the computer to interpret them, some Natural Language Processing (NLP) techniques can be used. A reasonable number of different techniques are used in the reviewed studies to perform the preprocessing.

Not all works describe using NLP in the preprocessing step and we can assume that either they did not use these techniques or considered them not sufficiently relevant to report. We found the use of 19 different techniques among the 44 works. The bar graphic in Figure 18 shows the techniques ordered by the number of studies that used them.

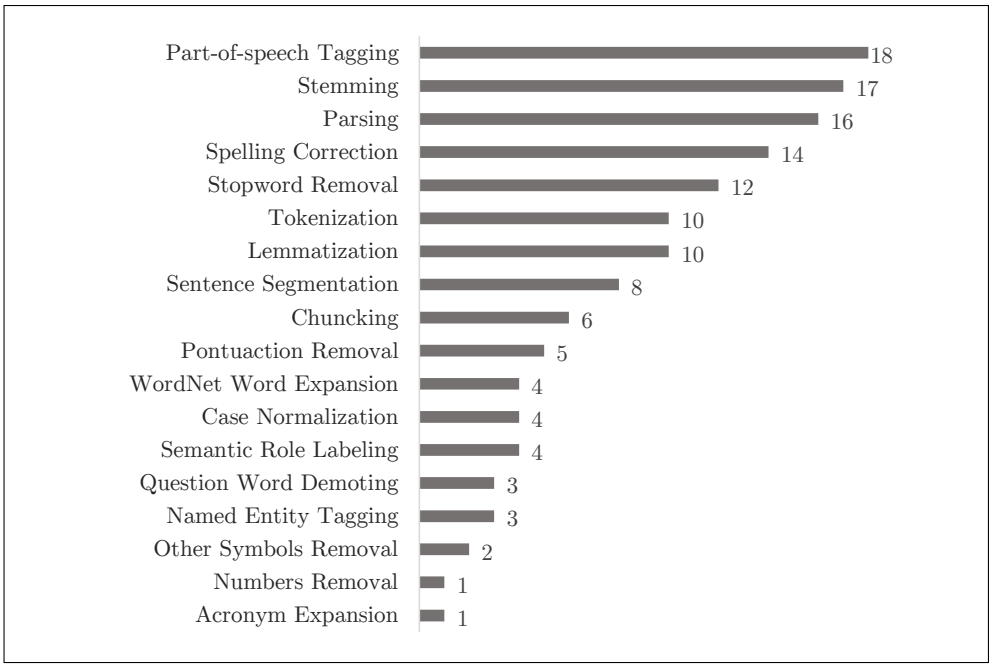


Figure 18 – NLP/Preprocessing techniques

3.4.4 RQ4: Features

In order to solve some problem using machine learning, some variables of the samples in the data must be identified in order to explain and predict the output. In ASAG, a large number of different features have been used in the literature to achieve good results. They can also be classified in different ways.

Maybe the *higher level* division that can be made is the one explicit defined in [9]: features can be response-based or reference-based. The **Response-based** approach extracts features only from the student answer itself (e.g. ngrams, etc). With regard to the **Reference-based** approach, it compares the student answer to an expected reference answer, using many different ways for measuring their similarity.

One way of classifying different similarity approaches is the following [95]:

- **String-based**: it measures how two sequences of characters or words are similar

based on the real composition of the words, analyzing their lexical similarity;

- **Corpus-based**: it is based on algorithms that collect statistics from large corpora and creates a model that can represent semantic associations;
- **Knowledge-based**: it is another way for measuring semantic similarity, but unlike corpus-based approach it is not based on statistics from corpora. Actually, it is based on semantic networks that models relationships between words, usually hand-crafted by human experts.

More details from similarity measures are presented in the following subsections, that are divided based on the natural language processing four categories: Lexical, Syntactical, Semantic, Discourse and Text Statistics. Each subsequent subsection will explore each one of them and present some of the most used and representative features.

3.4.4.1 Lexical and Text Statistics

In the lexical level, features consider only words by themselves. The most common model used in the reviewed works is Ngrams, as discussed in the previous chapter in Subsection 2.2.5. In the literature the n varies from one to six and the ngrams are made of letters or words. The ngrams model is used in more than 70% of the works. It is considered as a baseline feature. A special case of the word ngrams is when n equals 1, in which case is commonly known as a Bag-of-Words (BoW) model. BoW only takes in consideration the words and their frequency in some text, disregarding word order. This fact makes some authors ([56], [75]) alert not to take only BoW in consideration when handling ASAG.

Once the student's and teacher's answers are modeled using ngrams (after preprocessing procedures discussed in 3.4.3), the presence or absence of the important words is what matters. In case of $n > 1$, the presence of subsequent words or characters are also taken in consideration. This makes it easier to compare the answers between teachers and students.

Some studies used established metrics that have ngrams underneath like BLEU [96] and ROUGE [97]. Longest common substring is also considered as feature because it can indicate the longest ngram between the student and the reference answer.

Another way to extract features in the lexical level is to compare the student answer with a reference answer and assign a similarity score. The similarity measures based on the lexical level are called string-based because they are based on the raw content of the text, in the chain of characters and terms. When referring to lexical similarity, the *overlap* term can be used to indicate an intersection of characters or word between two text segments.

Techniques based on characters sequences are known as Longest Common Substring algorithm measures. Different metrics are available, such as Damerau-Levenshtein [98], Jaro-Winkler [99], among others.

The other possible way of measuring lexical similarity is to use a term (word) based technique. It works by comparing two strings and taking in account how much they share in terms of words. Some of the representative metrics used are City-Block (or Manhattan) Distance, Cosine Similarity, Dice's coefficient, Euclidean Distance, Jaccard similarity, among others.

Other greatly used features are text statistics like response's length, count of words, count of unique words, count of spelling errors, verb counts, number of characters, sentences, word average length and similar.

3.4.4.2 Syntactical

Despite the fact that ngrams with $n > 1$ can represent some part of word order, they do not model syntactical characteristics. There are, however, phrase and dependency ngrams that can model some syntactic meaning [60]. Phrase ngrams are the combination of the main verb and their noun phrase.

Syntactic ngrams are made of dependency relations where groups of ngrams words have syntactic connections. These dependencies can be obtained from a natural language parser like Stanford Parser [2]. The usual format is a triple containing two words and their relation dependency. These triples are used as features in 23% of the reviewed works. Other derived feature is to use the dependency path distance between words, as done in [60].

Another important syntactic feature is the similarity between student's and reference's answer PoS Tags. The part-of-speech represent the word's class and what is the behavior of that group in syntactic terms. Therefore, if two answers share many PoS tags they have a similar structure and are more likely to be meaning the same.

3.4.4.3 Semantic

The main goal of ASAG is to decide if the student's answer addresses a specific meaning, an associated semantic. One way of detecting desired meanings is to use Semantic Role Labeling (SRL), a natural language processing task that identifies predicate-argument relationships. The SRL detects semantic relations between words that are not necessarily syntactic related. As pointed in [56], SLR is generally only efficient in well-crafted sentences, but that is not a problem since a well-crafted sentence indicates a well-written answer in general. SRL is used in 9% of the reviewed studies.

[2] nlp.stanford.edu/software/lex-parser.shtml

Another greatly used approach is composed of knowledge-based features. Present in 25% of the reviewed studies, it is used to calculate similarity between words using a knowledge source. The most used source of knowledge similarity is WordNet [24], as discussed in the previous chapter in Subsection 2.2.3. Some similarities measures that can be used in WordNet are Leacock & Chodorow [100], Wu & Palmer [101], Lin [102], Resnik [103], Jiang & Conrath [104] and Shortest Path.

A third group of semantic features is formed by Textual Entailment (TE). TE consists of judging if one text can be inferred by another text. Some of the reviewed works interpret the ASAG problem as a textual entailment recognition problem [69]. Therefore, the use of entailment features is well justified and implemented in two works explicitly. One of them is the study of [64] that uses a TE recognition engine: BIUTEE. This tool tries to convert one text to another by applying a series of transformations. This is used in ASAG by making the student answers the test instance and the reference answer the hypothesis. As output, BIUTEE will return numerical entailment confidence values that are used as features. In [81], the EDITS system is used to generate the features. This system is an open source package for recognizing textual entailment in an adaptable environment for working with many different datasets.

The last group of semantic information features is composed of corpus-based similarity measures. Corpus-based measures uses large corpus to obtain statistical information that can later be used to calculate a relation value between words and documents. Three different similarity measures were identified in the reviewed works: Latent Semantic Analysis (LSA) [105], Explicit Semantic Analysis (ESA) [106] and "Extracting DIStributionally similar words using CO-occurrences" (DISCO) [107]. Each one of these three techniques are briefly described below:

- LSA: it is a technique that analyses relationships between documents and their terms, obtaining concepts that relate terms to documents. The idea of LSA is that similar words (at semantic level) will appear in different but similar texts. It works by constructing a matrix containing rows representing unique words and columns representing paragraphs or some unit of text [38]. A typical way of weighting the matrix is with tf-idf. After the construction of the matrix, a mathematical method named singular value decomposition is used to reduce the matrix dimensionality. Finally, words or the text are represented by a vector in what is called latent semantic space. This technique is used by eight of the reviewed works;

- ESA: it follows the same principles of LSA. They differ from one another mainly in two aspects. Firstly, ESA uses a large corpus (typically Wikipedia, as in the original paper that proposed the technique) instead of LSA where a set of any kind of documents can be used. Secondly, in ESA there is no dimensionality reduction and

then it is possible to assign human-readable labels to the concepts of the semantic vector space [106]. In the reviewed papers, three works use ESA as features: [75], [64] and [80];

- DISCO: following the principle of the distributional representation of words from LSA and ESA, DISCO also measures the similarity between words by the assumption that similar words occur in similar contexts. The technique is based on scanning the corpus using a context window of typically ±3 words for counting co-occurrences and construct a matrix where rows are made of unique words and a unique combination of (word, position) pair in the window is the column [80]. The matrix is then built by filling the values with the co-occurrence of words in the row and in the column. In the reviewed papers, two works uses DISCO as features: [89] and [80].

Finally, another recent approach that can be considered a corpus-based one are word embeddings techniques such as Word2Vec [40] and GloVe [108] as they rely on a great amount of text data to be trained, as seen in Section 2.4. As they can represent words in a semantic space, they are useful for ASAG because they create numerical representations that can be easily computed, but yet holding semantic meaning.

3.4.4.4 Discourse

The discourse analysis is the last step in a natural language processing analysis chain. As this stage is not much of interest to ASAG (because is related to longer texts) only two studies uses discourse features. In [61], a feature named Coherence counts the discourse connectors in an answer. In [54], five discourse features are used: count of identified discourse units, length of any identified list, the highest number associated with a numbered bullet, the number of discourse units headed by markers of conclusion and the number of those headed by "more information" type of connector.

3.4.5 RQ5: Machine Learning Methods

In ASAG, machine learning is used to solve a classification or regression problem. The model is built upon the answers of students and their correspondent grades assigned by a teacher. The goal is to predict which score should be assigned to a new answer.

Using the first division made in Section 2.3, we identified that only one work uses an unsupervised approach, using K-Means to cluster students answers. All other works uses supervised learning algorithms for classification or regression.

Another possible way to divide the ML methods is between "base" algorithms and ensemble techniques. A numerical analysis of the specific machine learning approaches used in the works can be seen in Figure 19.

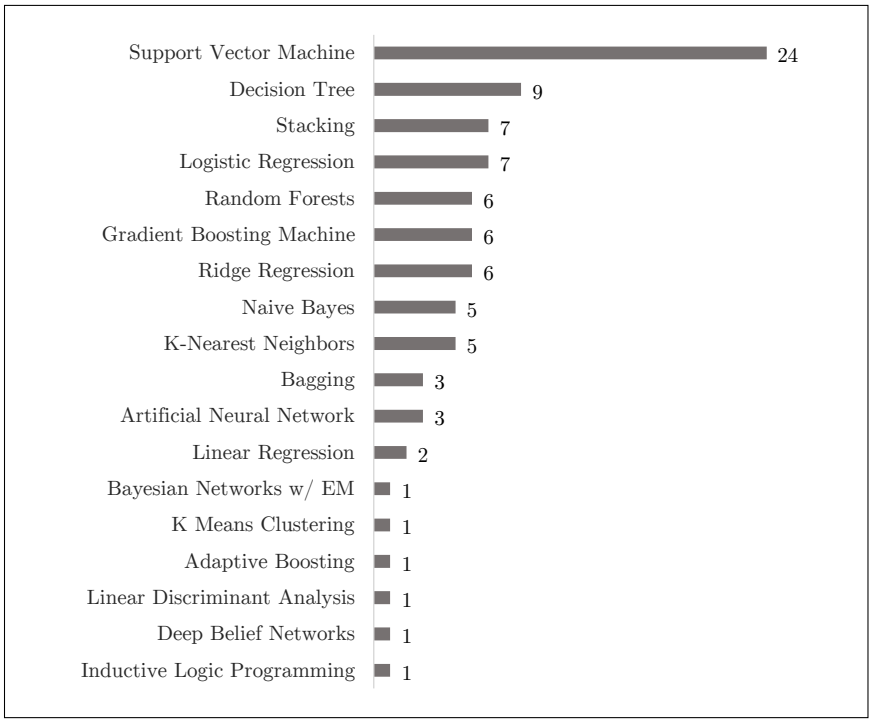


Figure 19 – Machine learning approaches

Considering techniques used by more than one work, the base algorithms found are (with the number of works that employed in parenthesis): Support Vector Machine (24), Decision Tree (9), Logistic Regression (7), Ridge Regression (6), Naive Bayes (5), K-Nearest Neighbors (5), Linear Regression (2) and Artificial Neural Networks (2). Concerning the ensemble approach, works used: Stacked Generalization (7), Random Forests (6), Gradient Boosting Machine (6) and Bagging (3).

The numbers from Figure 19 can be explained by the proven efficiency of the algorithms in [109], where the authors performed a comprehensive comparison of classification algorithms. From their *top-5* best performers, four are included in Figure 19, being: Support Vector Machine, Decision Tree, Random Forests and Gradient Boosting Machine (the exception being a more recent algorithm, Extreme Learning Machine). The other greatly used approaches from Figure 19 are classical (older) algorithms like Logistic Regression, Ridge Regression, Naive Bayes and K-Nearest Neighbors.

3.4.6 RQ6: Systems' Evaluation

This research question deals with the evaluation of the proposed systems. Each of the 44 reviewed papers describing a specific ASAG methodology was evaluated by the authors on the datasets presented in Subsection 3.4.2. Some studies evaluates in only one dataset whilst others report experiments in more than one. Another major difference among papers is the use of private or public datasets.

In this subsection, the different results achieved by the reviewed studies are pre-

sented. They are grouped by dataset, with one table made for each dataset [3] and one for the private datasets together to better compare the results. Even though the dataset grouping helps comparing, the different metrics employed and different number of classes can sometimes preclude a fair comparison.

Each table is composed by six columns: ID (the ID of the paper presented in Table 7, with chronological order), HHA Metric (the human-human agreement chosen metric), HHA Score (the human-human agreement correspondent score), SHA Metric (the system-human agreement metric), SHA Score (the system-human correspondent score) and the number of classes (in the format defined in Subsection 3.4.2). A X in any box indicates a not reported value. A score value of 1 indicates that the disagreement (if present before) was removed to perform the experiments.

It is important to state that the results presented in this section are only those from the selected papers for this systematic review. Other papers in the literature can have better results and the tables presented here should not be considered necessarily the state-of-the-art of the correspondent datasets. Also, results were extracted following some rules:

- If more than one method was used, the best one was picked;
- If the paper was evaluated in more than one dataset, one result was taken for each;
- If the results are reported by question, the mean of the questions is taken;
- If present, until the fourth decimal place was taken;
- Specifically in the two datasets from the SemEval '13, results from the UA (Unseen Answers) and 5-way task type were taken (for more details see [59]);
- In works that have more than one paper of the same or very similar system, the best one was chosen.

Following chronological order, in Table 10 the results on the CREE and CREG datasets from 2011 are presented. The ID 10 [78] is referencing the original work that presented the CREE dataset and ID 11 [93] the original CREG dataset. Besides the papers where these datasets were presented, only one more work [84] was found using CREG and none using CREE. The results showed that [84] achieved a very close accuracy from the original work, but did not get better results.

Still in 2011, the Texas dataset (as named by [1]) was released by [75], that was actually an upgrade of the [110] previous work. The papers 40 and 42 got two entries each

[3] Except CREE and CREG that are together due to the small number of uses and similarities between them.

Table 10 – CREE and CREG datasets

ID_Num	HHA Metric	HHA Score	SHA Metric	SHA Score	Classes
10	Agreement	1	Accuracy	0,884	2 classes
11	Agreement	1	Accuracy	0,846	2 classes
27	Agreement	1	Accuracy	0,844	2 classes

on the table because they evaluate their systems on both versions of the Texas dataset. Also, subsets of the dataset were used in some works and thus having different HHA scores. Due to those differences it is also hard to compare the results among different papers. Despite that, the table shows that some development have been made in the Texas dataset since its release.

Table 11 – Texas dataset

ID_Num	HHA Metric	HHA Score	SHA Metric	SHA Score	Classes
9	Agreement	1	Pearson's	0,518	0 to 5
37	Agreement	0,577	Agreement	0,61	0 to 5
40	Pearson's	0,644	Pearson's	0,59	0 to 5
40	Pearson's	0,586	Pearson's	0,55	0 to 5
41	Agreement	1	Pearson's	0,63	0 to 5
42	X	X	M.A.E.	0,67	0 to 5
42	X	X	M.A.E.	0,82	0 to 5
44	Pearson's	0,586	Pearson's	0,564	0 to 5

In 2012 ASAG research jumps into a new level when the ASAP dataset used in the Kaggle competition was released. The top five works (IDs 12 to 16) released their code and methodologies papers and have their results shown in Table 12. In addition to these papers, three more works (31, 37 and 38) were found using the ASAP dataset in their system's evaluation. In this case, the results are completely comparable (using quadratically weighted kappa). The winner of the Kaggle competition was [86]. Acknowledging this, [54] and [71] reported their results especially comparing with Tandalla's performance.

Table 12 – ASAP dataset

ID_Num	HHA Metric	HHA Score	SHA Metric	SHA Score	Classes
12	Agreement	1	Q. W. Kappa's	0,7711	0 - n, n: 2 - 3
13	Agreement	1	Q. W. Kappa's	0,7717	0 - n, n: 2 - 3
14	Agreement	1	Q. W. Kappa's	0,7575	0 - n, n: 2 - 3
15	Agreement	1	Q. W. Kappa's	0,7603	0 - n, n: 2 - 3
16	Agreement	1	Q. W. Kappa's	0,7653	0 - n, n: 2 - 3
31	Agreement	1	Q. W. Kappa's	0,768	0 - n, n: 2 - 3
37	Agreement	1	Q. W. Kappa's	0,78	0 - n, n: 2 - 3
38	Agreement	1	Q. W. Kappa's	0,67	0 - n, n: 2 - 3

The difference of 0,0037 of papers 13 and 31 is explained in Higgins[54] by three reasons. Firstly, in ASAP's competition, the learning parameters of the models could

be changed, features could be added or removed and other variants could optimize the results and submit these optimizations twice a day in a two months period. On the other hand, Higgins did not perform any optimizations. Secondly, Higgins also did not make any optimizations to individual questions as most ASAP's participants did. Finally, Higgins did not use any special rounding score method and simply rounded to the nearest integer.

Tandalla's approach involved manually crafted regular expressions to match simple patterns, made specifically per question. The goal of Ramachandran[71] was to compare their system-generated patterns with Tandalla's manual ones. Results showed that Ramachandran's system performed better than Tandalla's in eight out of the 10 questions. The mean was also greater by 0,0053, a larger difference than between Tandalla's and the second place team in Kaggle competition. Ramachandran also gives explanations for the two questions' worse performances.

In 2013, the SemEval '13 Task 7 competition took place as the Joint Student Response Analysis and Eighth Recognizing Textual Entailment Challenge [94]. In the competition, two new public datasets were released, Beetle and SciEntsBank. In Table 13 the results achieved in the Beetle dataset (composed by electronic answers) are presented. Papers from ID 20 to 26 are from the competition itself and only one more work was found evaluating in the Beetle dataset. The results are completely comparable.

Table 13 – Beetle dataset

ID_Num	HHA Metric	HHA Score	SHA Metric	SHA Score	Classes
20	X	X	Macro-Average F1	0,423	5 classes
21	X	X	Macro-Average F1	0,619	5 classes
22	X	X	Macro-Average F1	0,455	5 classes
23	X	X	Macro-Average F1	0,431	5 classes
24	X	X	Macro-Average F1	0,327	5 classes
25	X	X	Macro-Average F1	0,569	5 classes
26	X	X	Macro-Average F1	0,315	5 classes
32	X	X	Macro-Average F1	0,566	5 classes

The other dataset released in the SemEval '13 Task 7 competition is the SciEntsBank, composed of different science domains. Like in the Beetle's table, in Table 14 papers between 20 and 26 participated in the original competition and in this case, four works from outside the competition. In this case, almost all works are comparable, with the exception of ID 32 that is using another metric. The results shows the great development performed in the dataset since its beginning.

Table 14 – SciEntsBank dataset

ID_Num	HHA Metric	HHA Score	SHA Metric	SHA Score	Classes
20	X	X	Weighted-Average F1	0,590	5 classes
21	X	X	Weighted-Average F1	0,625	5 classes
22	X	X	Weighted-Average F1	0,537	5 classes
23	X	X	Weighted-Average F1	0,266	5 classes
24	X	X	Weighted-Average F1	0,419	5 classes
25	X	X	Weighted-Average F1	0,598	5 classes
26	X	X	Weighted-Average F1	0,372	5 classes
32	X	X	Macro-Average F1	0,566	5 classes
40	X	X	Weighted-Average F1	0,470	5 classes
41	X	X	Weighted-Average F1	0,582	5 classes
42	X	X	Weighted-Average F1	0,672	5 classes

Finally, in Table 15 we have all other evaluations that were not performed in public datasets. Details of the dataset used in each result can be seen in Subsection 3.4.2. A great variety of metrics, number of classes and the nature of the datasets is present, making it impossible to perform direct comparisons. The table has an expository and compilation purpose, precisely to highlight the wide difference when it comes to ASAG.

Table 15 – Other datasets

ID_Num	HHA Metric	HHA Score	SHA Metric	SHA Score	Classes
1	Kappa's	>0,75	Precision	0,93	1 - 6 matches
2	X	X	Agreement	0,6797	0 - n, n: 1 - 4
3	X	X	F1-Score	0,4704	1 - 16 matches
3	X	X	F1-Score	0,4974	1 - 16 matches
4	Kappa's	0,724	Accuracy	0,755	5 classes
5	Pearson's	0,96	Pearson's	0,92	0 - n, n: 28 - 30
6	Pearson's	0,81	Pearson's	0,86	0 to 10
7	Q. W. Kappa's	0,81	Q. W. Kappa's	0,75	0 - n, n: 2 - 3
8	X	X	Precision	0,6528	2 classes
17	Agreement	1	Pearson's	0,58	0 to 4
17	Agreement	1	Pearson's	0,43	0 to 4
18	Kappa's	0,69	F1-Score	0,77	2 classes
18	Kappa's	0,728	F1-Score	0,66	2 classes
19	Agreement	1	Agreement	0,585	0 to 5
28	Kappa's	0,72	Accuracy	0,68	3 classes
29	Agreement	1	Pearson's	0,862	0 to 10
30	Kappa's	>0,81	Spearman's	0,927	2 classes
33	X	X	Q. W. Kappa's	0,7575	0 to 4
34	Pearson's	0,69	Pearson's	0,67	1 to 6
35	X	X	Accuracy	0,859	5 classes
36	X	X	Accuracy	0,864	9 classes
38	Q. W. Kappa's	0,86	Q. W. Kappa's	0,96	2 classes
39	X	X	Accuracy	0,85	2 classes
40	Pearson's	0,86	Pearson's	0,84	0 to 5
42	X	X	M.A.E.	0,88	0 to 5
43	Agreement	0,9	Pearson's	0,7975	1 to 5

3.5 Summary

First, the systematic review was planned, the protocol and research questions were defined and the inclusion, exclusion and quality criteria created. The final selection resulted in 44 papers and the six research questions were answered based on them.

The temporal distribution of studies revealed a stagnant state at first, but the scenario starts to change when six publicity available datasets were released in 2011, 2012 and 2013. Three of them came from different authors and three from two competitions, the ASAP's 2012 and SemEval's 2013 ones. Those datasets opened the evaluation era in ASAG, where different methodologies could now be directly compared.

Among the 44 papers, 28 different datasets were identified. A table gathering all datasets and their characteristics was created and eight aspects were analyzed. Datasets are usually in English, about science questions and from diversified age of respondents. The numbers of questions, answers and reference answers have large ranges of minimum and maximum values. The grading scale is usually from two to five classes and the responses length normally ranges from 7 to 63 words in average.

Many different NLP and preprocessing techniques are used among works and the most used ones are part-of-speech tagging, stemming, parsing, spelling correction and stopwords and other symbols removal.

The features extracted to model answers can be grouped in four categories, the same as those studied in natural language processing: lexical, syntactical, semantic and discourse. These four aspects of natural languages are combined in order to model text as close as possible as the concepts they represent and the worthy grade of each answer. The selected features were used with several machine learning algorithms, with their frequency number of uses reported.

Finally, results achieved by the reviewed studies were grouped according to their correspondent dataset. Results were analyzed in terms of different agreement metrics between humans and proposed systems. Works that used private datasets were grouped and exposed the variety of metrics, classes and score values between different studies.

3.6 Portuguese Related Works - Literature Review

After performing the systematic review, the lack of works using Portuguese data was noticed. In order to fill this gap, we performed a simplified literature review aiming at finding some Portuguese related ASAG works.

Firstly, the search string was created in a similar way of the English one:

("correção automática"OR "avaliação automática") AND (questão OR exercício OR resposta OR questões) AND (discursivas OR discursiva OR dissertativa OR aberta OR subjetiva)

This search string was used only in Google Scholar. The first 100 results were analyzed within the ten first results page and 16 works were downloaded and fully read. From those, related to the field, seven works fitted in the ASAG definition. They were not included in the previous presented systematic review as they were written in Portuguese. Also, most of them do not uses a machine learning approach.

The seven selected Portuguese researches have their datasets presented in Table 16. Works were published between 2012 and 2016. The topic addressed by the questions is reported by six of the seven researches and comprise of Geography, Biology, Reading Comprehension, Teleinformatics Engineering, Database Systems and Portuguese. Most respondents are from college, one work is done with High School students, one with basic education of young and adult people and one only reports the respondents as "people".

Table 16 – Portuguese datasets

Ref.	Pub. Year	Topic	Educ. Level	NoQ	NoApQ	NoRA	Grading Scale	Length
[111]	2012	Geography	EJA/Adults	5	3	1+	0 to 10 (C)	X
[112]	2012	Biology/Geography	University	2	180	0	0 to 6 (D)	53 words
[113]	2013	Reading Comprehesion	University	1	68	1+	{0, 1, 2}	X
[114]	2013	Teleinformatics Eng.	University	13	12	1	X	X
[115]	2013	Database Systems	University	31	549	1	0 to 10 (C)	X
[116]	2014	X	"People"	30	10	5	4 classes	X
[117]	2016	Portuguese/Geography	High School	3	25	1	0-10 (D)	24 words

Some terms used in the table are contracted due to available space. Their definitions follows below.
Ref.: Reference. **Pub. Year**: Publication Year. **NoApQ**: Number of Answers per Question. **NoRA**: Number of Reference Answers. **Educ. Level**: Educational Level. **EJA**: *Ensino de Jovens e Adultos* (Young and Adults Education). **Eng.**: Engineering. **C**: Continuous. **D**: Discrete.

The number of questions vary from 1 to 31, far less then researches from Table 8. The number of answers per question is also smaller, being from only 3 to 549. Most works deals with only one reference answer whereas two of them do not specify if it is one or more. The grading scale is composed from 3 to 10 classes and two works deals with continuous variables. The average length of answers is only reported in two out of the seven studies: 24 and 53 words.

Following, Table 17 shows a summary of the seven Portuguese researches. Preprocessing and NLP techniques are used as same as in the systematic review, comprising of stopwords removal, morphological reduction, parsing, synonyms expansion, POS tagging and name entity recognition.

Table 17 – Portuguese researches

Ref.	Techniques	Preprocessing/NLP	HHA Metric	HHA Score	SHA Metric	SHA Score	Scale
[111]	Fuzzy Logic	Stopwords Rmv. Morphological Red. Synonyms Exp.	X	X	Mean Error	0,38	0 to 10 (C)
[112]	LSA SVD	Stopwords Rmv. Stemming	X	X	Accuracy	0,8735	0 to 6 (D)
[113]	CFG	Tokenization	X	X	Accuracy	>0,9	{0, 1, 2}
[114]	Similarity Functions	Stopwords Rmv. Other Symbols Rmv. Stemming	X	X	X	X	X
[115]	Ngrams, Similarity Functions and MLR	Stopwords Rmv. Stemming	X	X	Std. Error	0,33	0 to 10 (C)
[115]	Ngrams, Similarity Functions and MLR	Stopwords Rmv. Stemming	X	X	Accuracy*	0,9215	0 to 10 (C)
[116]	Linguistc Rules	Parsing Morphological Red. Synonyms Exp.	X	X	Precision	0,9212	4 classes
[117]	LSA WordNet	Tokenization Named Entity Recg. POS Tagging Lemmatization Stopwords Rmv.	Pearson's	0,7	Accuracy	0,8158	0-10 (D)

Some terms used in the table are contracted due to available space. Most contractions were used before in text. Those who were not, definitions follows below.
SVD: Singular Value Decomposition. **CFG**: Context-Free Grammar. **MLR**: Multiple Linear Regression. **Rmv.**: Removal. **Red.**: Reduction. **Exp**: Expansion. **Recg.**: Recognition. **Std.**: Standard.
*: in this case, the authors considered classes to be in the same class if grades are up to 1.0 point of difference.

Some of the techniques explored by the works in Table 17 involves quite manually work (fuzzy rules, linguistic rules and context-free grammars (a set of syntactic rules)). The other researches uses LSA, Ngrams, Similarity functions and WordNet to grade the answers (more details can be found in the referenced works or further in this work). Only one of these works employed a machine algorithm to grade answers ([115], that used Multiple Linear Regression).

Only one work [117] reported the human-human agreement: 0,7, using Pearson's correlation coefficient. Results of system-human agreement are reported using accuracy, precision and error (for continuous variables). Only [114] does not report results for system-human agreement as their research focus on evaluating different similarity metrics by performing comparisons among themselves.

3.7 Review Update

The systematic review presented in this chapter covered research until 2016. This section will briefly present an update for 2017 and 2018. This was performed for both reviews: Portuguese and Systematic Review.

3.7.1 Portuguese Literature Review Update

The search for more works was performed in two ways: **1)** running the same query in Google Scholar and analyzing the first 100 results, exactly as done in Section 3.6 and **2)** by searching works that referenced the seven retrieved researches of Section 3.6.

This procedure resulted in two new researches: [118] and [119]. The work of [119] is a continuation from the 2012's paper [112] and it uses the same data for the experiments. The performed preprocessing was: tokenization, special characters removal, stopwords removal and stemming. Instead of using LSA and SVD as the 2012's work, they used linear regression and ngrams similarity as techniques. They reported the human-human agreement (using accuracy) as being 0.94 for the biology question and 0.85 for the geography question. Results achieved for system-human agreement (also using accuracy) are 0.82 for the biology question and 0.86 for the geography question.

The second research [118] is also a continuation of a previous work from 2016's paper [117] and it also uses the same data for the experiments. The preprocessing and techniques are also the same from the 2016's work. The authors did not present a comparison with their previous work and the work was cited only when stating that *"the processing was described in [117]"*. The most apparent difference are the reported results for the system-human agreement. In [118] the agreement is presented using Pearson's Correlation and it has an average value for the three questions of 0,809 in the best method.

3.7.2 Systematic Review Update

As a systematic procedure was already performed and presented earlier in this chapter, we opted for a simplified methodology for the update. Papers were retrieved by looking for works that cited the most comprehensive and updated ASAG review [1] or one of the six most recent works found in the systematic review (from 2016). This procedure led to 42 papers related to our research question. Using the same criteria as before, we reduced the number of papers to 15.

The goal of analyzing new works from 2017 and 2018 was to identify if new techniques and features reported better results than prior works. The objective was to identify the latest trend in the research area and check for patterns across new works. This is the reason why the results are not reported in the same manner as before. The results in this subsection are focused on highlighting the defined objective for this update.

The 15 new works from 2017 and 2018 are presented with their titles and references in Table 18.

Table 18 – Selected papers (references and titles)

Ref.	Title
[120]	A Comparison of Features for the Automatic Labeling of Student Answers to Open-ended Questions
[121]	Matching , Re-ranking and Scoring : Learning Textual Similarity by Incorporating Dependency Graph Alignment and Coverage Features
[122]	Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading
[123]	Creating Scoring Rubric from Representative Student Answers for Improved Short Answer Grading
[124]	Work Smart – Reducing Effort in Short-Answer Grading
[125]	Using Rule-Based Methods and Machine Learning for Short Answer Scoring
[126]	Evaluating Semantic Analysis Methods for Short Answer Grading Using Linear Regression
[127]	Automatic assessment of communication skill in non-conventional interview settings: a comparative study
[128]	ANN Based Evaluation of Student's Answers in E-tests
[129]	Sentence level or token level features for automatic short answer grading?: use both
[130]	A Multimodal Assessment Framework for Integrating Student Writing and Drawing in Elementary Science Learning
[131]	Automatic Short Answer Grading and Feedback Using Text Mining Methods
[132]	Human and Automated CEFR-based Grading of Short Answers
[133]	A Short Answer Grading System in Chinese by Support Vector Approach
[134]	Automatic Chinese Short Answer Grading with Deep Autoencoder

Features from bag-of-words and ngrams modeling are still very used in ASAG systems [120, 124, 125, 121, 131]. Ngrams features were also found to perform better than other features [121], including word embedding based ones [120].

A growing technique is the use of word embedding resources. In [120] the Wiki2vec[4] library is used to generate vectors for DBpedia entities, used in their work to generate features. In [123] the InferSent[5] library is used to produce sentence embeddings, providing some semantic representation for English sentences. The authors from [129] presented a new equation for generating sentence features from the embeddings of the text in the question, student answer and reference answer.

The machine learning algorithms used to perform classification and regression remains the same, being: Random Forests, Support Vector Machine, Decision Tree, Logistic Regression, Naive Bayes and Ridge Regression [120, 124, 125, 133, 129]. A single paper reported the use of a clustering algorithm, k-means, to group answers and give specific feedbacks to each group [131]. The greater difference from the results in Section 3.4 is that artificial neural network based algorithms are increasing their use [128], specially using deep learning.

[4] https://github.com/idio/wiki2vec
[5] https://github.com/facebookresearch/InferSent

Following the general trend in using deep learning in various natural language tasks, some authors propose to also use it in the ASAG context. The use of a Deep Autoencoder is proposed in [134] for the classification of answers. In [122] the authors introduced a novel method by cascading three neural building blocks: a Siamese Bidirectional Long Short-term Memory (a Recurrent Neural Network), a pooling layer based on earth-mover distance and a final regression layer. In [130], the authors presented a new method using a Convolutional Neural Network arguing that it is suitable for ASAG for some reasons: it accepts inputs of arbitrary length (the case for student answers), it considers the words' order in sentences, it was effective in recent applications and it does not require human engineering of features as it automatically learns relevant features from the text itself.

Although it is not yet a rule, the trend identified in [1] of an "Evaluation Era" is being confirmed: the majority (8/15) of works evaluates their system on one or more of the public datasets. From those, the most popular (6 uses) is the "Texas" or "Computer Science (CS)" dataset from [75]. The SciEntsBank dataset [94] is also very popular, with 5 uses among the works. The ASAP-SAS is used in 2 works and CREE, CREG and Beetle are used in only 1 (in a single work [124] that used all the 6 public datasets at once). Despite the use of public datasets, results are often incomparable due to the use of different metrics among works. Also, authors do not search exhaustively for all recent work that uses the same dataset and evaluation metric as theirs, resulting in incomplete literature comparisons.

Other differences from previous works consists in data from others domains and sometimes coupled with other assessment items. In [127], the data comes from job interviews testing communication skills. Alongside with short answers, they also automate the assessment of essays and videos. Similarly, in [130] the assessment of short answers is performed in conjunction with drawings. Lastly, [132] works with data for predicting the level of a student in the Common European Framework of Reference for Languages.

New ideas for feature engineering are also present in the works. [120] used a Semantic Annotator that tags specifics words with information in a wiki. From there, they extract information to be used as feature. In [121] the authors introduced an "approximate dependency subgraph aligment" approach. In [123], they explore a new concept of using a "reference answer" for each specific grade, instead of just using it for the correct/top grade. They obtain class-specific representatives answers by clustering, selecting and ranking the student answers.

Finally, in [122] the familiar concept of data augmentation in images is adapted to ASAG in order to increase the amount of available data. They perform a similar procedure already proposed in [67]: to use student answers awarded with the max grade as they were teacher reference answers, increasing the pairs (student answer, reference answer).

4 DATA COLLECTION AND ANALYSIS

This chapter begins by introducing the *Auto-Avaliador* CIR web system, developed to be used in ASAG contexts. Then, the following sections will go over the process of creating and analysing a new Portuguese ASAG dataset using the web system, the first to be publicity available and the data basis for this work.

4.1 The *Auto-Avaliador CIR* Web System

In order to help teachers and students in the problems of manual assessment of short answer activities, discussed in the Introduction, comes the *Auto-Avaliador Colaborativo e Inteligente de Respostas (CIR)* (Collaborative and Intelligent Automatic Evaluator of Answers). As seen in Chapter 3, automatically grading answers is not an easy task and results are still being improved by new researches and technologies. However, an ASAG system can already be put in production, specially in not too rigorous evaluations. For instance: a teacher might be interested in using the automatic grading for feedback activities purposes, without account these grades in the student's school report.

In this context, the *Auto-Avaliador CIR* was created, a web environment for the development of dynamics involving questions and answers in a learning context. Using it, teachers can sign up and start creating tests and questions. In its turn, students have access to the tests and questions created by their teacher (and by others as well) and can answer and submit their responses. Student's answers will then be available for teachers to be assessed, providing feedback to students, that will see their grades in the system.

The "collaborative" term from the *CIR* acronym is because teachers can collaborate between themselves and with the system, specially in two ways. Firstly, a teacher can add reference answers and concepts to their colleagues' questions, complementing and improving them. Moreover, teachers will be able to grade answers of any tests and questions from the system, even those not created by them, contributing more to the student's feedback.

The "intelligent" term is present because when the system possess enough answers for a specific question, it will be able to start automatically grading answers from this point forward. This is possible due to the application of NLP and ML techniques, discussed in the fundamentals and systematic review chapters. However, for the good operation of the automatic system, many answers are required, hence, the more teachers and students are using the system, the better it will work.

For more details, screenshots and operation of the *Auto-Avaliador* CIR web system and its development process, please refer to Appendix A (in Portuguese).

4.2 Data Collection

The data addressed in this work is originated from the *Auto-Avaliador CIR* web system. All the samples collected with the system (questions, answers and grades) were obtained using a statistical method called **Convenience Sampling** (also called Haphazard Sampling). This sampling method is used because the samples are obtained according to what is easier to get for the researcher and samples are collected until a certain desired amount is reached [135].

The system's first use came from five biology elementary school teachers from a Educational Professional Master class in the Pampa Federal University - Jaguarão/RS. Together, these teachers discussed and created one test with 15 questions in the *Auto-Avaliador CIR* system (the full question list can be seen in Annex A).

The subject matter addressed by the questions consists mainly of human body topics, mostly seen in the 8th grade of elementary school. Some examples are: *"Explique o mecanismo de inspiração e de expiração do ar no corpo humano"* (Explain the inspiration and expiration mechanism of the human body) and *"Quais são as diferenças entre veias e artérias?"* (What are the differences between veins and arteries?). For each question, between two and four reference answers were also created by the teachers, alongside with between three and six keywords.

The recorded exam was then applied to 326 elementary school students (8th and 9th grades, about 12-14 years old) and to 333 high school students (10th to 12th grades, about 14-17 years). Table 19 presents the cities, school names, category (public or private school), application category, class grade and number of students from the schools where the test was applied. The application was made with the supervision of the student's teachers and each student had to come up with its own answers to the questions.

Table 19 – Test application in schools

City	Category	Application	Name	Grade	Number of Students
Londrina-PR	Public	Transcribed	Antônio de Moraes Barros	9th	36
Londrina-PR	Private	Directly	Ateneu	9th	15
Londrina-PR	Public	Transcribed	Dario Vellozo	8th	6
Londrina-PR	Private	Directly	Dôminos	9th	18
Londrina-PR	Private	Directly	Educacional MAF	9th	16
Londrina-PR	Private	Directly	Educativa	9th	36
Londrina-PR	Private	Directly	Interativa	9th	35
Cambé-PR	Public	Transcribed	Maestro Andrea Nuzzi	8th	75
Jaguarão-RS	Public	Directly	Manoel Pereira Vargas	9th	12
Londrina-PR	Private	Directly	Universitario	9th	77
Londrina-PR	Private	Transcribed	Londrinense	10-12th	333

Some students answered directly in the web application but, in some schools with less conditions or difficulties to access computers, the application was made in paper and then transcribed (rigorously, including spelling errors, spaces, accentuation, *etc*) to the

system. Students were instructed to try their best, even if it involved guessing, in order to collect all sort of answers and grades.

In possession of the answers, 14 undergraduate biology students from the final year of college assessed the answers considering a predefined scale. Graders assigned one of four possible grades to each answer:

- **Zero:** when the answer is at least mostly wrong, out of scope or nonsense;
- **One:** if the answer has something correct but it is still mostly wrong or incomplete;
- **Two:** if the answer is correct but has some wrong detail or missing important content;
- **Three:** if the answer is mostly correct, with the important points presented.

4.3 Data Example

This section presents a complete example from collected data. Table 20 shows a question along with two possible reference answers and four concepts. In Table 21, graded student answers are shown, with two examples for each possible class.

Table 20 – Question, reference answers and concepts

Category	**Text**
Question	*Qual a diferença entre a célula animal e a célula vegetal?*
Reference Answer	*A célula animal e vegetal apresentam formato diferenciado. A célula animal possui formato irregular, enquanto a célula vegetal apresenta uma forma fixa.*
Reference Answer	*A parede celular é uma estrutura exclusiva das células vegetais. Ela corresponde a um envoltório externo à membrana plasmática.*
Concept	*membrana plasmática*
Concept	*celulose*
Concept	*formato irregular*
Concept	*forma fixa*

The example question asks about differences between animal and plant cells. One reference answer points the difference of shapes, being irregular or fixed. The other discuss the cell wall, in which plant cells have external to the plasma membrane. Concepts are composed of keywords as: plasma membrane, cellulose, irregular shape and fixed shape.

Student answers for the question addressed in Table 20 are presented in Table 21 ordered by the received grade.

The worst grade (0) was given to students that wrote incorrect facts about science. In these two examples, students compared animal and plant cells, but writing that differences consisted in “not dying” or “possessing DNA”, which is very incorrect, as both cell types possesses DNA and both dies.

Table 21 – Students answers

Text	Grade
As células animais morrem e as vegetais não.	0
Célula animal, possui DNA, e a célula vegetal não possui.	0
Célula animal mais complexa, a vegetal tem funções bem diferentes da célula animal.	1
A membrana plasmática é diferente. Acho eu.	1
As diferenças entre elas são a estrutura, formato, e componentes que constitui a célula!	2
A célula animal é formada por diferentes tipos de organelas das células vegetais.	2
As células animais são todas aquelas que compõem os seres vivos do reino animália, composto por membrana plasmática, citoplasma e núcleo verdadeiro. As células vegerais contem estruturas como parede celulares, plastídios e grandes vacúolos	3
Na célula animal temos membrana, citoplasma e núcleo na vegetal tem 3 a mais parede celular cloroplasma vácuo e na célula vegetal a respiração celular é a fotossíntese, que ocorre no cloroplasmo.	3

Students awarded with a 1 wrote something correct but not sufficient. The first one tells that the animal cell is more complex and have different functions from plant, which is true, but too shallow. The other only says "the plasma membrane is different" but without writing this difference specifically or saying which of animal or plant have it or not. This student also wrote "I guess", which demonstrates his lack of knowledge in the topic.

Students with answers graded as 2 wrote expected content but were not specific. They wrote about the differences correctly, but just in a general aspect, without being specific about it, which resulted in not receiving the full grade.

Finally, students with a grade of 3 wrote correct and complete answers, addressing the expected content. They were very clear, specific and complete about their answers. Moreover, the answers are clearly longer than the previous answers. By just glancing at the table is possible to notice that, at least in these examples, longer answers consisted in more correct answers, whilst shorter answers were incorrect. Also, the presence or not of the expected concepts is also noticeable, by considering the resulting grades.

4.4 Data Analysis

This section presents some considerations about the data collected, regarding quantities, labels distributions, statistical information and others. Also, in order to make the data fully and publicity available, it was published in Kaggle, a website designed for hosting datasets and data science competitions. This work's dataset is available at *https://www.kaggle.com/lucasbgalhardi/pt-asag-2018*.

4.4.1 Labels Distribution

The test containing 15 questions was applied to 659 students in total, so 9885 answers could be expected. However, students left some answers in blank. Additionally, in rare cases answers were completely equal, consisting in duplicated data, that was removed for all further analysis and experiments. The number of usable answers available was 7473, distributed between different grades as shown in Figure 20.

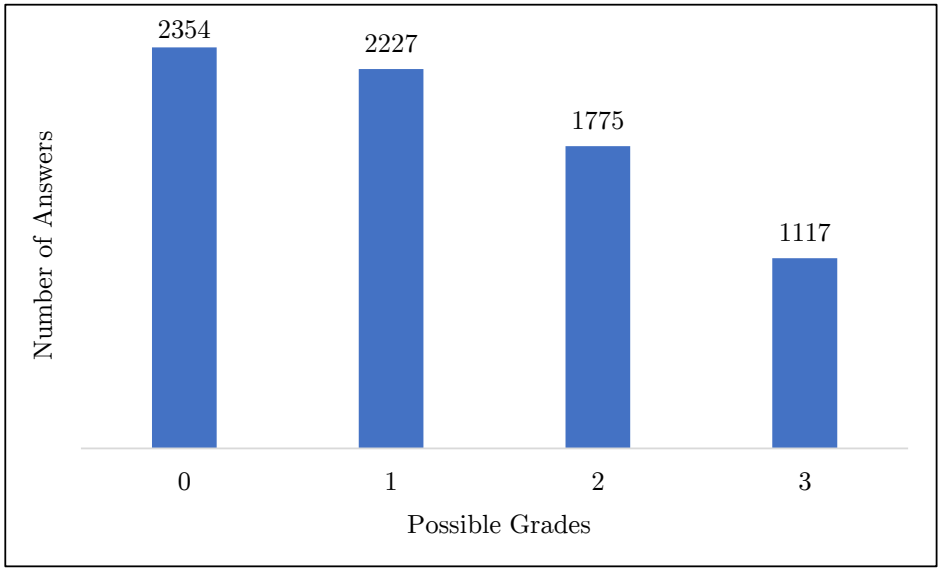


Figure 20 – Labels' distribution

The complete label distribution per question is presented in Table 22. Questions 3, 7, 8, 9 and 10 (highlighted in the table) are somewhat unbalanced and marked for further considerations in next chapters. The other questions are reasonably balanced, which is good for performing machine learning experiments. Each question has an average of 498 graded answers, ranging from 348 to 615.

Table 22 – Labels' distribution

Q_ID	0	1	2	3	Q_ID	0	1	2	3
1	173	159	182	101	**9**	276	63	42	41
2	100	240	213	44	**10**	234	46	45	23
3	144	320	51	14	**11**	187	126	199	60
4	99	100	137	72	**12**	159	94	101	122
5	134	124	114	85	**13**	114	159	118	131
6	149	179	134	60	**14**	43	117	205	191
7	312	148	22	5	**15**	104	70	113	145
8	126	282	99	23	**Sum**	2354	2227	1775	1117

4.4.2 Inter-rater Reliability

In the Introduction of this work, it was discussed that humans' subjectivity can have a great impact on grading. In this study, as in many others from ASAG (HHA metrics and scores in Subsection 3.4.6), we measured how humans agrees between themselves regarding the grade to be assigned to an answer (the inter-rater reliability or agreement).

From the 15 questions, four (1, 9, 11 and 12) had all the answers graded by more than one rater. The metric chosen to measure the agreement is weighted Cohen's Kappa, as it is one of the most common and used metric for performing this kind of evaluation in ASAG [11] and in other domains and applications as well [136].

The weighted kappa statistic was created to account for different levels of disagreement [137]. So, if one answer is given 1 by one grader and 2 by another, the disagreement is not as weighty as if the grades were 1 and 3 (in the first case there is a 1-point disagreement and in the second case there is a 2-point disagreement).

Weights assigned to the kappa statistic are mainly linear or quadratic. The difference between them is that in the linear scheme the disagreement from 0 to 1 and from 1 to 2 is equally weighted, which is not the case for the quadratic approach, where the higher the two different scores, higher the *penalty*. It is recommended to report both weighting approaches when possible [136].

Disagreements between raters in the four questions is very intense, as can be seen in Table 23. Important to state that graders were all presented with the same criteria for assigning grades, as shown in Section 4.2. In the table it is possible to notice that the higher the distance the less disagreements, as would be expected. However, graders strongly disagrees among themselves, specially in a 1-point difference, a number that is higher than completely agreements (distance zero) in two out of the four questions.

Table 23 – Disagreement between raters

Distance/ Kappa Score	**Q1**	**Q9**	**Q11**	**Q12**
0	256	260	264	206
1	281	94	266	170
2	75	64	37	87
3	3	4	5	13
Linear	0,4	0,43	0,39	0,37
Quadratic	0,57	0,54	0,52	0,5

Analysing the kappa scores according to the [32] guidelines, graders have between fair to moderate agreement (0.2 - 0.4 it is considered "fair" and between 0.4 - 0.6 it is considered "moderate"). Interesting to observe that scores are similar among questions.

4.4.3 Common Words and Bigrams - Word Cloud

One way of visualizing text data in an intuitive manner is by the use of a Word Cloud. A Word Cloud is a figure formed by many words in a specific shape, usually a cloud or a simple rectangle. It is a good way to get a glimpse at the most frequent words and bigrams within a corpus, as it shows words in a variety of font sizes and the more frequent a word or bigram is, bigger it gets represented.

In Figure 21 there is a Word Cloud made with the answers from the students of this work. To enable all of the questions to appear equally, 340 answers of each question were randomly selected to compose the corpus for visualization (otherwise questions with more answers would repress questions with less answers). Stopwords were removed, with the exception of the *no* word.

Figure 21 – Word Cloud

The bigger words are *corpo* (body), *célula* (cell), *não* (no) and *pois* (because). It is intuitive to notice that words are referring to human body related content. Also, words like “because” and “no” appears frequently due to the explanation nature of science questions. Other common words are compositions of noun phrases (bigrams, that are often also *collocations*) like *parede celular* (cell wall), *tecido epitelial* (epithelial tissue), *material genético* (genetic material) and similar. More correlations between the words in Figure 21 and the questions can be seen by analysing the question list (Annex A).

4.4.4 Statistics

In Table 24 some statistics extracted from each answer are presented. Fifteen answers were removed as they consisted in outliers (students that send music lyrics or food recipes). The column *Sum* is the simple sum of each statistic, except in *Unique Words* (gray marked) where the computation is made considering all words and not *by answer*.

Table 24 – Statistics per answer

X	Min	Max	Avg	Std Dev.	Sum
Sentences	1	6	1,6	0,8	11.924
Words	1	136	14,9	10,8	111.352
Characters	1	982	92,9	68,3	693.181
Uniques Words	0	57	12,8	8,11	7607
Commas	0	9	0,78	1,29	5826
Word Avg. Length	0	12	5,2	1,3	-
Words per Sentece Avg.	1	87,7	12,8	8,5	-

5 EXPERIMENTS, RESULTS AND DISCUSSION

This chapter presents the performed experiments, their methodological procedures and results. Section 5.1 discuss the general methodology used for all experiments. Sections 5.2 to 5.6 details into the specifics from each different experiment and its results. Then, Section 5.7 performs comparisons between approaches from previous sections and some possible combinations for them. Finally, Section 5.8 reports a comparison between the agreement among raters and the agreement between the best model and human grading.

Additionally, at the end of each section (from Sections 5.2 to 5.6), a table containing detailed results of the section is presented. However, they are reported only for informative and record purposes at first. A detailed discussion of these section's results is performed in Section 5.7.

5.1 General Methodology

This section presents the general methodology used for the experiments reported in the following sections of this chapter. A general flowchart, representing the experiments' steps and connections, is reported in Figure 22.

The experiments started with the early definition of the evaluation metric, preprocessing techniques, ML algorithms and implementation libraries (described from Subsection 5.1.1 to 5.1.4). Then, six different approaches were defined to be tested (described from Sections 5.2 to 5.6). These approaches (called groups) had their own separated experiments, with each group varying several aspects such as the preprocessing techniques, ML algorithms, techniques and other *internal parameters* (Figure 22).

The results obtained by each group are compared and discussed together in Section 5.7. In this section, a combination of the groups is also reported, performed in some different combinations. By testing four different manners of combining and five different top-k groups, 20 combinations are created. From those, the best performer is taken and used to compare to the Human Agreement, reported in Section 5.8 (Figure 22).

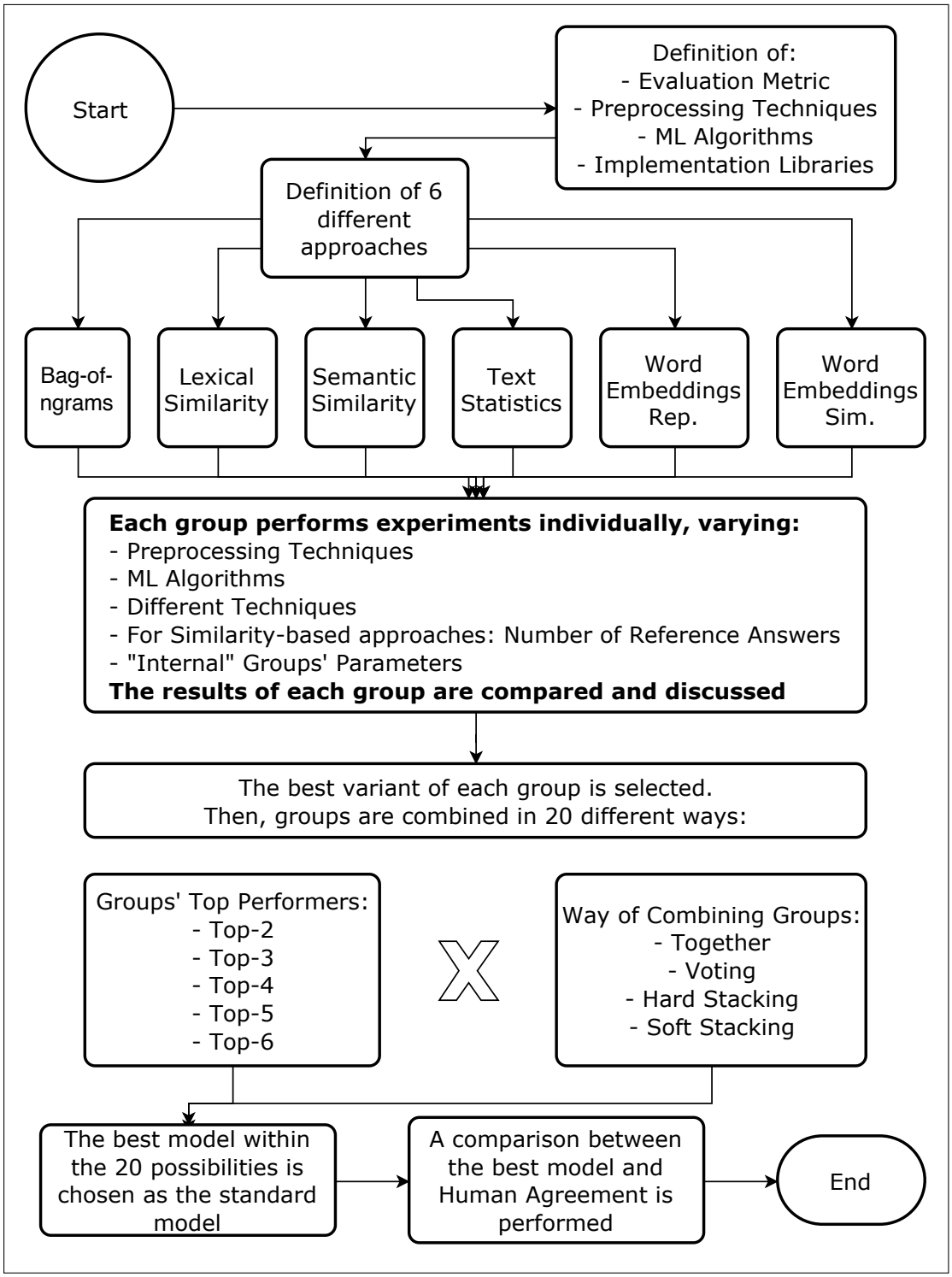


Figure 22 – Experiments' General Flowchart

5.1.1 Evaluation Metrics

In Section 2.3 it was introduced that there are some different metrics that can be used for evaluating machine learning models like: Confusion Matrix, Accuracy, Precision, Recall, F1, Cohen's Kappa, Pearson's and Spearman's Rank. These metrics are also seen in Subsection 3.4.6 that presents evaluations for all reviewed works.

While confusion matrix and its derived metrics (Accuracy, Precision, Recall and F1) gives a good parameter for measuring performance, it lacks some issues. Firstly, these metrics can be deceiving if the data is imbalanced [109], which is the case for some of the questions in this work. This could be remedy by a comparison with a dummy classifier (that assigns every sample with the most frequent class) [59], but still it is harder to visualize the real performance.

The Cohen's Kappa correlation coefficient can better show the performance, way more independently of the data imbalance than accuracy. Also, it is a good measure for accounting for chance agreement [11, 92]. Pearson's is also greatly used, but usually for cases when the measured variables are continuous (Subsection 3.4.6). Spearman's rank also have its advantages in specific cases, but it is not too used in ASAG research (Subsection 3.4.6).

Considering the aforementioned aspects of different metrics and the ability of weighted Cohen's Kappa to measure ordinal variables (Subsection 4.4.2), we opted for two metrics: Cohen's Kappa (with linear and quadratic weights, as explained in Subsection 4.4.2) and Accuracy (for easy and quickly interpretability). However, accuracy is used only for informative purposes. For the experiments reported in the following sections, only kappa is considered (the average between the scores produced by both weighting schemes is used for direct comparison, namely from here on *bk* value). When not specifically stated in the following sections, scores are referring to the averaged *bk* score among all questions (specially in the graphic visualizations: the vertical axis reports this performance metric, which was used for general purpose comparisons).

5.1.2 Preprocessing

Five text preprocessing techniques were considered in this work when performing the experiments:

- **Case normalization:** to not differentiate between upper and lowercase;
- **Non-alphanumeric characters removal:** as they do not add any value;
- **Accents removal:** to enhance matches between answers with and without accents;
- **Morphological reduction:** to make it easier to match words with only morphological differences. This can be accomplished with the use of lemmatization or stemming, algorithms that reduces words to their root or reduced form. This is an important technique for Portuguese as it is a language with rich morphology (for more details, see Subsection 2.2.1);
- **Stopwords removal:** used to remove very common words so that when measuring similarity they are not taken in consideration.

5.1.3 Machine Learning Algorithms

In Subsection 3.4.5 all the machine learning algorithms used in ASAG research were presented. From Figure 19, the top-6 (in terms of use in works) are, respectively: Support Vector Machine (SVM[1]), Decision Tree (DT), Stacking, Logistic Regression, Random Forests (RF) and Gradient Boosting Machine (GBM). DT is greatly used mainly because of its capacity for interpretability, a desired ability in many applications. Stacking is a combination of other approaches (more details in Section 5.7).

Moreover, SVM, DT, RF and GBM are in the top-5 performers in [109] exhaustive machine learning algorithm's comparison. In order to choose between algorithms, we selected the intersection from the top-performers in [109] with the top used in Subsection 3.4.5: Support Vector Machine, Random Forests and Gradient Boosting Machine. Decision Tree was left off as RF can be considered an improved version of DT. Stacking is used in Section 5.7 for experiments that combines different approaches. All base models uses default hyperparameters settings[2]. All experiments are performed using 5-fold cross validation [55, 88, 61, 57, 83]. The XGB (eXtreme Gradient Boosting) library is used in this work as the implementation of the Gradient Boosting Machine algorithm [138].

5.1.4 Implementation Libraries

The Python libraries used for implementing the techniques addressed in this chapter are grouped in Table 25.

Table 25 – Implementation libraries

Technique	**Library**	**Version**	**Reference**
Basic Preprocessing	Python	3.6.1	-
Stemming	NLTK	3.2.3	[139]
Stopwords Removal	NLTK	3.2.3	[139]
Lemmatization	Cogroo	4	[140] [141]
Ngrams Extraction	scikit-learn	0.18.1	[142]
Gradient Boosting	xgboost	0.7	[138]
Machine Learning	scikit-learn	0.18.1	[142]
Similarity Metrics	textdistance	3.0.3	[143]
Semantic Similarity	NLTK/WordNet	3.2.3	[139] [24]
Text Statistics	Python	3.6.1	-
WordEmbeddings	gensim/NILC	3.7	[3]

[1] With RBF (Radial Basis Function) Kernel

[2] Except for the n_estimators in Random Forest, changed to 100 to match with XGBoost and because its default value will also change to 100 in the next release version of the library.

[3] Gensim: radimrehurek.com/gensim/ - NILC: nilc.icmc.usp.br/embeddings.

5.2 Bag-of-ngrams

Ngrams is one of the most common ways to model language and a powerful predictor for ASAG [1, 80, 85, 67]. It is based on the idea that words' presence or absence can predict the desired output. Using ngrams for ASAG modeling means that the learning algorithm will base the patterns' searches among the words used by the students. It will attempt to find which of the words (or sequences of characters) used by students correspond to correct answers and which do not.

As ngrams works on the principle of presence or absence of text's pieces, it is a question-specific feature: important words for a question are not important to another. Hence, each question has its own bag-of-ngrams sparse matrix of features (a document-term matrix), where each document (student answer) is represented as a row and each ngram as a column. Each cell contains a weight, that can be defined in different ways (see Subsection 2.2.5).

For performing the experiments, three aspects were considered (beyond the three machine learning algorithms). Firstly, the preprocessing: case normalization and accents removal increases the chances that the same ngram, but with different case or accent, is mapped to the same column. Non-alphanumeric characters does not add any value. Regarding stopwords removal and morphological reduction, they would expect to increase the results, by removing too common words and disregarding their morphological variation. However, we put the theory into test and created six variations for these parameters, being the multiplication of $stopwords\_removal = \{True, False\} \times morphological\_reduction = \{None, Stemming, Lemmatization\}$.

Secondly, the ngrams extraction: there is a lot of variation in the literature. Two types of ngrams can be used: characters or words. Words' ngrams are more common, but characters' also have some advantages, such as being more robust regarding students' spelling errors [9]. In the literature, it was found that word ngrams always ranges its n between 1 and 3 [91, 9, 67] and so this value was fixed. However, for characters' ngrams this value varies [83, 73, 9] and the literature values were considered as base for defining and testing six different ranges $\{(2,4), (2,5), (3,6), (4,6), (4,7), (5,8)\}$.

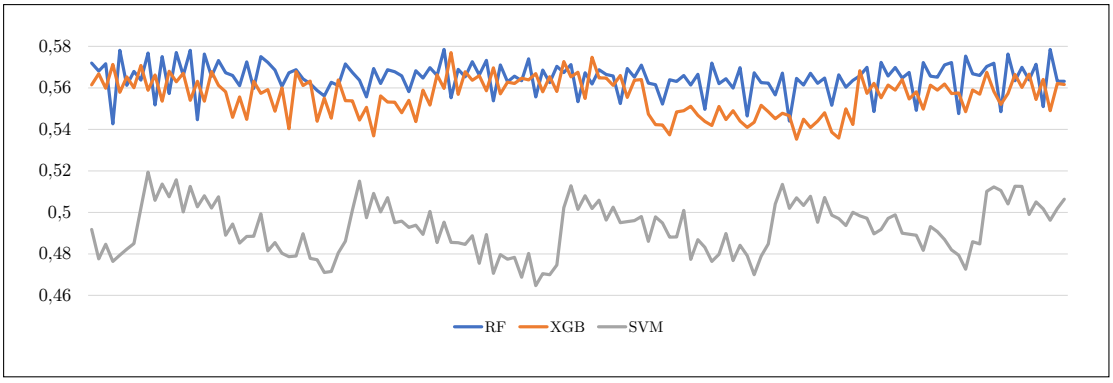
Another parameter that must be considered when extracting ngrams is the matrix size concerning the terms (columns). This modeling scheme usually creates very sparse matrices, that have lots of columns for terms that are only used once or few times. Therefore, a way for reducing the matrix dimensionality is to use a criteria (usually the term frequency) to cut off columns. This technique is widely used in ASAG and values for how much to keep are diversified [54, 91, 9]. For words' ngrams the max value was ranged between 250 and 650, at a hundred pace. For the characters' ngrams the max value for each range was defined according to the number that roughly indicates at least 30 for the

frequency value for kept terms (respectively: $\{500, 600, 550, 350, 450, 300\}$).

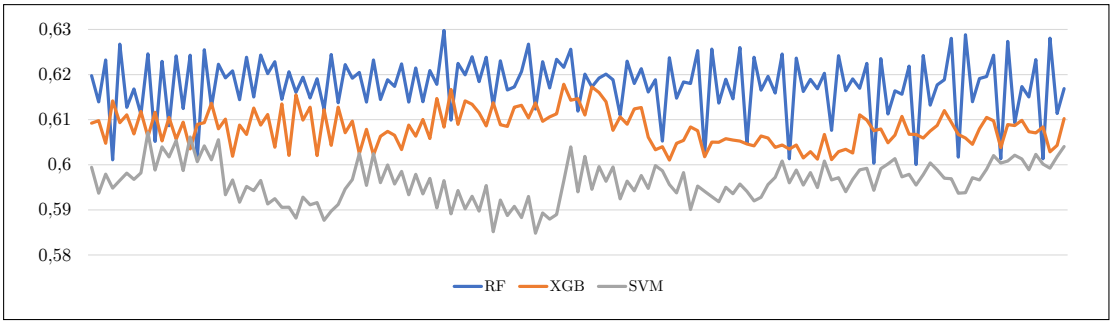
Finally, the weighting scheme for filling the cells was considered. The two most used in the literature were tested: term frequency and term frequency-inverse document frequency (tf-idf) [72, 86, 63, 91]. These aspects combined creates 5400 executions to test for the best parameter set (6 preprocessing values × 5 word ngrams' max value × 6 character ngrams' max value and range × 2 weighting schemes × 15 questions).

The first step in analysing the results is to average scores between questions to get a single score. This averaged score is then used for comparing the 360 (5400/15) possible combinations, each with three classifiers scores in three metrics. Then, the *bk* score is used to determine the higher score and its corresponding parameter set, that was found to be using stopwords removal, using lemmatization, Random Forest, word ngrams max features = 450, character ngrams max features = 500 and its range = (2,4).

Concerning the comparison between the three classifiers, Random Forest performed better, as shown in Figure 23 (that reports only some of the executions for better visualization). However, XGB also performed good, following RF's performance very closely. SVM got scores not that far way when considering accuracy (Figure 23b). Even so, there is a huge difference when comparing it with the others using the *bk* score (Figure 23a). That means that it achieves almost as much correct predictions as RF and XGB does, but when it gets the wrong prediction, the error is greater between classes.



(a) *BK* scores



(b) Accuracy scores

Figure 23 – Classifiers' performance

The preprocessing parameters (morphological reduction and the presence or not of stopwords) do not highly impact performance when considering the average between questions (differences were up to 0,01479 of *bk* value). Individually, by each question, their impact is somewhat larger. Nevertheless, even considering that it is a small difference, lemmatization still usually performs better than stemming and *none*, with the number of "wins" (when the technique outperformed the others) being 43%, 35% and 22% respectively. As for stopwords, its removal performs better than not removing in 64% of the cases (only 36% of executions performed better without removing stopwords).

Finally, Figure 24 shows the performance of some executions using Term Frequency versus the same executions using Term Frequency-Inverse Document Frequency. Although it is not by too much, it is clear that TF performed always equally or better than TF-IDF.

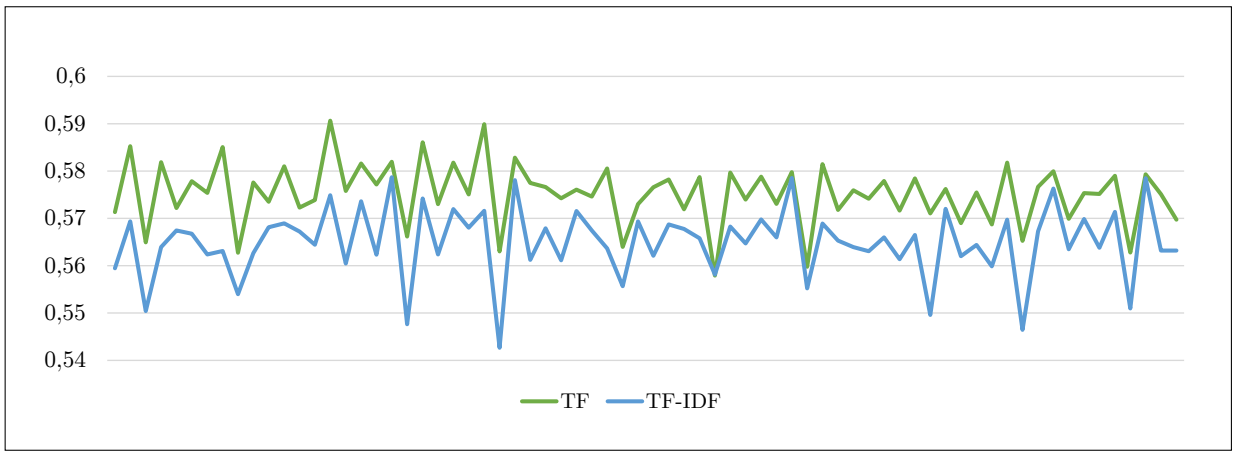


Figure 24 – TF vs TF-IDF

All things considered, Table 26 presents the results from the execution with the best parameter set for all the questions and all the metrics: Accuracy (Acc), Linear Kappa (LK) and Quadratic Kappa (QK).

Table 26 – Ngram results for all questions

Q_ID	Acc	LK	QK	Q_ID	Acc	LK	QK
1	0,527	0,535	0,688	**9**	0,682	0,430	0,538
2	0,660	0,584	0,706	**10**	0,819	0,772	0,868
3	0,749	0,536	0,614	**11**	0,552	0,466	0,567
4	0,684	0,715	0,836	**12**	0,506	0,520	0,663
5	0,650	0,637	0,733	**13**	0,418	0,320	0,404
6	0,525	0,433	0,542	**14**	0,624	0,576	0,702
7	0,665	0,318	0,410	**15**	0,711	0,754	0,862
8	0,625	0,439	0,552	**Mean**	0,626	0,536	0,646

5.3 Lexical Similarity

This set of features is based on the lexical level of language analysis, as discussed in Subsection 2.2.1 and Subsubsection 3.4.4.1. The features are extracted considering the similarity between students' and references' answers. This type of similarity is widely employed in ASAG research [1].

In the lexical level, similarity is given by how related words appears to be, based on their constituents (letters). That means that two words that are very similar in shape (or even equals: homonyms) can be get a high similarity score, even if the meaning is not related (e.g. cell and cell phone). This problem is considered in this work and covered by semantic features in the next section.

Although only considering the lexical level, there are different groups of metrics that can be considered to measure similarity. Some of these metrics are grouped in [95] survey in their *"String-Based Similarity"* section and used in this work. Other metrics not present in the survey are referenced in the below list (with 12 metrics), grouped by four different types:

1. **Token-based(3):** it measures similarity between two strings by considering the intersection of characters in both texts. Three different metrics were selected: Cosine, Overlap and Sorensen (Dice);

2. **Edit-based(3):** metrics of this type are based on counting the minimum number of operations performed to transform one string into the other. Levenshtein, Hamming [144] and Jaro-Winkler were used;

3. **Sequence-based(2):** unlike token-based, here the order counts and similarity is based on sequences. One way is to measure the *longest common substring* between two given strings. The principle is that sentences with longest shared sequences are more likely to be similar. A variation of this idea is also employed in this work, using the RatcliffObershelp similarity [145];

4. **Compression-based(4):** it is similar to edit-based but similarity is extracted from the shortest computer program that can convert one string (in this case, represented as a bit vector) to another. The representative algorithm used was Normalized Compression Distance [146]. Four different variations were considered, depending on the compressor, being described in the *textdistance* library (Subsection 5.1.4) as *bwtrle*, *bz2*, *lzma* and *zlib*.

To extract the features, each student answer is compared against all the teachers' answers to the question. So, if there are three available teachers' answers for a question,

the feature set for each student answer will consist of 36 features (3 reference answers × 12 metrics).

For performing the experiments, a similar scheme as the one presented in the previous section was used. The morphological reduction and stopwords removal were maintained. Moreover, a new binary parameter was implemented: to do or not to do Question Demoting (QD). This technique consists in removing from the students' and teachers' answers words that are also used in the question statement. This idea was introduced in the work of [75] and used afterwards in [85] and [82]. Therefore, the parameter variation was defined as $stopwords\_removal = \{True, False\} \times morphological\_reduction = \{None, Stemming, Lemmatization\} \times question\_demoting = \{True, False\}$. The best parameter set was found to be: to use **none** morphological reduction, **Random Forest** and **applies** both **stopwords removal** and **question demoting**. A visualization of the question demoting impact on the performance can be seen in Figure 25.

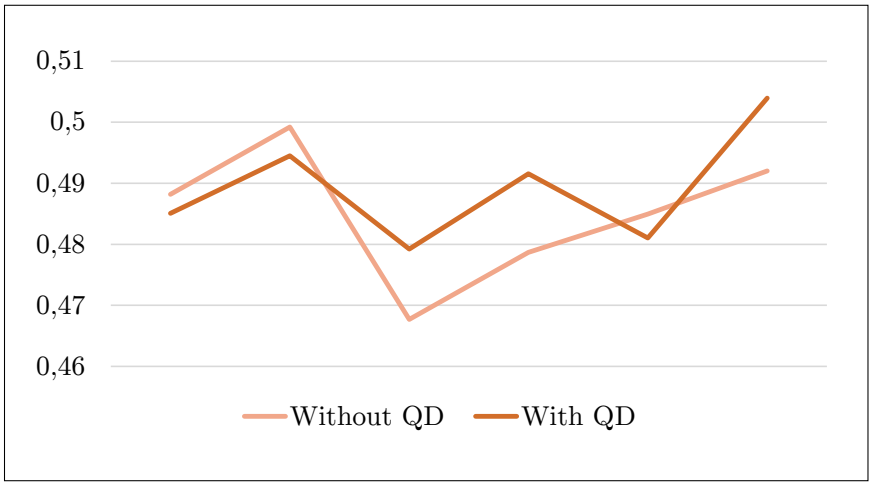


Figure 25 – Question demoting impact

As can be seen, the impact of performing question demoting or not had not a huge impact on performance, with three wins for performing and three for not performing. The difference between the average among executions with and without performing question demoting is only 0.004, which is wispy, but a little better when performed. Important to reminder that the difference is based on the average of questions and the influence of question demoting considering each question individually is very different.

5.3.1 Using Student Answers

Despite the relatively good performance obtained so far, it can still be improved. An interesting idea introduced in the work of [67] (and not found in other works ever since) was to use the students' answers who were awarded with the maximum grade as they were also reference answers. This idea is interesting because students may relate more with the correct answer from another classmate's than with the teacher's. In order

to implement this idea, between four and six student top answers were selected for each question to compose a new set of reference answers, with each question containing eight reference answers in total. The number of eight was decided for two main reasons: 1) one of the questions has only eight correct answers summing the teachers' and students' answers, so in order to not have one question being different in the numbers, this value was used as ground; 2) with a number above eight, the dimensionality would get too high and would slow the experimental executions.

Using the best parameter set discovered in the previous experiment, a new experiment was set: to vary the number of reference answers from one to eight. The average between questions' scores by the number of reference answers can be seen in Figure 26. It also reports the previous approach score (which used between 2 and 4 teachers' answers).

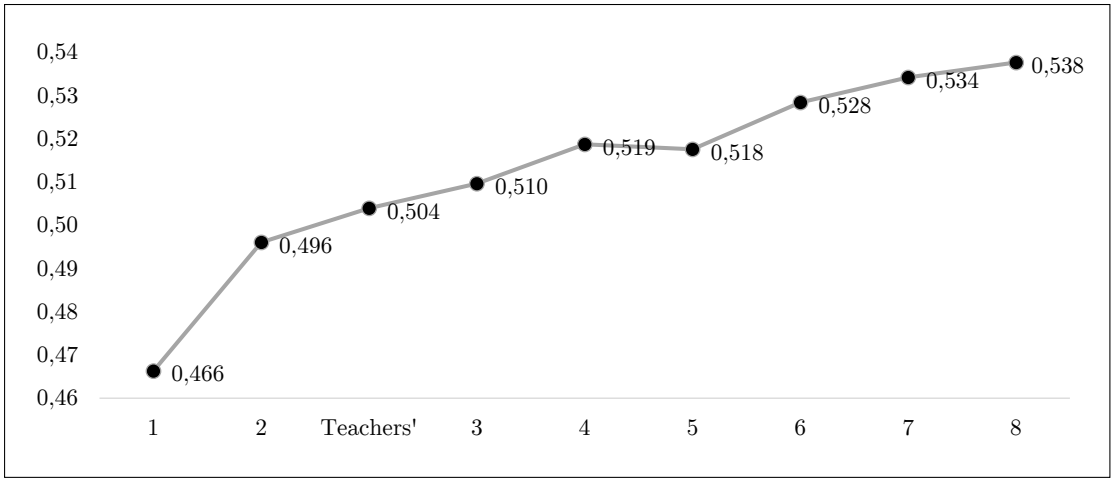


Figure 26 – Impact of the number of reference answers

As it can be noticed, as the number of reference answers increases the performance tends to increase as well. Therefore, in order to accomplish the best possible score, the number of eight reference answers was chosen. Numbers greater than eight could be investigated in the future.

Regarding the scores obtained by each question individually, Figure 27 presents the difference between the new approach with eight reference answers and the old one. Points above the zero line represents wins for the new approach and below wins for the old one.

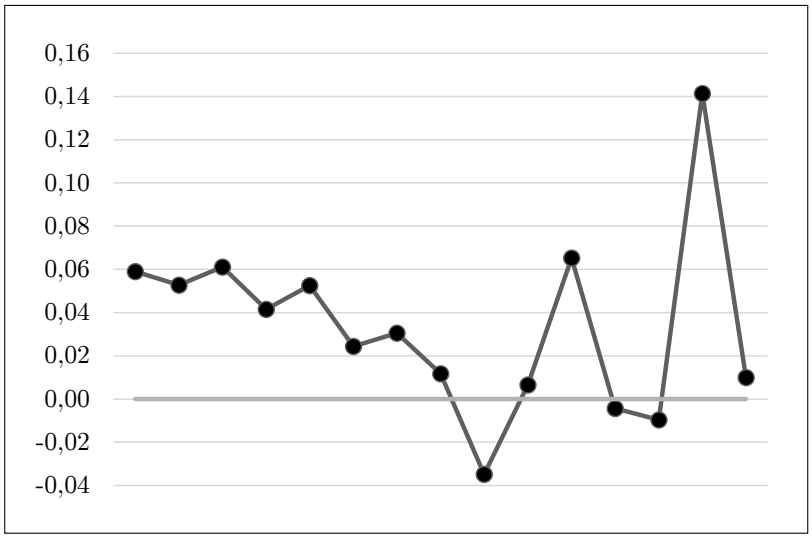


Figure 27 – Differences between the old and the new approach for each question

Figure 27 shows that not all questions are positively affected by more reference answers. However, most of them are, and with scores up to 0.14 of difference. Negative differences are seen only for three of the questions and are only up to 0.04. Hence, the new approach can be considered better in general.

Considering the number of times that a specific number of reference answers performed better among the 15 questions, Figure 28 exhibits this statistic. As it can be noticed, the number seven got the higher scores in most of the time. However, the numbers four, five, six and eighth also got a considerable number of wins as well (with the number eight also obtaining the best averaged score).

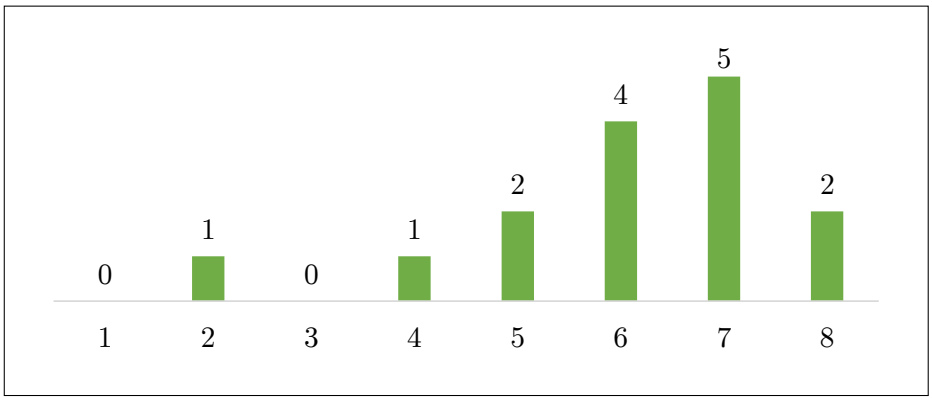


Figure 28 – Number of *win times* for each number of reference answers

The results for each question, using the new approach, achieved with the best parameters are presented in Table 27.

Table 27 – Lexical Similarity results for all questions

Q_ID	Acc	LK	QK	Q_ID	Acc	LK	QK
1	0,491	0,470	0,619	**9**	0,690	0,313	0,379
2	0,653	0,579	0,703	**10**	0,799	0,716	0,814
3	0,711	0,440	0,502	**11**	0,554	0,450	0,540
4	0,576	0,578	0,721	**12**	0,462	0,425	0,546
5	0,589	0,591	0,719	**13**	0,393	0,267	0,344
6	0,529	0,408	0,496	**14**	0,570	0,512	0,656
7	0,690	0,369	0,454	**15**	0,701	0,727	0,830
8	0,574	0,341	0,460	**Mean**	0,599	0,479	0,585

5.4 Semantic Similarity

As stated in the previous section, words that are similar in their shape can be not similar when considering their meaning. Also, words with very different shapes can have very close meaning. This property of natural languages brings a challenge to its correct processing by computers. Among other purposes, semantic networks were created to aid with this issue. The most representative and popular semantic network is WordNet [24], a network where words are grouped in synsets, that are interlinked by their conceptual-semantic and lexical relationships, providing means to measure semantic similarity.

There are a few established algorithms that can compute word-to-word similarity in WordNet. They do so by walking through the links between synsets and measuring how close or distant they are, if they have hierarchical relationships, among other indicators. Six of these algorithms were considered for the experiments: Leacock & Chodorow [100], Wu & Palmer [101], Lin [102], Resnik [103], Jiang & Conrath [104] and Shortest Path. These metrics are also used by many other ASAG works [9, 75, 64, 80, 85]. The corpus used for statistical information required by the Resnik, Lin and Jiang & Conrath algorithms was the Mac-Morpho Brazilian Portuguese annotated corpus (words with their part-of-speech tags) [147].

In order to use word-to-word similarity for measuring answers similarity, an algorithm was implemented as proposed in [110], with the difference that here the median function was also experimented, in contrast with only the mean function of the original algorithm. Moreover, the use of both functions is also explored. The complete algorithm is shown in Algorithm 2 and Algorithm 1 (Algorithm 2 has a call to Algorithm 1). The algorithm is applied to every $(stuAns, refAns)$ pair in the dataset.

Algorithm 1: Semantic Similarity (Adapted from [110])

Require: $t1, t2, metric, meanMedianBoth$

$maxScores \leftarrow []$

for $raSynSet$ in $t1$ **do**
  $raScores \leftarrow []$
  for $saSynSet$ in $t2$ **do**
    if $raSynSet.postag == saSynSet.postag$ **then**
      $raScores.append(raSynSet.similarity(saSynSet, metric))$
    end if
  end for
  $maxScores.append((max(raScores)))$
end for
if $meanMedianBoth ==' mean'$ **then**
  return $mean(maxScores)$
else
  return $median(maxScores)$
end if

Algorithm 2: WordNet Symmetrical Metrics' Scores

Require: $refAns, stuAns, meanMedianBoth$

$openClass \leftarrow \{'NOUN','VERB','ADJ','ADV'\}$

$metrics \leftarrow \{'lch','path','res','wup','lin','jcn'\}$

$refAnsTagged \leftarrow tag\_sentence(refAns)$

$refAnsTaggedS \leftarrow [(w,t)$ for (w,t) in $refAnsTagged$ if t in $openClass]$

$ra \leftarrow from\_tagged\_sentence\_to\_synsets(refAnsTaggedS)$

$stuAnsTagged \leftarrow tag\_sentence(stuAns)$

$stuAnsTaggedS \leftarrow [(w,t)$ for (w,t) in $stuAnsTagged$ if t in $openClass]$

$sa \leftarrow from\_tagged\_sentence\_to\_synsets(stuAnsTaggedS)$

$results \leftarrow []$

if $meanMedianBoth ==' both'$ **then**

  for $metric$ in $metrics$ **do**

    $results.append((sim\_t1\_t2(ra, sa, metric,' mean') +$

    $sim\_t1\_t2(sa, ra, metric,' mean'))/2)$

  end for

  for $metric$ in $metrics$ **do**

    $results.append((sim\_t1\_t2(ra, sa, metric,' median') +$

    $sim\_t1\_t2(sa, ra, metric,' median'))/2)$

  end for

else

  for $metric$ in $metrics$ **do**

    $results.append((sim\_t1\_t2(ra, sa, metric, meanMedianBoth) +$

    $sim\_t1\_t2(sa, ra, metric, meanMedianBoth))/2)$

  end for

end if

return $results$

As the preceding sections, some parameters were tested to seek for better performance. In this case, it was mandatory to perform lemmatization as the library do not support the morphological variations. Accents were not removed, to increase matches in the semantic network (that uses the proper accents). The varying parameters were: stopwords removal $\{True, False\}$, question demoting $\{True, False\}$ and the aggregate function to use inside the algorithm $\{'mean','median','both'\}$, totalizing 12 possible combinations. As a novelty from previous sections, the aggregate function comparison is shown in Figure 29.

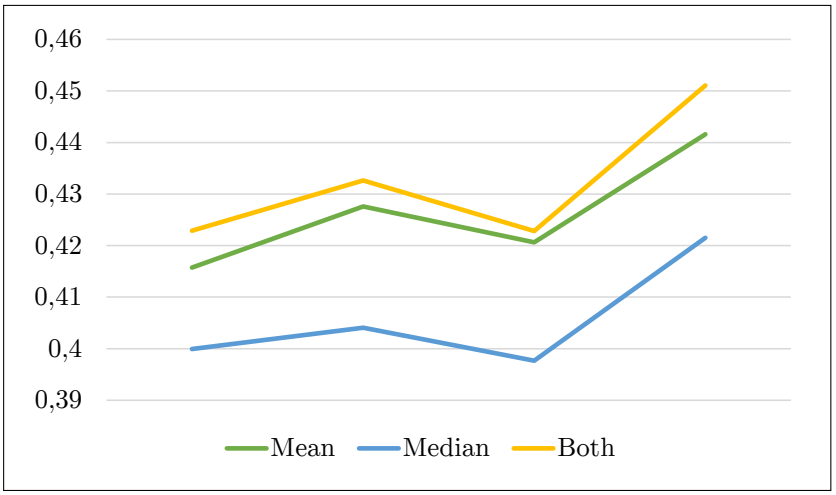


Figure 29 – Aggregate function impact

As expected, the mean function performed better than the median function, as it is part of the original algorithm in [110]. However, the median function also had a great performance, specially for some of the questions. Therefore, by taking these aspects in account, it is easy to understand why the use of both functions together performed better than their alone application.

Concerning the other testing parameters, the results followed those from the lexical similarity, with **stopwords removal and question demoting succeeding** at improving overall performance. Once again, **Random Forest** obtained the best aggregate result. However, results followed previous sections and XGB and SVM also had a good performance.

5.4.1 Using Student Answers

Following the idea introduced in the preceding section, an experiment to measure the impact of more reference answers on semantic similarity was tested. The same eight reference answers from before were used and the experiment followed the same protocol, varying the number of reference answers from one to eight. The average between questions' scores by the number of reference answers can be seen in Figure 30. It also reports the previous approach score (which used between 2 and 4 teachers' answers).

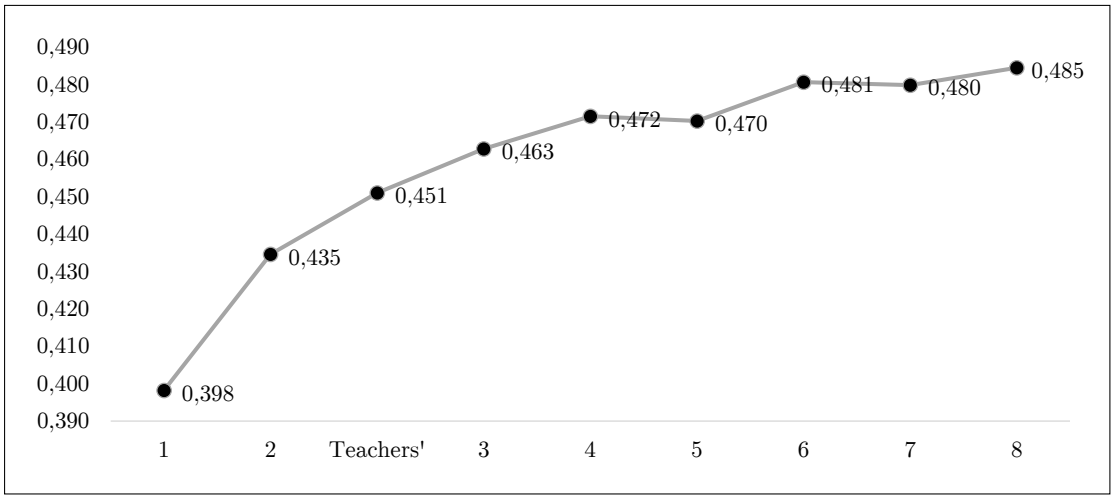


Figure 30 – Impact of the number of reference answers

As it can be noticed, as the number of reference answers increases the performance tends to increase as well, just like it happened for the lexical similarity as well. These results instigates on performing future experiments with numbers greater than eight.

The results of the impact of using student answers as reference answers, shown in Figures 26 and 30, indicates that this strategy can bring really good performance. One of the reasons for that may be related to the matureness of teachers and students. As these two groups of people are in different levels of maturity, the correct answer written by a student can possibly related more to an answer than one written by a teacher, because the former would write using a different language style than the latter. This can possibly be one of the reasons and it must be investigated by future researches, as it is a matter of great impact on performance.

Regarding the scores obtained by each question, Figure 31 presents a comparison between the previous approach and the new one with eight reference answers. It shows the difference between the new and the old approach. Points above the zero line represents wins for the new approach and below wins for the old one.

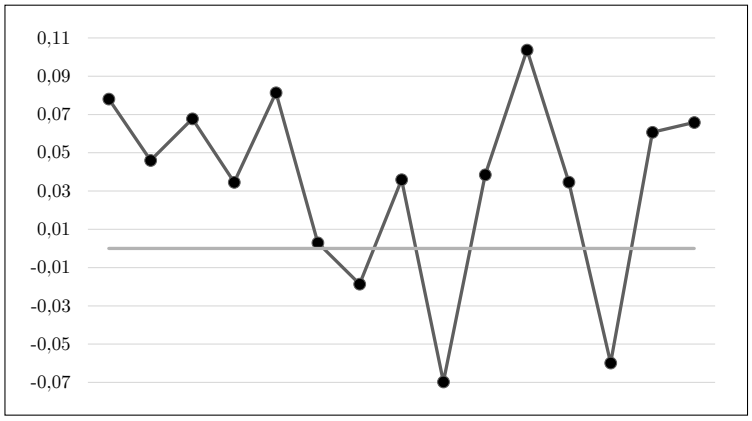


Figure 31 – Differences between the old and the new approach for each question

Just like for the lexical similarity, not all questions are positively affected by more reference answers. As a matter of fact, the graphics looks very similar, indicating that some questions are less likely to improve performance using the new approach, independently if it is lexical or semantic similarity. However, most of the points are above the line, indicating that the new approach is better in general. Positive differences are up to almost 0.11 and negative differences up to 0.07.

The same analysis performed on Figure 28 is also performed for the semantic similarity results, presented in Figure 32. In this case, the number eight performs better by far. However, numbers four, five and six presented considerable wins. The sum of *win times* among the eight possibilities is not 15 because some tie cases happened.

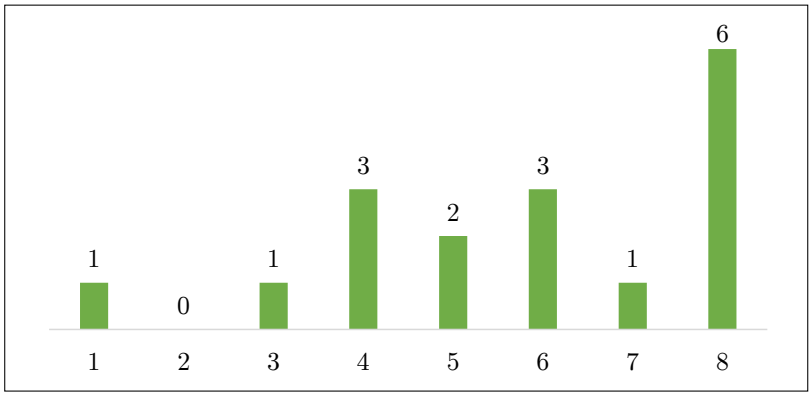


Figure 32 – Number of *win times* for each number of reference answers

The results for each question, using the new approach, achieved with the best parameters are presented in Table 28.

Table 28 – Semantic Similarity results for all questions

Q_ID	Acc	LK	QK	Q_ID	Acc	LK	QK
1	0,494	0,466	0,608	**9**	0,671	0,417	0,505
2	0,613	0,510	0,629	**10**	0,787	0,614	0,664
3	0,671	0,387	0,456	**11**	0,465	0,320	0,396
4	0,554	0,557	0,705	**12**	0,473	0,455	0,587
5	0,543	0,499	0,605	**13**	0,330	0,160	0,219
6	0,473	0,345	0,444	**14**	0,523	0,418	0,540
7	0,676	0,343	0,429	**15**	0,664	0,687	0,797
8	0,553	0,327	0,444	**Mean**	0,566	0,434	0,535

5.5 Text Statistics

Another group of features that should be considered is Text Statistics. These features extracts some stats from the student answer. Also, there can be features extracted from some ratio between student and reference answers. There are many different types of text statistics used in ASAG literature. In this section, some of them are explored and

used in the experiments. The list below presents the features used in this work, how many each group uses (there are 22 in total) and which works in the literature also uses it.

- **Length Ratio(4):** the length ratio between the student and the question statement. Also, the maximum, minimum and mean of the ratio between the student answer and the reference answers (variations used in [77, 76]). A large distance between the student and reference answer may indicate an incorrect answer;
- **Counts(16):** count per answer of: characters [74, 9, 54, 53], words [53, 54, 62], sentences [54], commas, unique words [53], negation words [62] and each part-of-speech (POS) tag (in the universal POS tagset) [129]. Style of answers by the counts of their components may indicate better writing;
- **Average Word Length(1):** the simple average of the length of words in the answer [53]. Can indicate if answers with larger words turns in correct or incorrect grading;
- **Words per Sentence (Average)(1):** the size of each sentence in terms of words. Another style writing feature to measure if shorter or larger sentences can lead to correct answers;

5.5.1 Experiments

For this set of features none prior preprocessing is performed. This is done because some of the features needs the text in its raw condition. Also, morphological reduction and stopwords removal would affect answers in a proportional way, probably not affecting the results. From a first execution with all the features, three analysis were performed. The first was to compare the three classifiers, with results in Table 29.

Table 29 – Text Statistics results

X	Acc	LK	QK	BK
RF	0,533	0,341	0,423	0,382
XGB	0,533	0,342	0,426	0,384
SVM	0,546	0,324	0,402	0,363

The results in Table 29 showed that RF and XGB performed almost the same. SVM got a higher accuracy than the two others. However, it performed worst than them considering the kappa scores. Overall, the results indicates only fair to moderate agreement and scores are much lower than previous feature sets. The second analysis is shown in Figure 33.

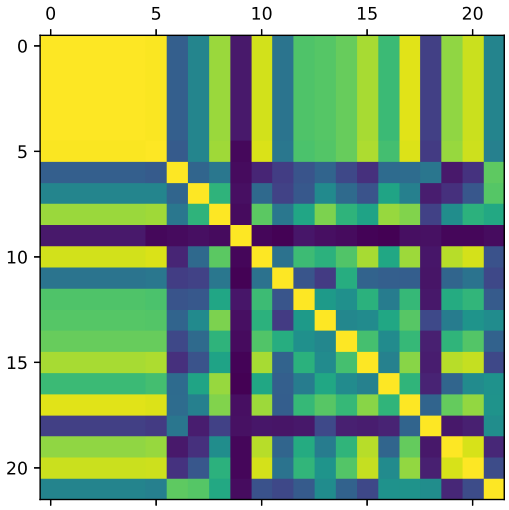


Figure 33 – Correlation matrix

Figure 33 presents a matrix where lines and columns represents each feature and their intersections exhibits how two features are correlated. The main diagonal is full correlated, because it indicates the correlation of a feature with itself. Darker cells indicates low correlation and clearer cells indicates higher correlation. The matrix exposes a high correlation between the first six features: the four length ratio features and the character and word count. They are highly correlated among themselves because they follow the same proportion among most samples. This conclusion about these features is confirmed by the third analysis, shown in Figure 34.

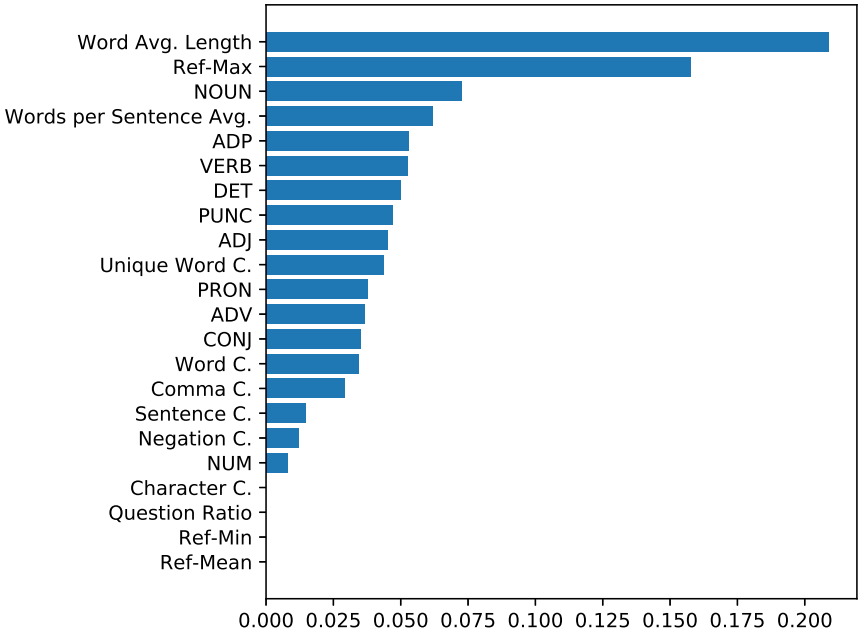


Figure 34 – Features' importance

The scores presented in Figure 34 were extracted from a XGB execution. In its internal behaviour, XGB assigns an importance value for features in order to better use them to build the final model. It considered the character count and three of the ratio features as having zero importance because they were redundant with the other two features. Therefore, the algorithm kept only those features that can impact on performance.

Based on these analysis, a new experiment removing those four redundant features was performed. The analysis comparing the three machine learning algorithms was performed again, with results in Table 30. Comparing with Table 29, SVM got a little decrease in all metrics, XGB performed exactly the same in all metrics and RF increased its performance in all metrics. The same performance for the XGB is because its internal implementation already disconsidered the redundant features. As for the SVM and RF, it affected them oppositely.

Table 30 – Text Statistics results (after)

X	Acc	LK	QK	BK
RF	0,543	0,352	0,431	0,392
XGB	0,533	0,342	0,426	0,384
SVM	0,543	0,318	0,395	0,357

Finally, the results using Random Forest, without redundant features and detailing performance for all questions are presented in Table 31.

Table 31 – Text Statistics results for all questions

Q_ID	Acc	LK	QK	Q_ID	Acc	LK	QK
1	0,398	0,278	0,378	**9**	0,666	0,266	0,326
2	0,637	0,524	0,621	**10**	0,707	0,406	0,491
3	0,665	0,312	0,330	**11**	0,491	0,370	0,463
4	0,480	0,394	0,503	**12**	0,441	0,317	0,378
5	0,492	0,454	0,584	**13**	0,349	0,160	0,192
6	0,427	0,226	0,259	**14**	0,549	0,428	0,523
7	0,645	0,244	0,327	**15**	0,586	0,549	0,650
8	0,611	0,355	0,443	**Mean**	0,543	0,352	0,431

5.6 Word Embeddings

One of the greatest novelties in natural language processing in the last few years is word embeddings. As discussed in Section 2.4, the work of [40] introduced Word2Vec in 2013, a technique for representing words in vectors in an efficient manner. The authors presented two different models for accomplishing their goal: Skip-gram and CBow. From their work, several researches followed the embeddings path, leading to new and refined techniques.

In 2014, researches from Stanford University released GloVe (Global Vectors for Word Embeddings) [108]. The main difference from the Word2Vec algorithm is that GloVe, beyond using context-based learning as Word2Vec, also uses global text statistics from the whole corpus by constructing a word co-occurrence matrix (like older methods such as Latent Semantic Analysis). This additional technique can improve the overall results, as demonstrated in their work [108].

Following, in 2016 the Facebook Research team presented FastText [148], a new extension to the Word2Vec algorithm. In Word2Vec, each word in the corpus is considered as an atomic entity, used for training the model. The novelty from FastText is that it treats words as a composition of character ngrams and hence, the vector for a specific word is determined from the sum of its character ngram's vectors. This representation difference can have a great impact depending on the data.

Training these word embeddings algorithms on a large corpus is a laborious activity. In order to help researchers to deal with natural language processing tasks in Portuguese, [149] trained word embeddings on a large Portuguese corpora, composed of 17 different corpus, totalizing 1.395.926.282 tokens. They used this data to train on the three aforementioned algorithms (Word2Vec, GloVe and FastText), making available different versions for each of them, concerning the model (Skip-gram or CBow) and the number of vector's dimensions.

An even newer approach (2017), that implements sentence embeddings instead of word embeddings, was proposed in [150]: InferSent. However, sentence embeddings are out of the scope of this work as there is no sentence embedding's implementation readily available for the Portuguese language.

Regarding the use of embeddings on the ASAG literature, eight works were found using some of the algorithms. Two of them uses InferSent [129, 123]. Among the other six works, Word2Vec was used in five, GloVe was used in two and FastText was not used. The training data used among the six works was Wikipedia (2), Google News (2) and a composition of Wikipedia, the British National Corpus and the WaCky corpus (2) (more details in [151]). Concerning the number of the vector's dimensions, two works uses 100, two uses 300 and the other two 400.

For this work, 15 embeddings models were considered, based on the availability of pre-trained Portuguese embeddings from [149] and in the literature's use. Twelve of them are a combination of $algorithm = \{Word2Vec, FastText\} \times number\_of\_dimensions = \{50, 100, 300\} \times skip\_or\_cbow = \{SkipGram, CBow\}$. As GloVe do not have a skip-gram/cbow variation, it only varies for the number of dimensions (50, 100 and 300).

All these word embeddings algorithms, implementations and variants gives as output a vector representation for each word. In order to use these vectors as features for

machine learning in ASAG, some different approaches are used in the literature. The most common and used way to represent answers using word embeddings is to sum or average among the vectors of each content word of the answer. These representation techniques (referred from now on as summation and average models) are used in five of the six works [9, 82, 90, 80, 120]. A variation of this modeling technique is presented in [80]. Instead of using only the vector's sums, it also represents each answer as the concatenation of three vectors: summation, min and max. The min and max vectors are created by taking the max and min element for each dimension i among all the ith dimensions for each word (example: the 5th dimension of the min vector is the minimum value across all 5th position vectors extracted from each answer's word). This models is referred here as summinmax.

Considering the three ways of representing an answer **(summation, average and summinmax)**, they can be used as features for ML in two different manners: 1) as they are, that is, using only the student answer itself as feature (in a way related to the bag-of-ngrams) referred in this section as the **representation** approach; 2) to create a similarity score between the student and reference representation vectors, using a metric that calculates the distance between real number vectors (usually using the Cosine distance, but also the Manhattan (city-block) and Euclidean distance) referred in this section as the **similarity** approach.

Another approach used in two works [85, 80] was to implement Mohler and Michalcea's algorithm [152] to calculate text semantic similarity, using the same algorithm as the one presented in the previous section (Semantic Similarity). The only difference is that instead of using WordNet's algorithms (such as Lin, Shortest Path, etc) to calculate word-to-word similarity, the cosine, euclidean and Manhattan distance are used as metrics. Preliminary experiments using this approach produced results way worse than other approaches. In addition, it took much more time than other approaches to run. For these reasons, this approach was no further used in the experiments.

Finally, other two different techniques were found for generating features [123, 129]. However, as they depend on the representation technique (summation, average or summinmax), they were taken in account after the performance of the experiment that determined the best representation approach.

For the preprocessing, four variations were considered: $stopwords\_removal = \{True, False\} \times morphological\_reduction = \{None, Lemmatization\}$. Stemming was not take into account as the embedding models were trained using real words and as stemming produces non-words, it was not considered. Accents were not removed for the same reason, to match the words used for training the embeddings in [149].

All things considered, 360 executions were performed to determine the best parameter set and perform comparisons among variants. The combinations are composed of: 4 preprocessing options × 15 embedding models (Word2Vec, GloVe, etc...) × 3 repre-

sentation techniques (summation, average and summinmax) × 2 approaches (only representation or similarity). Also, as performed for all other experiments, the three different classifiers were also tested.

For the similarity approach, all the eight reference answers introduced in the Lexical Similarity Section were used as features. This was done because each similarity pair (student and reference answer) uses only three features (Cosine, Euclidean and Manhattan scores), which is too little. By using the eight reference answers, each row was composed of 24 features. Another reason is that using more reference answers in the two previous sections improved the results. Another technique used for all approaches was Question Demoting. Its efficiency was also demonstrated in previous sections and preliminary experiments using embeddings also showed some improvement.

Considering everything that was discussed in this section so far, Subsection 5.6.1 reports the results for the performed experiments, presents graphic visualizations and the final results as performed in the previous sections of this chapter.

5.6.1 Results and Discussion

The first result is that the best performance was achieved with: **Random Forest**, **lemmatization**, **not** removing stopwords, using **FastText** with **300** dimensions and the **skip-gram** model and using the **summinmax representation** only (not similarity).

Concerning the preprocessing, lemmatization succeed once again, proving its efficiency by performing better in 88% of the executions, while only 12% performed better without its application. Regarding the stopwords removal procedure, even tough its application performed better in 74% of executions, not applying it led to the best score when combining all the parameters (although it was not a very significant difference).

Regarding the machine learning algorithms, results followed the previous sections, with Random Forest being in the best execution, RF and XGB performing almost the same and SVM performing worst considering kappa but being very close from RF and XGB considering accuracy.

With regard to the representation technique, Figure 35 illustrates the performance for summation, average and summinmax (without similarity).

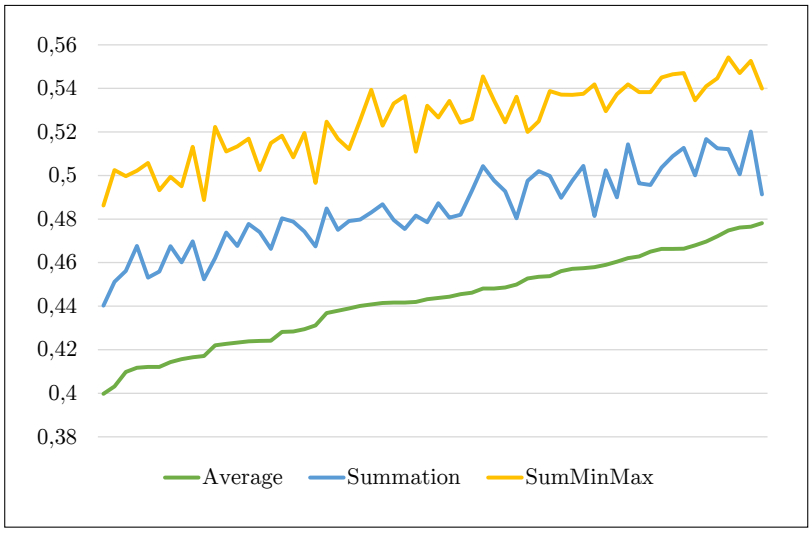


Figure 35 – Comparison among different representation techniques

The approach proposed by [80] (summinmax) showed to perform better against common approaches such as summation and average, at least for this dataset. According to the authors, the idea of this representation is to capture the boundaries of each sentence. Summinmax performed, in average, 0,034 better than summation and 0,076 better than the average approach. Figure 35 also shows that the differences among approaches are constant throughout all executions.

Concerning the similarity approach, it is worth mentioning that it performed a little better than the average representation approach and way worst than summation and summinmax represented in Figure 35, with its best score achieving 0,495. Comparing to the 0,55 of the representation approach, it is considerable worst. However, this value is a little better than the best score obtained using WordNet similarity in the previous section (that was 0,485). As regard to which representation approach performed better for embeddings similarity, the results are interlaced, each one of three techniques won a considerable percentage of executions and the average was very close. Summing up, the representation technique had a great impact when using it by itself but a low impact for similarity usage. Also, similarity using word embeddings performed a little better than similarity using WordNet.

Following, the results regarding the embeddings model's characteristics are presented in Figures 36, 37 and 38. Firstly, Figure 36 shows a comparison among the executions that used cbow against skip-gram. The graphic represents the difference from skip-gram minus cbow, so the dots above the zero line represents skip-gram wins and dots below cbow wins. It is clear that the skip-gram approach performed mostly better among all executions. The difference is even greater in the right side of the graphic, that represents the FastText runs, as opposed to the left Word2Vec runs.

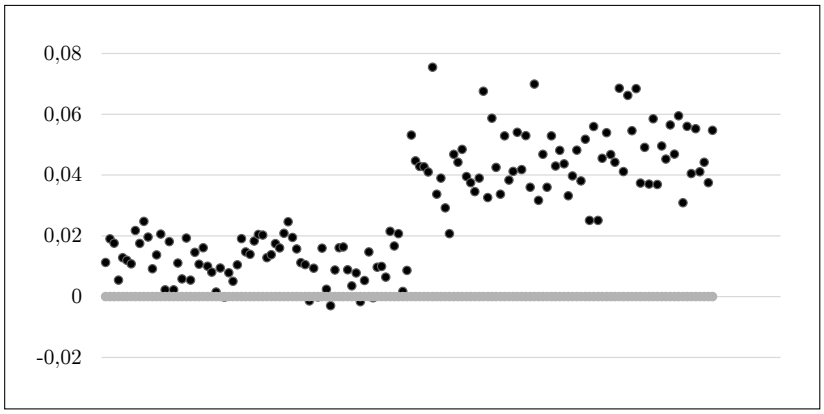


Figure 36 – Comparison between cbow and skip-gram approaches

Regarding the number of dimensions, Figure 37 compares the performance of the three values. Generally, using 300 dimensions performed better than 100 that performed better than 50. However, the lines are intertwined, that shows that despite the overall trend that more dimensions gives better results, the differences are not too high and in some cases the 50 line crosses the 100 line and this in its turn crosses the 300 line.

Figure 37 – Comparison between different dimensions for embeddings' vectors

Finally, the comparison between Word2Vec, GloVe and FastText is reported in Figure 38 [4]. As it can be seen, Word2Vec performed consistently worst than the other approaches, winning only once against GloVe. FastText and GloVe had a smaller difference between themselves but FastText performed better most of the time. These results follows the chronological releases of each approach, when the more recent the algorithm, the better it performs. This can be considered an expected result, as GloVe and FastText were proposed as extensions to improve Word2Vec's performance.

[4] Cbow executions were left out so Word2Vec and FastText would have the same execution points as GloVe in order to perform the comparison. Also, skip-gram was picked over cbow as it performed better.

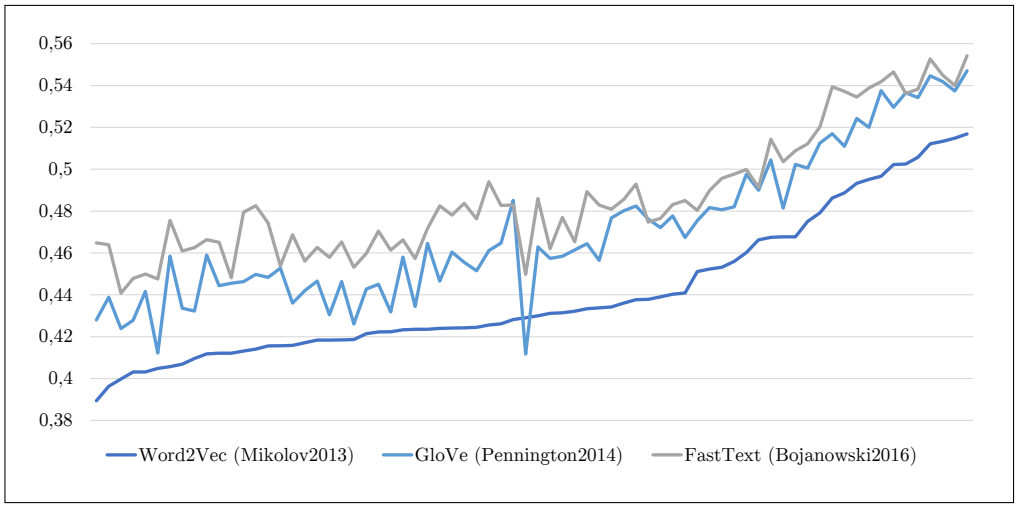


Figure 38 – Comparison between different word embeddings' algorithms

After founding out the best possible combination for word embeddings, the feature representation from [123] (Equation 5.1) and [129] (Equation 5.2) (referred as Marvaniya's and Saha's representation features, respectively) were considered. The q, r, a letters indicates the embeddings' vectors from the question statement, the reference answer and the student answer, respectively. Also, $*$ and $-$ represents element-wise multiplication and subtraction, respectively. Finally, the $|$ symbol represents concatenation. The authors in [123] excluded the question embedding for limiting the high dimensionality (their embeddings' implementation InferSent uses 4096 for each sentence embedding, which would give a 8192 for their implementation and 24576 for Saha's representation).

$$S_{feat}(q, r, a) = [a - r|a * r] \tag{5.1}$$

$$S_{feat}(q, r, a) = [r * a|r - a|r * q|r - q|a * q|a - q] \tag{5.2}$$

The authors in [129] Saha's representation used the question to extract *the novel information expected in the student answer* and *the novel information expressed in the student answer*. However, a much easier and straightforward technique allows for achieving the same goal: Question Demoting, used in the two preceding sections. Also, triple the feature vector dimensionality only for this purpose seems exaggerated. As a consequence, we first experimented Marvaniya's representation, which resulted in no improvement in the performance (more details further ahead). Therefore, considering the question demoting option, the huge dimensionality and Marvaniya's performance, we opted not to use Saha's representation.

Regarding Marvaniya's representation, we tested a implementation using the best parameters (Random Forest, lemmatization, no stopwords removal, FastText, skip-gram,

300 dimensions and using summinmax representation). However, the final dimensionality got too high (300 dimensions $\times$ 3 (summation, min and max vectors) $\times$ 2 ($a - r$ and $a * r$)), totalizing 1800 dimensions for **each** reference answer (the number of reference answers was tested from 1 to 8 but the exact same result was obtained for all n values of reference answers. We hypothesize that this is caused by the high dimensionality of the feature vector (from 1800 to 14400 dimensions), specially harmful because each question has only an average of 498 samples (the proportion between samples and features is too high)).

Nonetheless, the results for the Marvaniya's representation, in comparison with the previous *best score* approach, are presented in Table 32.

Table 32 – *Best Score* vs Marvaniya's representation

Q_ID	***Best Score***	**Marvaniya's**	Q_ID	***Best Score***	**Marvaniya's**
1	0,549	0,545	**9**	0,467	0,463
2	0,645	0,639	**10**	0,738	0,734
3	0,474	0,509	**11**	0,472	0,489
4	0,709	0,688	**12**	0,588	0,585
5	0,613	0,591	**13**	0,346	0,320
6	0,478	0,449	**14**	0,615	0,575
7	0,347	0,367	**15**	0,808	0,789
8	0,484	0,465	**Mean**	**0,556**	**0,547**

As it can be seen on the table, Marvaniya's representation was only able to improve results for three questions (ids 3, 7 and 11, gray marked). However, considering the overall performance, Marvaniya's failed at improving the results. Therefore, the *best score* approach was kept as the chosen one. The detailed results for all the questions and metrics are presented in Table 33.

Table 33 – Word Embeddings results for all questions

Q_ID	Acc	LK	QK	Q_ID	Acc	LK	QK
1	0,465	0,465	0,633	**9**	0,697	0,416	0,518
2	0,667	0,588	0,701	**10**	0,802	0,699	0,776
3	0,716	0,445	0,503	**11**	0,524	0,424	0,520
4	0,640	0,646	0,772	**12**	0,508	0,520	0,656
5	0,556	0,549	0,678	**13**	0,433	0,312	0,381
6	0,513	0,420	0,536	**14**	0,604	0,549	0,681
7	0,674	0,310	0,384	**15**	0,722	0,757	0,858
8	0,613	0,423	0,544	**Mean**	0,609	0,502	0,610

5.7 Combining

After experimenting with each approach on the five preceding sections, Table 34 groups the six approaches and their parameter configuration (it was decided to include the word embeddings' similarity as the sixth approach for analysis as it showed considerable good results). All approaches uses Random Forest as the classification algorithm. Each method received an acronym, being in order from top to bottom in the table: Bag-of-ngrams, Word Embedding Representation, Lexical Similarity, Word Embedding Similarity, WordNet Similarity (Semantic Similarity) and Text Statistics.

Table 34 – All six approaches and their characteristics

	Morp. Red.	SWR	Q.D.	N.o.R.A	Other Parameters	Dimensionality
NGRAMS	Lmm.	True	-	-	wnmf=450,cnmf=500,range=(2,4)	450+500=950
WEREP	Lmm.	False	True	-	fasttext,300d,skip,summinmax	300x3=900
LEXSIM	None	True	True	8	-	8x12=96
WESIM	Lmm.	True	True	8	word2vec,300d,skip,summinmax	8x3=24
WNSIM	Lmm.	True	True	8	agg_func='both'	8x12=96
TXST	None	False	-	-	-	18

Morp. Red.: Morphological Reduction. **SWR**: StopWords Removal. **Q.D.**: Question Demoting. **N.o.R.A.**: Number of Reference Answers. **Lmm.**: Lemmatization.

The results achieved for each question by the six different approaches are presented in Table 35 (using the *bk* scores for comparison). The columns of the table are ordered from the best to the worst approach, from left to the right (considering the mean score in the last row). The table is colored in order to highlight discrepancies among questions. The colors are specially helpful at spotting questions where a specific approach was particularly bad or good. It also helps spotting questions that diverged from the more common pattern.

Table 35 – Results from all previous sections side by side

Q_ID	NGRAMS	WEREP	LEXSIM	WESIM	WNSIM	TXST
1	0,612	0,586	0,560	0,447	0,543	0,328
2	0,645	0,636	0,672	0,646	0,557	0,573
3	0,575	0,475	0,481	0,439	0,415	0,321
4	0,775	0,728	0,667	0,654	0,647	0,448
5	0,685	0,638	0,659	0,591	0,530	0,519
6	0,487	0,452	0,446	0,387	0,394	0,242
7	0,364	0,389	0,401	0,292	0,377	0,286
8	0,495	0,442	0,445	0,439	0,357	0,399
9	0,484	0,461	0,323	0,476	0,425	0,296
10	0,820	0,714	0,769	0,576	0,658	0,449
11	0,517	0,501	0,478	0,479	0,390	0,417
12	0,592	0,572	0,502	0,446	0,511	0,347
13	0,362	0,326	0,317	0,232	0,216	0,176
14	0,639	0,599	0,582	0,595	0,461	0,475
15	0,808	0,794	0,764	0,741	0,761	0,600
Mean	0,591	0,554	0,538	0,496	0,483	0,392

A first insight from Table 35 is that the techniques considering only words and their representation (NGRAMS and WEREP) got the higher scores. They also have a much larger dimensionality than the other feature's set (900's against dozens from the others). The only question that NGRAMS got a smaller score than WEREP was in question 7. For all the others, WEREP follows NGRAMS closely, but do not achieves the same performance.

A second attention caller is TXST, holding the worst performance. This is expected as this group of features only accounts for simple text style statistics. Even though TXST got a bad performance compared to the others, it got reasonable scores for questions 2, 8 and 11, not that far away from the others. It even won from WNSIM in four of the questions.

Following, as intermediates, there are the three similarity approaches. The lexical similarity approach usually got higher scores than the other two. The difference between WE and WN similarity is smaller but WE usually performs better than WN. Interesting to notice that LEXSIM got a specially low score for question 9, losing by far from the other similarity approaches. Still regarding question 9, the WESIM method got a score higher than WEREP and almost as the same as NGRAMS.

Other discrepancies from the mean score sequence are from questions 2 and 7. For these questions, LEXSIM got the highest score, even greater than NGRAMS and WEREP. Another noticeable discrepancy is from question 9, in which WESIM wins from four approaches and gets real close to NGRAMS. It is by far the best question from WESIM.

Finally, concerning WNSIM, its score for question 7 was considerable good, even beating NGRAMS and losing only for WEREP and LEXSIM. However, as the second worst approach, question 7 is its only highlight. In fact, WNSIM performs so badly that it even loses for TXST in four questions (2, 8, 11 and 14) and it gets close of losing in another two (5 and 13).

In summary, the main insight from Table 35 is that despite of their general performance order, each approach has its advantages in specific questions. The reason behind this finding must be certainly explored by future researches. Preliminary and shallow analysis were performed and failed to give an answer to the question. The non-conformity among questions motivated an idea to increase the general performance by somehow combine all of the approaches, getting the best from each one (explored in the next subsection).

5.7.1 Combining different approaches

The concept of combining different machine learning models in order to minimize error rates and increase generalization was introduced in 1992 by [153] and named *Stacked*

Generalization (also popularly known as *Stacking*). Roughly speaking, the main idea is that taking a final prediction from several different models is better than accounting with only one model. A similar idea was used when creating the Random Forest and other ensemble learning algorithms (as discussed in Section 2.3). The idea is to combine several weak learners in order to create a stronger learner. The *weak learners* can be: 1) models using the same algorithm but training in different parts of data (as RF); 2) models trained in the same data, but with each model using a different machine learning algorithm; 3) a blend of different feature extraction approaches. The last alternative is what is considered *Stacking* in this chapter.

Since its introduction, stacking has been widely employed in different machine learning applications. In ASAG, this technique was specially applied in the 2012 and 2013's competitions, as competitors were seeking for the best possible performance in their models [53, 88, 91, 83, 86, 79, 67]. However, stacking is not restricted to be used in competitions and since 2013 other ASAG works reported its usage in their models [54, 9, 71].

In order to perform experiments using all the different groups of features presented in Table 34, three stacking strategies were considered, along with an *aggregation* approach:

- Voting Classifier (Voting): in this approach, each weak model outputs a prediction for each sample. The final prediction attributed to a specific sample will consist in the mode function applied to the predictions of all weak learners. In other words, each model *votes* for a prediction and the class with majority votes wins. Ties are resolved according to the precedence (from top to bottom, in the other defined in Table 34);

- Hard Stacking (Hard S.): with this strategy, each model outputs its predictions and these values are used as input into a new machine learning model. In other words, the features from the new model will consist in the output of the weak learners;

- Soft Stacking (Soft S.): this approach is very similar to hard stacking. The only difference is that the input for the new model do not consists in the weak models' predictions but rather in the probabilities for each class. In other words, each model will output a specific percentage that a sample belong to each considered class (four in this work: $\{0, 1, 2, 3\}$). Then, these probabilities are used as the features for the new model;

- All features in one feature vector (Together): this is the most simple approach and can not be considered a stacking procedure. It is, however, another possibility for combining all the different feature sets. It works by concatenating all features in only one feature vector and training a single model.

Furthermore, experiments uses five different combinations of the six possible approaches. They consist in the top-k approaches from Tables 34 and 35:

- Top-2: NGRAMS + WEREP;
- Top-3: Top-2 + LEXSIM;
- Top-4: Top-3 + WESIM;
- Top-5: Top-4 + WNSIM;
- Top-6: Top-5 + TXST.

All things considered, the four stacking approaches are combined with the five possible combinations of feature sets, creating 20 possible combinations. The results from the experiments (executed with the same methodology defined in Section 5.1 and using Random Forests (as it is the one that performed better)) are presented in Table 36.

Table 36 – Results from the possible 20 combinations

Top-k/Method	Hard S.	Voting	Together	Soft S.	Mean
2	0,566	0,558	0,566	0,557	0,562
3	0,551	0,568	0,572	0,566	0,564
4	0,539	0,562	0,572	0,577	0,563
5	0,536	0,563	0,579	0,587	0,566
6	0,531	0,552	0,564	0,591	0,560
Mean	0,545	0,561	0,571	0,576	-

The first insight from Table 36 is that the worst approach was hard stacking, the second worst was voting, the best was soft stacking and the second best was the *together* approach. One possible explanation between the performance of the two-best against the two-worst is that the first group uses more information (all features directly or each class' probability) whereas the second group uses less information (only the final predictions).

Another insight from the table is that the number of the k groups did not have a great impact on the mean performance. This is possibly caused due to the fact that the first 2-best approaches already holds most of performance, leaving little for the others to contribute. Nevertheless, the best score was obtained with the combination of all different approaches (top-6). Also, the second and third best scores were obtained by the top-5 row. This points out to the fact that all approaches can contribute for achieving a greater combined score. All things considered, the **best stacking combination** consisted of **Soft Stacking** and by using all the groups (**Top-6**): the **Soft-6** approach.

In order to better measure the efficiency of the soft-6 model, it was compared to the best single-method: ngrams. Table 37 presents the results from both models among

all the 15 questions and their average. The soft-6 model got nine wins against only six from the ngrams model. This means that the soft-6 approach performed better when considering each question. However, a tie happened when averaging all the scores (using the *bk* score).

Table 37 – Comparison between Ngrams and Soft-6

Q_ID	NGRAMS	SOFT-6	Q_ID	NGRAMS	SOFT-6
1	0,612	0,583	**9**	0,484	0,495
2	0,645	0,673	**10**	0,820	0,845
3	0,575	0,592	**11**	0,517	0,529
4	0,775	0,743	**12**	0,592	0,579
5	0,685	0,679	**13**	0,362	0,259
6	0,487	0,464	**14**	0,639	0,667
7	0,364	0,398	**15**	0,808	0,816
8	0,495	0,543	**Mean**	0,591	0,591

In order to break the tie, the accuracy, linear kappa and quadratic kappa of the average among questions were considered, reported in Table 38. Despite the *bk* tie, when considering linear and quadratic kappa alone, soft-6 got a slightly more *balanced* score, that is, the difference between the quadratic and linear kappa of soft-6 is slightly smaller than the same difference in ngrams. Furthermore, the soft-6 model got a value of 0,01 accuracy greater than ngrams, which settles the tie up for the winning of Soft-6. Additionally from the scores, the soft-6 model also got a higher generalization power, as it relies on several different features as opposed to the ngrams model.

Table 38 – Comparison between Ngrams and Soft-6 (part 2)

	Acc	LK	QK
NGRAMS	0,626	0,536	0,646
SOFT-6	0,636	0,541	0,641

All things considered, Table 39 reports all the scores obtained by the best approach of this chapter (Soft-6) among all the 15 questions.

Table 39 – Soft-6 final results

Q_ID	Acc	LK	QK	Q_ID	Acc	LK	QK
1	0,556	0,523	0,643	**9**	0,690	0,442	0,548
2	0,692	0,619	0,727	**10**	0,848	0,806	0,884
3	0,762	0,562	0,622	**11**	0,566	0,482	0,576
4	0,654	0,679	0,807	**12**	0,557	0,524	0,633
5	0,621	0,621	0,736	**13**	0,404	0,239	0,279
6	0,510	0,409	0,520	**14**	0,637	0,601	0,733
7	0,688	0,358	0,439	**15**	0,708	0,760	0,871
8	0,645	0,486	0,600	**Mean**	0,636	0,541	0,641

5.8 Comparison with Human Grading

The human performance at grading short answers was already discussed in Subsection 3.4.6 that reported the Human-Human Agreement (HHA) of literature works and in Subsection 4.4.2 that presented the inter-rater reliability for this work's dataset. In this section, the agreement's scores between human raters and between one of the human raters and the soft-6 model are compared. The scores are reported in Table 40 using linear and quadratic kappa, for all questions where both scores are available (SHA: System-Human Agreement).

Table 40 – HHA vs SHA agreement

Linear Kappa			**Quadratic Kappa**		
ID	**HHA**	**SHA**	**ID**	**HHA**	**SHA**
1	0,40	0,52	**1**	0,57	0,64
9	0,43	0,44	**9**	0,54	0,55
11	0,39	0,48	**11**	0,52	0,58
12	0,37	0,52	**12**	0,50	0,63
Mean	0,40	0,49	**Mean**	0,53	0,60

Considering the average between the four questions, both SHA and HHA and both linear and quadratic kappa scores are within the range of *moderate* agreement (as to [32]'s guidelines, scores between 0.4 and 0.6). Except for question 9, that has very close scores between SHA and HHA, the other questions have a large difference between SHA and HHA. The results reported in Table 40 shows that the SHA scores are higher than the HHA scores. This means that there was more agreement between the soft-6 model and one of the raters than among human raters. This result is not particularly bad or good, but indicates that the model really learned from the scores assigned by human raters to the answers. Therefore, the model learned in such way that it disagrees less than two humans does. Also, it can indicates that the model performed really great or that some human raters misunderstood the assignment criteria (or even both cases mixed).

Regarding the literature results for the same kind of comparison, a considerable amount of them does not even report the HHA agreement. From those who does report it, it is not that common for SHA scores to be greater than HHA scores. However, it is not that unusual as well [71, 52, 60, 66, 59, 92] (more details can be seen in Tables 10 to 15 from Subsection 3.4.6).

6 FINAL CONSIDERATIONS

The automatic grading of short answers is a valuable resource for the improvement of students' evaluations. However, research using Portuguese data is scarce, specially considering works with a great amount of data, suitable for a machine learning approach. Considering this, this work explored the Automatic Short Answer Grading research field by following some steps. The final considerations of this work are divided among the following sections.

Section 6.1 reinforces the delimitation of this dissertation's work and presents some threats to its validity. Section 6.2 presents the final considerations regarding the systematic literature review. Section 6.3 presents a closure about the *Auto-Avaliador CIR* web system. Section 6.4 exposes some lessons learned from the data collection and dataset construction. Section 6.5 highlights the main findings uncovered by the experiments. Then, Section 6.6 presents the contributions of this work for the education area and possible directions for its application. Finally, Section 6.7 ponders on some possibilities for future continuations of this research work.

6.1 Work's Delimitation

The scope of this work consists in: a systematic literature review, development of a web system, the data collection and several experiments on the data. Provided that the work consisted in a broad range of different activities, some of them were not deeply explored and some opportunities for future researches appeared. One of them regards to the *Auto-Avaliador CIR* web system, that was designed without specialists in web development and graphical designers. Also, the usability evaluation could be much more explored. Another aspect that had no further considerations was the human inter-rater reliability. The criteria was simply exposed to the evaluators and the agreement measured.

6.2 Systematic Literature Review

At the beginning of this work, the first step to uncover the Automatic Short Answer Grading field was to look for literature reviews. From the ones that were found, the idea of performing a new, updated and systematic literature review was conceived. The work's objective was to perform a systematic review in the research field of automatic short answer grading with works using a machine learning approach. The first step was to create a research protocol and plan the research methodology. Then, the review was conducted by following the defined systematic guidelines. The final selection resulted in 44 papers and six research questions were answered based on them.

We first explored the data used in ASAG research, revealing the great variety of datasets and its characteristics, from the language to the number of questions, answers, etc. Then, we looked at which natural language processing and machine learning techniques are the most used in the field. After that, we presented the core of the research, how answers are modeled in order to extract features that can predict their scores. Finally, we showed how researchers evaluated their systems and how their works can (or can not) be compared to each other.

All the presented results shows the essence and evolution of ASAG research using machine learning methods. Also, the review's most important contribution relies on its systematic protocol, making it replicable and comprehensive. Furthermore, its main result was to create a mapping of relevant ASAG works for future works' reference. Yet, another important result was to confirm, as previous research suggested, that the ASAG field is heading more and more towards an evaluation era, using publicity available datasets to evaluate new ideas and methods.

6.3 *Auto-Avaliador CIR*

The *Auto-Avaliador CIR* web system was created in order to collect data in an easier and intuitive way from teachers and students. As the system would need to be developed from the start, we decided to evolve the idea and create a system where it would be possible to use it as a virtual learning environment, if demand arise. The idea was to create a collaborative system, where teachers from all places could grade answers and thus create enough basis to train machine learning models to perform further grading (thus the *intelligent* from its name).

Despite the effort, the primary goal for creating the *Auto-Avaliador CIR* prevailed. It was used mainly to collect the data needed to perform the research. However, nothing prevents from using it as a prototype and to develop a better version in the future, that could fulfill the broader aim of the system: to be a tool that can assist teachers and students in their learning process. With all properly set, a student could write his answers in an exam and be practically instantly graded. Provided with proper feedback, the student could know where to put his focus to keep studying right away.

6.4 Data Collection

As the systematic review revealed a lack of research regarding Portuguese data, a new ASAG dataset was planned to be created. It was done by collecting data from the real world, counting with the participation of 659 students, 14 undergraduate students and 13 teachers. The ASAG dataset presented in this work, in the Portuguese language, is the first one, as far as we know, to be made publicly available. The dataset possesses a

reasonable large amount of data, compared to other literature datasets. Its creation was intended to perform experiments using Portuguese tools, in order to test for the generality of another languages' techniques. Furthermore, it is made available so future researches can present and test new models against this dataset, reporting comparable results.

Nevertheless, some considerations must be presented as they are possible threats to the validity of the data:

- The technique used for collecting samples was the Convenience Sampling, which imposes a restriction regarding the generality of the data and the further experiments based on them;
- Aspects that might had an influence in the data quality are the students' age, grade level, city and if their school is a private or public institution;
- The grades assigned by the human graders chosen to be part of the study. They were in the last year of biology faculty, but their low inter-rater reliability indicates some disagreement regarding the criteria or some other difference;
- Another aspect that might influence the results based on this dataset is the type of the questions. They are all from biology and regarding human body topics. Maybe the subject matter can influence in the quality of data and in the experiments;
- Finally, the label distribution is somewhat imbalanced. This can have an influence in the training of a machine learning model and its further predictions.

6.5 Experiments

Counting on a systematic literature review, that identified the most relevant and used ASAG techniques, and on the newly created Portuguese dataset, several experiments were performed. They explored different approaches used on the literature, along with several possible variations for them. The justified and systematic experiments' procedures produced relevant insights that can be used by future researches:

- Despite its simplicity, ngrams is still a very powerful ASAG predictor, even considering other newer techniques (as pointed in the literature and confirmed in this work);
- The idea of considering correct student answers as they were reference answers can a have a huge performance impact in similarity approaches. The impact of this technique must be more explored by future works as the results presented in this work showed very promising results. This idea can possibly hold a deep impact on future ASAG researches;

- The application of Word Embeddings, one of the most recent and sophisticated NLP techniques, showed very promising results. Among the different embeddings' methods, FastText (the newest considered technique) obtained the best performance for this dataset;
- At least for this work's dataset, response-based approaches proven to perform better than similarity-based ones;
- Each experimented approach has its advantages and the use of them all together proved to produce the best possible model;
- Each question reacts very differently to preprocessing and other techniques, making it harder to decide which to keep. The averaged impact of different techniques was often small due to this high intra-variability;
- Compared to other literature works, the dataset presented in this work got a somewhat high inter-rater disagreement.

Concerning the overall effectiveness of the proposed ASAG method, results showed to be satisfactory, with kappa scores indicating between moderate to substantial agreement between our model and human grading.

6.6 Contribution for the Education Field

Even though the work of this dissertation is mainly related to Computer Science and Artificial Intelligence, it is also a work with a direct application in the field of Education. The potential of this research at becoming basis for future researches and products is high. The applicability of the idea to the industry is enormous and could bring a lot of benefits, extensively discussed in the introduction of this work, specially in virtual learning environments in distance education. Also, the results achieved by the experiments are encouraging, signaling that such application as a commercial product is a real possibility.

One possible application of this work could be as a plugin for the Moodle environment. Moodle is an online platform (free software) for distance education, with one of the largest numbers of installations, users and courses, having more than 25 thousand installations, 360 thousand courses and more than 4 millions students over 155 countries. Essentially, is an open and online software for assisting in online courses or as a complementary environment for physical classes. One of the great advantages of Moodle is that it supports plugins, extensions to its features. Considering this support, a plugin for automatically grading student's short answers could be created and coupled into existing systems, being of great value for assisting teachers in the Moodle environment.

6.7 Future Works

A possible direct continuation for this work is to evaluate the final proposed model (named *Soft-6*) against public datasets available in other languages (by performing an adaptation that other datasets from another language requires). Additionally, the experiments performed in this work could present different results when evaluated in a larger Portuguese ASAG dataset. Future researches could collect and publish such bigger datasets.

Regarding a big picture of ASAG, one of the greatest challenges identified in the related literature and in this work is to create a model that can performs consistently good independently of the question. The high performance variability among different questions needs to be investigated. One idea is to explore the *difficultness* of questions and investigate if it correlates with automatic model's performance.

Concerning NLP technologies, the last few years witnessed a huge development in word embeddings. A new representation model called ELMo achieved state-of-the-art results in many NLP tasks. The great novelty of ELMo is to distinguish between different meanings of the same word, requiring the word's context to decide its respective embedding. Beyond the availability of new state-of-the-art word embeddings techniques, another trend is to embed not only words, but also sentences. Sentence Embeddings techniques (such as Quick-Thoughts, InferSent and Google's Universal Sentence Encoder) gives a single representation for a whole sentence, presenting better results than the simple averaging of a sentence's words' vectors. All these new technologies can be applied to ASAG, as it is a specialized task of NLP.

REFERENCES

[1] BURROWS, S.; GUREVYCH, I.; STEIN, B. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, Springer, p. 60–117, 2015.

[2] KLEIN, E.; LOPER, E. *Natural Language Processing with Python.* [S.l.: s.n.], 2009. ISBN 9780596516499.

[3] RASCHKA, S. *Python Machine Learning.* [S.l.: s.n.], 2015. ISBN 9781783555130.

[4] GIRONES, J. *J48 decision tree.* 2010. <http://data-mining.business-intelligence.uoc.edu/home/j48-decision-tree>. Accessed: 2018-07-24.

[5] MATHWORKS. *Confusion Matrix.* 2010. <https://www.mathworks.com/products/demos/machine-learning/confusion_matrix/confusion_matrix.html>. Accessed: 2018-07-24.

[6] SKYMIND. *A Beginner's Guide to Word2Vec and Neural Word Embeddings.* 2016. <https://skymind.ai/wiki/word2vec>. Accessed: 2019-01-15.

[7] JACOMINI, M. A.; PENNA, M. G. d. O. Carreira docente e valorização do magistério: condições de trabalho e desenvolvimento profissional. *Pro.posições*, v. 27, n. 2, p. 177–202, 2016. ISSN 0103-7307.

[8] NASCIMENTO, M. d. G. C. d. A.; SANTOS, J. V. Sessão Especial 05 - Políticas educacionais e currículo: interfaces na educação infantil e ensino fundamental 37ª Reunião Nacional da ANPEd – 04 a 08 de outubro de 2015, UFSC – Florianópolis. 2015.

[9] SAKAGUCHI, K.; HEILMAN, M.; MADNANI, N. Effective Feature Integration for Automated Short Answer Scoring. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 1049–1054, 2015.

[10] HALEY, D. T. et al. Measuring improvement in latent semantic analysis-based marking systems: Using a computer to mark questions about HTML. *Conferences in Research and Practice in Information Technology Series*, v. 66, p. 35–42, 2007. ISSN 14451336.

[11] LIU, O. L. et al. Validation of automated scoring of science assessments. *Journal of Research in Science Teaching*, v. 53, n. 2, p. 215–233, 2016. ISSN 10982736.

[12] PASSERO, G.; FILHO, A. H.; DAZZI, R. Avaliação do uso de métodos baseados em lsa e wordnet para correção de questões discursivas. In: *Brazilian Symposium on Computers in Education (SBIE).* [S.l.: s.n.], 2016. v. 27, p. 1136.

[13] SANTOS, J. C. A. d. et al. Avaliação automática de questões discursivas usando lsa. *Universidade Federal do Pará*, Universidade Federal do Pará, 2016.

[14] ABED. *Censo EAD Brasil 2016 - Relatório Analítico de Aprendizagem a Distância no Brasil.* [S.l.: s.n.], 2016. 157 p. ISBN 9788559724592.

[15] WILLIAMSON, D. M.; XI, X.; BREYER, F. J. A Framework for Evaluation and Use of Automated Scoring. *Educational Measurement: Issues and Practice*, v. 31, n. 1, p. 2–13, 2012. ISSN 07311745.

[16] BUTCHER, P. G.; JORDAN, S. E. A comparison of human and computer marking of short free-text student responses. *Computers & Education*, Elsevier, v. 55, n. 2, p. 489–499, 2010.

[17] PAGE, E. B. The imminence of... grading essays by computer. *The Phi Delta Kappan*, JSTOR, v. 47, n. 5, p. 238–243, 1966.

[18] GAY, L. R. The Comparative Effects of Multiple-Choiche versus short-answer tests on retention. *Journal of Educational Measurement*, v. 17, n. 1, p. 45–50, 1980.

[19] CONOLE, G.; WARBURTON, B. A review of computer-assisted assessment. *Alt-J*, v. 13, n. 1, p. 17–31, 2005. ISSN 0968-7769. Disponível em: <https://journal.alt.ac.uk/index.php/rlt/article/view/983>.

[20] HIRSCHMAN, L. Automated grading of short-answer tests. *IEEE Intelligent Systems, Trends and Controversies section*, v. 15, n. 5, p. 22–37, 2000.

[21] MANNING, C. D.; SCHÜTZE, H. Foundations of statistical natural language processing. *ACM SIGMOD Record*, v. 31, n. 3, p. 37, 2000. ISSN 01635808. Disponível em: <http://portal.acm.org/citation.cfm?doid=601858.601867>.

[22] ALLEN, J. *Natural language understanding*. [s.n.], 1995. v. 224. 372–374 p. ISBN 0805303340. Disponível em: <http://books.google.com/books?id=l4lQAAAAMAAJ{&}pgi>.

[23] MANNING, C. D.; RAGAHVAN, P.; SCHUTZE, H. An Introduction to Information Retrieval. *Information Retrieval*, n. c, p. 1–18, 2009. ISSN 13864564.

[24] MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, ACM, v. 38, n. 11, p. 39–41, 1995.

[25] MOHRI, M. *Foundations of Machine Learning - Book*. [s.n.], 2012. ISSN 026201825X. ISBN 9780262018258. Disponível em: <http://www.cs.nyu.edu/{~}mohri/mlboo>.

[26] BISHOP, C. M. *Pattern Recognition and Machine Learning*. [s.n.], 2007. v. 16. 049901 p. ISSN 1017-9909. ISBN 978-0-387-31073-2. Disponível em: <http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.2819119>.

[27] QUINLAN, J. R. Induction of decision trees. *Machine learning*, Springer, v. 1, n. 1, p. 81–106, 1986.

[28] AWAD, M.; KHANNA, R. *Machine Learning in Action: Examples*. [S.l.: s.n.], 2015. 209–240 p. ISSN 08856125. ISBN 9781617290183.

[29] BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

[30] FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001.

[31] COHEN, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, Sage Publications Sage CA: Thousand Oaks, CA, v. 20, n. 1, p. 37–46, 1960.

[32] LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. *biometrics*, JSTOR, p. 159–174, 1977.

[33] BOUATAY, F.; MHENNI, F. *Use of the cactus cladodes mucilage (Opuntia ficus indica) as an eco-friendly flocculants: Process development and optimization using stastical analysis*. [s.n.], 2014. v. 8. 1295–1308 p. ISSN 17356865. ISBN 2188795008. Disponível em: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20{&}path=ASIN/0596529>.

[34] PÉREZ-MARÍN, D.; PASCUAL-NIETO, I.; RODRÍGUEZ, P. Computer-assisted assessment of free-text answers. *The Knowledge Engineering Review*, Cambridge Univ Press, v. 24, n. 04, p. 353–374, 2009.

[35] CAMACHO-COLLADOS, J.; PILEHVAR, M. T. From Word to Sense Embeddings: A Survey on Vector Representations of Meaning. p. 1–46, 2018. ISSN 1096-9861.

[36] HARRIS, Z. S. Distributional structure. *Word*, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954.

[37] FIRTH, J. R. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, Basil Blackwell, 1957.

[38] DUMAIS, S. T. Latent semantic analysis. *Annual review of information science and technology*, Wiley Online Library, v. 38, n. 1, p. 188–230, 2004.

[39] MIKOLOV, T. et al. Efficient estimation of word representations in vector space. p. 1–12, 2013. ISSN 15324435.

[40] MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 3111–3119.

[41] ROY, S.; NARAHARI, Y.; DESHMUKH, O. D. A perspective on computer assisted assessment techniques for short free-text answers. In: SPRINGER. *International Computer Assisted Assessment Conference*. [S.l.], 2015. p. 96–109.

[42] ZIAI, R.; OTT, N.; MEURERS, D. Short answer assessment: Establishing links between research strands. In: *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. [S.l.: s.n.], 2012. p. 190–200.

[43] VALENTI, S.; NERI, F.; CUCCHIARELLI, A. An overview of current research on automated essay grading. *Journal of Information Technology Education*, Informing Science Institute, v. 2, p. 319–330, 2003.

[44] HASANAH, U. et al. A review of an information extraction technique approach for automatic short answer grading. In: IEEE. *Information Technology, Information Systems and Electrical Engineering (ICITISEE), International Conference on*. [S.l.], 2016. p. 192–196.

[45] KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. 2004, p. 1–26, 2004.

[46] KITCHENHAM, B. Guidelines for performing Systematic Literature Reviews in Software Engineering. 2007.

[47] DE, R.; FALBO, A. Mapeamento Sistemático. 2013.

[48] PETERSEN, K.; FELDT, R.; AL, E. Systematic Mapping Studies in Software Engineering. 2008.

[49] BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, v. 80, n. 4, p. 571–583, 2007. ISSN 01641212.

[50] ZAMBONI, A. et al. Start uma ferramenta computacional de apoio à revisão sistemática. In: *Proc.: Congresso Brasileiro de Software (CBSoft'10), Salvador, Brazil*. [S.l.: s.n.], 2010.

[51] KESHAV, S. How to read a paper. *ACM SIGCOMM Computer Communication Review*, ACM, v. 37, n. 3, p. 83–84, 2007.

[52] ROSÉ, C. P. et al. A hybrid text classification approach for analysis of student essays. *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing - Volume 2*, p. 68–75, 2003.

[53] PETERS, J.; JANKIEWICZ, P. Automated Student Assessment Prize (ASAP) Peters and Jankiewicz. *ASAP '12 SAS Methodology Paper*, 2012.

[54] HIGGINS, D. e. a. Is getting the right answer just about choosing the right words? The role of syntactically-informed features in short answer scoring. *arXiv:1403.0801v2 [cs.CL]*, 2014.

[55] PULMAN, S. G.; SUKKARIEH, J. Z. Automatic short answer marking. *Proceedings of the second workshop on Building Educational Applications Using NLP - EdAppsNLP 05*, v. 1, n. June, p. 9–16, 2005.

[56] SIL, A. et al. Automatic grading of scientific inquiry. *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, p. 22–32, 2012.

[57] ALDABE, I. et al. Supervised Hierarchical Classification for Student Answer Scoring. 2015.

[58] MAKATCHEV, M.; VANLEHN, K. Combining Bayesian Networks and Formal Reasoning for Semantic Classification of Student Utterances. *In Proceedings of the 2007 Conference on Artificial Intelligence in Education*, p. 307–314, 2007. ISSN 0922-6389.

[59] DZIKOVSKA, M. O.; NIELSEN, R. D.; BREW, C. Towards Effective Tutorial Feedback for Explanation Questions: A Dataset and Baselines. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 200–210, 2012.

[60] NIELSEN, R. D. et al. Learning to Assess Low-level Conceptual Understanding. *FLAIRS Conference*, p. 427–432, 2008.

[61] MADNANI, N. et al. Automated Scoring of a Summary-Writing Task Designed to Measure Reading Comprehension. *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, p. 163–168, 2013.

[62] NYE, B. D.; HAJEER, M.; CAI, Z. Improving Classification of Natural Language Answers to ITS Questions with Item-Specific Supervised Learning Method : Classifier Evaluation. *Flairs Conference*, p. 463–468, 2015.

[63] WANG, H. C.; CHANG, C. Y.; LI, T. Y. Assessing creative problem-solving with automated text grading. *Computers and Education*, v. 51, n. 4, p. 1450–1466, 2008. ISSN 03601315.

[64] LEVY, O. et al. UKP-BIU: Similarity and Entailment Metrics for Student Response Analysis. *Second Joint Conference on Lexical and Computational Semantics, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, v. 2, p. 285–289, 2013.

[65] LUO, J. et al. Predicting Student Grade based on Free-style Comments using Word2Vec and ANN by Considering Prediction Results Obtained in Consecutive Lessons. *Conference on Educational Data Mining*, p. 396–399, 2015.

[66] LEE, C. D. et al. Exploring effect of rater on prediction error in automatic text grading for open-ended question. *Proceedings of the 17th International Conference on Computers in Education, ICCE 2009*, p. 462–466, 2009.

[67] HEILMAN, M.; MADNANI, N. ETS: Domain Adaptation and Stacking for Short Answer Scoring. *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval)*, v. 2, p. 275–279, 2013.

[68] SOROUR, S. E. et al. A Predictive Model to Evaluate Student Performance. *Journal of Information Processing*, v. 23, n. 2, p. 192–201, 2015. ISSN 18826652.

[69] SUKKARIEH, J. Z. Using a MaxEnt classifier for the automatic content scoring of free-text responses. *American Institute of Physics Conference Proceedings*, p. 41–48, 2010. ISSN 0094243X.

[70] JIMENEZ, S.; BECERRA, C.; GELBUKH, A. SOFTCARDINALITY: Hierarchical Text Overlap for Student Response Analysis. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, v. 2, n. SemEval, p. 280–284, 2013.

[71] RAMACHANDRAN, L.; CHENG, J.; FOLTZ, P. Identifying Patterns For Short Answer Scoring Using Graph-based Lexico-Semantic Text Matching. *Workshop on Innovative Use of NLP for Building Educational Applications*, v. 10, p. 97–106, 2015.

[72] HOU, W.-J.; TSAO, J.-H. Automatic Assessment of Students' Free-Text Answers with Support Vector Machines. *International Journal on Artificial Intelligence Tools*, v. 20, n. 02, p. 327–347, 2011. ISSN 0218-2130.

[73] BICICI, E.; GENABITH, J. van. CNGL: Grading Student Answers by Acts of Translation. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, v. 2, n. SemEval, p. 585–591, 2013.

[74] ZESCH, T.; HEILMAN, M. Reducing Annotation Efforts in Supervised Short Answer Scoring. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 124–132, 2015.

[75] MOHLER, M.; BUNESCU, R.; MIHALCEA, R. Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, p. 752–762, 2011.

[76] GLEIZE, M.; GRAU, B. LIMSIILES: Basic English Substitution for Student Answer Assessment at SemEval 2013. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, v. 2, n. SemEval, p. 598–602, 2013.

[77] ZHANG, Y.; SHAH, R.; CHI, M. Deep Learning + Student Modeling + Clustering: a Recipe for Effective Automatic Answer Grading. *Proceedings of the 9th International Conference on Educational Data Mining*, p. 562–567, 2016.

[78] MEURERS, D. et al. Integrating parallel analysis modules to evaluate the meaning of answers to reading comprehension questions. *International Journal of Continuing Engineering Education and Life-Long Learning*, v. 21, n. 4, p. 355, 2011. ISSN 1560-4624.

[79] OTT, N. et al. CoMeT: Integrating different levels of linguistic modeling for meaning assessment. *Second Joint Conference on Lexical and Computational Semantics , and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, v. 2, n. SemEval, p. 608–616, 2013.

[80] MAGOODA, A. et al. Vector Based Techniques for Short Answer Grading. *International Florida Artificial Intelligence Research Society Conference Ahmed*, p. 238–243, 2016.

[81] KOUYLEKOV, M. et al. Celi: EDITS and Generic Text Pair Classification. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, v. 2, n. SemEval, p. 592–597, 2013.

[82] SULTAN, M. A.; SALAZAR, C.; SUMNER, T. Fast and Easy Short Answer Grading with High Accuracy. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 1070–1075, 2016.

[83] ZBONTAR, J. Short Answer Scoring by Stacking. *ASAP '12 SAS Methodology Paper*, p. 1–7, 2012.

[84] HORBACH, A.; PALMER, A.; PINKAL, M. Using the text to evaluate short answers for reading comprehension exercises. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, v. 1, p. 286–295, 2013.

[85] ROY, S.; BHATT, H. S.; NARAHARI, Y. An Iterative Transfer Learning Based Ensemble Technique for Automatic Short Answer Grading. v. 285, p. 1622–1623, 2016.

[86] TANDALLA, L. Scoring Short Answer Essays. *ASAP-SAS Methodology Paper*, 2012.

[87] LEEMAN-MUNK, S. P.; WIEBE, E. N.; LESTER, J. C. Assessing elementary students' science competency with text analytics. *Proceedins of the Fourth International Conference on Learning Analytics And Knowledge - LAK '14*, p. 143–147, 2014.

[88] CONORT, X. Short Answer Scoring — Explanation of "Gxav" Solution. *ASAP '12 SAS Methodology Paper*, p. 1–22, 2012.

[89] GOMAA, W. H.; FAHMY, A. A. Arabic Short Answer Scoring with Effective Feedback for Students. *Journal of Computer Applications*, v. 86, n. 2, p. 35–41, 2014. ISSN 09758887.

[90] SULTAN, M. A.; BOYD-GRABER, J.; SUMNER, T. Bayesian Supervised Domain Adaptation for Short Text Similarity. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 927–936, 2016.

[91] JESENSKY, J. Automated Student Assessment Prize: Short Answer Scoring - Team JJJ Technical Methods Paper. *ASAP '12 SAS Methodology Paper*, p. 1–26, 2012.

[92] MOHARRERI, K.; HA, M.; NEHM, R. H. EvoGrader: an online formative assessment tool for automatically evaluating written evolutionary explanations. *Evolution: Education and Outreach*, v. 7, n. 1, p. 15, 2014. ISSN 1936-6434.

[93] MEURERS, D. et al. Evaluating Answers to Reading Comprehension Questions in Context: Results for German and the Role of Information Structure. *Proceedings of the TextInfer Workshop on Textual Entailment*, p. 1–9, 2011.

[94] DZIKOVSKA, M. et al. SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. *Seventh International Workshop on Semantic Evaluation*, p. 263–274, 2013.

[95] VIJAYMEENA, M.; KAVITHA, K. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, v. 3, n. 2, p. 19–28, 2016.

[96] PAPINENI, K. et al. Bleu: a method for automatic evaluation of machine translation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 40th annual meeting on association for computational linguistics*. [S.l.], 2002. p. 311–318.

[97] LIN, C.-Y.; HOVY, E. Automatic evaluation of summaries using n-gram co-occurrence statistics. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. [S.l.], 2003. p. 71–78.

[98] LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. In: *Soviet physics doklady*. [S.l.: s.n.], 1966. v. 10, n. 8, p. 707–710.

[99] WINKLER, W. E. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *ERIC*, ERIC, 1990.

[100] LEACOCK, C.; MILLER, G. A.; CHODOROW, M. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, MIT Press, v. 24, n. 1, p. 147–165, 1998.

[101] WU, Z.; PALMER, M. Verbs semantics and lexical selection. In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. [S.l.: s.n.], 1994. p. 133–138.

[102] LIN, D. et al. An information-theoretic definition of similarity. In: CITESEER. *Icml*. [S.l.], 1998. v. 98, n. 1998, p. 296–304.

[103] RESNIK, P. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[104] JIANG, J. J.; CONRATH, D. W. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[105] LANDAUER, T. K.; DUMAIS, S. T. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, American Psychological Association, v. 104, n. 2, p. 211, 1997.

[106] GABRILOVICH, E.; MARKOVITCH, S. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: *IJcAI*. [S.l.: s.n.], 2007. v. 7, p. 1606–1611.

[107] KOLB, P. Disco: A multilingual database of distributionally similar words. *Proceedings of KONVENS-2008, Berlin*, 2008.

[108] PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543.

[109] ZHANG, C. et al. An up-to-date comparison of state-of-the-art classification algorithms. *Expert Systems with Applications*, Elsevier, v. 82, p. 128–150, 2017.

[110] MOHLER, M.; MIHALCEA, R. Text-to-text semantic similarity for automatic short answer grading. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09*, p. 567–575, 2009.

[111] VILELA, R. F. et al. SCATeDi : Sistema Inteligente para Avaliação de Desempenho Escolar em Avaliações Discursivas. *Workshop de Informática na Escola*, n. 1984, p. 11, 2012. ISSN 2316-6541.

[112] SANTOS, J. C. A. et al. Aplicação de um método LSA na avaliação automática de respostas discursivas. *Revista Brasileira de Informática na Educação*, v. 0, n. 0, p. 10–19, 2012.

[113]SALTON, G. D.; CARNIEL, C. A.; MELLO, B. A. D. Regras sintáticas livres de contexto na correção automática de Unidades de Leitura. p. 217–222, 2013.

[114]ÁVILA, R. L. F.; SOARES, J. M. Uso de técnicas de pré-processamento textual e algoritmos de comparação como suporte à correção de questões dissertativas: experimentos, análises e contribuições. n. Cbie, p. 727–736, 2013. ISSN 2316-6533. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/2551>.

[115]FIGUEIRA, A. D. S. et al. Módulo de Avaliação Automática de Questões Discursivas no Ambiente Virtual de Aprendizagem LabSQL. *Iberian Conference on Information Systems and Technologies (CISTI)*, p. 1–5, 2013. ISSN 21660727.

[116]FLORES, E. M.; RIGO, S. J.; BARBOSA, J. L. V. Um modelo para avaliação automática de respostas textuais com uso de regras linguísticas. *Brazilian Symposium on Computers in Education (SBIE)*, v. 25, n. 1, p. 1153, 2014. ISSN 2316-6533. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/3061>.

[117]PASSERO, G.; FILHO, A. H.; DAZZI, R. Avaliação do Uso de Métodos Baseados em LSA e WordNet para Correção de Questões Discursivas. n. Cbie, p. 1136, 2016. ISSN 2316-6533. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/6799>.

[118]FILHO, A. H. et al. Bloom's Taxonomy-Based Approach for Assisting Formulation and Automatic Short Answer Grading. n. Cbie, p. 238, 2018. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/7978>.

[119]SIROTHEAU, S.; SANTOS, J.; FAVERO, E. Avaliação automática de respostas textuais curtas por similaridades de n -gramas: refinamentos por regressão linear. n. Cbie, p. 1433, 2018. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/8104>.

[120]ALVARADO, J. G. et al. A Comparison of Features for the Automatic Labeling of Student Answers to Open-ended Questions. *Edm*, p. 55–65, 2018.

[121]KOHAIL, S.; BIEMANN, C. Matching , Re-ranking and Scoring : Learning Textual Similarity by Incorporating Dependency Graph Alignment and Coverage Features. *18th International Conference on Computational Linguistics and Intelligent Text Processing. Budapest*, 2017. Disponível em: <https://www.inf.uni-hamburg.de/en/inst/ab/lt/publications/2017-kohail-biemann-cicling.pdf>.

[122]KUMAR, S.; ROY, S. Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading. p. 2046–2052, 2017.

[123]MARVANIYA, S. et al. Creating Scoring Rubric from Representative Student Answers for Improved Short Answer Grading. n. October, 2018.

[124]MIESKES, M.; PAD, U. Work Smart – Reducing Effort in Short-Answer Grading. v. 2018, n. Nlp4call, p. 57–68, 2018.

[125]MULONGO, B.; PIHLQVIST, F. Using Rule-Based Methods and Machine Learning for Short Answer Scoring. 2018.

[126]NAU, J.; Haendchen Filho, A.; PASSERO, G. Evaluating Semantic Analysis Methods for Short Answer Grading Using Linear Regression. *PEOPLE: International Journal of Social Sciences*, v. 3, n. 2, p. 437–450, 2017.

[127]RAO, P. S. B. et al. Automatic assessment of communication skill in non-conventional interview settings: a comparative study. *Proceedings of the 19th ACM International Conference on Multimodal Interaction - ICMI 2017*, v. 17, n. 1, p. 221–229, 2017. Disponível em: <http://dl.acm.org/citation.cfm?doid=3136755.3136756>.

[128]SAAD, M. B.; JACKOWSKA-STRUMILLO, L.; BIENIECKI, W. ANN Based Evaluation of Student's Answers in E-tests. p. 155–161, 2018.

[129]SAHA, S. et al. Sentence level or token level features for automatic short answer grading?: use both. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 10947 LNAI, p. 503–517, 2018. ISSN 16113349.

[130]SMITH, P. A. M. et al. A Multimodal Assessment Framework for Integrating Student Writing and Drawing in Elementary Science Learning. *IEEE Transactions on Learning Technologies*, v. 1382, n. c, p. 1–14, 2018. ISSN 19391382.

[131]SUZEN, N. et al. Automatic Short Answer Grading and Feedback Using Text Mining Methods. p. 1–20, 2018. Disponível em: <http://arxiv.org/abs/1807.10543>.

[132]TACK, A. et al. Human and Automated CEFR-based Grading of Short Answers. *Proceedings of the 12th Workshop on Innovative Use ofNLP for Building Educational Applications*, p. 169–179, 2017. Disponível em: <http://www.aclweb.org/anthology/W17-5018>.

[133]WU, S.-h. et al. A Short Answer Grading System in Chinese by Support Vector Approach. p. 125–129, 2018.

[134]YANG, X. et al. *Automatic Chinese Short Answer Grading with Deep Autoencoder*. [S.l.]: Springer International Publishing, 2018. v. 58. 277 p. ISSN 19409818. ISBN 978-3-642-21868-2.

[135]SAUNDERS, M. N. *Research methods for business students, 5/e*. [S.l.]: Pearson Education India, 2011.

[136]VANBELLE, S. A New Interpretation of the Weighted Kappa Coefficients. *Psychometrika*, v. 81, n. 2, p. 399–410, 2016. ISSN 00333123.

[137]COHEN, J. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, American Psychological Association, v. 70, n. 4, p. 213, 1968.

[138]CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016. (KDD '16), p. 785–794. ISBN 978-1-4503-4232-2. Disponível em: <http://doi.acm.org/10.1145/2939672.2939785>.

[139]BIRD, S.; LOPER, E. Nltk: the natural language toolkit. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. [S.l.], 2004. p. 31.

[140]KINOSHITA, J.; SALVADOR, L. do N.; MENEZES, C. E. D. de. Cogroo: a brazilian-portuguese grammar checker based on the cetenfolha corpus. In: *LREC*. [S.l.: s.n.], 2006. p. 2190–2193.

[141]PASSERO, G. *CoGrOO4Py*. 2016. <https://github.com/gpassero/cogroo4py>. Accessed: 2018-07-24.

[142]PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

[143]ORSINIUM, G. *Text Distance*. 2017. <https://github.com/orsinium/textdistance>. Accessed: 2018-07-24.

[144]HAMMING, R. W. Error detecting and error correcting codes. *Bell Labs Technical Journal*, Wiley Online Library, v. 29, n. 2, p. 147–160, 1950.

[145]RATCLIFF, J. W.; METZENER, D. E. Pattern-matching-the gestalt approach. *Dr Dobbs Journal*, MILLER FREEMAN, INC 411 BOREL AVE, SAN MATEO, CA 94402-3522, v. 13, n. 7, p. 46, 1988.

[146]CILIBRASI, R.; VITÁNYI, P. M. Clustering by compression. *IEEE Transactions on Information theory*, IEEE, v. 51, n. 4, p. 1523–1545, 2005.

[147]ALUÍSIO, S. et al. An account of the challenge of tagging a reference corpus for brazilian portuguese. In: SPRINGER. *International Workshop on Computational Processing of the Portuguese Language*. [S.l.], 2003. p. 110–117.

[148]BOJANOWSKI, P. et al. Enriching Word Vectors with Subword Information. 2016. ISSN 00385298.

[149]HARTMANN, N. et al. Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks. n. Section 3, 2017.

[150]CONNEAU, A. et al. Supervised learning of universal sentence representations from natural language inference data. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017. p. 670–680. Disponível em: <https://www.aclweb.org/anthology/D17-1070>.

[151]BARONI, M.; DINU, G.; KRUSZEWSKI, G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [S.l.: s.n.], 2014. v. 1, p. 238–247.

[152]MIHALCEA, R.; CORLEY, C.; STRAPPARAVA, C. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. 2006.

[153]WOLPERT, D. H. Stacked generalization. *Neural networks*, Elsevier, v. 5, n. 2, p. 241–259, 1992.

Appendix

APPENDIX A – PUBLISHED PAPER - *AUTO-AVALIADOR* CIR

Auto-Avaliador Colaborativo e Inteligente de Respostas

Lucas B. Galhardi[1], **Jacques D. Brancher**[1]

[1]Programa de Pós-graduação em Ciência da Computação –
Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina – Paraná – Brasil

{lucasbgalhardi,jacques}@uel.br

Resumo. *Avaliações são muito utilizadas nos contextos de aprendizagem para verificar quanto conhecimento está sendo retido pelos alunos. Questões discursivas podem avaliar níveis diferentes do aprendizado dos alunos, quando comparadas com questões de múltipla escolha. Entretanto, devido a sua facilidade na correção, questões de múltipla escolha são geralmente mais utilizadas. Visando auxiliar nesse problema e apresentar ao professor uma ferramenta que lhe permita aplicar avaliações com questões discursivas sem receio do tempo de correção, surge o Auto-Avaliador Colaborativo e Inteligente de Respostas, uma ferramenta para a avaliação automática deste tipo de questões.*

1. Cenário de Uso

Avaliações são recorrentemente utilizadas no ambiente de aprendizado para verificar o conhecimento retido pelos alunos. Apesar de sua importância, professores geralmente passam por dificuldades ao avaliar respostas discursivas de salas de aula lotadas. Muitas vezes o trabalho da correção tem que ser levado para casa, comprometendo a qualidade de vida do professor [Jacomini and Penna 2016]. Pesquisas indicam que cerca de 75% dos professores afirmam levar trabalho para casa com frequência, como a correção de atividades [Nascimento and Santos 2015]. Essa situação sobrecarrega o professor e, consequentemente, diminui seu tempo para a preparação de aulas e outras atividades, prejudicando todo o ambiente de aula [Silva and Rosso 2008].

A sobrecarga de trabalho também pode levar o professor a optar por mais questões de múltipla escolha ao invés de questões que exijam uma resposta discursiva. Entretanto, questões discursivas avaliam níveis de aprendizado muitas vezes não contemplados por questões de múltipla escolha [Burrows et al. 2015]. Como são muitos alunos para apenas um professor, se questões discursivas forem utilizadas, ocasionará uma demora para os alunos obterem o *feedback* apropriado. Além disso, quando os alunos finalmente obtiverem sua avaliação, ela pode ser diferente de outro colega que respondeu de maneira muito parecida, pois o cansaço e a própria subjetividade humana podem influenciar a avaliação dos professores [Santos et al. 2016, Passero et al. 2016].

Visando ajudar nesses problemas, surge a correção assistida por computação, que tem como vantagens: formalização dos critérios de avaliação, feedback rápido (tanto para os alunos quanto para os professores) e consequentemente mais tempo disponível para o professor usar em suas outras atividades. Entretanto, avaliar respostas discursivas automaticamente não é uma tarefa simples para o computador e ainda é uma área de pesquisa na computação, com resultados como 70%, 80% e 90% de acurácia, dependendo de vários

fatores [Burrows et al. 2015]. Por isso, em um primeiro momento, essa tecnologia poderia ser aplicada em ambientes virtuais de aprendizagem, em um contexto de feedback rápido para o aluno, e não para sua avaliação final.

O uso de Ambientes Virtuais de Aprendizagem (AVAs) e a Educação à Distância (EAD) vem crescendo bastante no mundo e no Brasil. Em 2016, o Censo EAD.BR contabilizou 561.667 alunos matriculados em cursos regulares totalmente à distância. Além disso, mais quase 3 milhões de brasileiros realizam algum curso livre (não regulamentado) utilizando o EAD e AVAs para sua aprendizagem [ABED 2016]. Diante desses números, fica clara a necessidade de um suporte aos professores para a avaliação automática de exercícios discursivos, retirando o trabalho exaustivo e mecânico da avaliação feita atualmente pelo professor.

É nesse contexto que entra o sistema Auto-Avaliador Colaborativo e Inteligente de Respostas (Auto-Avaliador CIR), um ambiente online para o desenvolvimento das dinâmicas envolvendo questões e respostas em um contexto de aprendizado. Nele, professores podem se inscrever e criarem provas e questões a serem respondidas por estudantes. Por sua vez, estudantes terão acesso às provas e questões cadastradas pelos professores e poderão enviar suas respostas. Professores terão então acesso às respostas dos estudantes e poderão avaliá-las, fornecendo feedback ao aluno, que poderá ver suas notas em seu espaço no site.

O colaborativo da sigla vem do fato de que professores poderão colaborar entre si e com o sistema, principalmente de duas maneiras. A primeira é que um professor pode acrescentar respostas de referência e conceitos às questões de seus colegas, complementando e melhorando as questões. Além disso, professores poderão avaliar as respostas de quaisquer provas e questões do sistema, de forma a contribuir com o *feedback* do aluno.

O inteligente vem do fato de que o sistema, quando possuir respostas e suas respectivas avaliações, suficientes para uma questão, poderá avaliar automaticamente a resposta dos alunos subsequentes. Isso é possível devido a utilização de algoritmos de Aprendizado de Máquina, um dos ramos da Inteligência Artificial. Porém, para o correto funcionamento do sistema, muitas respostas e avaliações são necessárias, por isso quanto mais professores e alunos participarem colaborando melhor o sistema será, em termos de acurácia da nota gerada automaticamente.

2. Desenvolvimento

O processo de desenvolvimento do software foi iniciado com a concepção do relacionamento dos dados, em forma de um diagrama entidade-relacionamento. Após isso veio a decisão da linguagem/*framework* a ser utilizada para a programação, onde Python foi o escolhido.

A escolha de Python para programação web é de certa forma incomum visto que muitas aplicações utilizam PHP, Java ou Javascript. Porém, a escolha é justificada pelas bibliotecas utilizadas para o processamento de linguagem natural e aprendizado de máquina, respectivamente o NLTK [Bird and Loper 2004] e o Scikit-learn [Pedregosa et al. 2011]. Essas bibliotecas são umas das mais importantes para o desenvolvimento em suas áreas e portanto foram escolhidas para implementação do sistema de avaliação automática de respostas.

Visando criar um sistema unificado, foi usado Python para a aplicação web também, utilizando o *framework* Django[1] para um desenvolvimento mais rápido e focado. Para a interface foi utilizado o tradicional HTML e CSS, através do *framework* Bootstrap para agilizar e padronizar as telas, além de tornar o sistema responsivo para o uso em diversos dispositivos. A arquitetura utilizada pelo sistema, banco de dados *MySQL* e dispositivos de acesso pode ser visto na Figura 1.

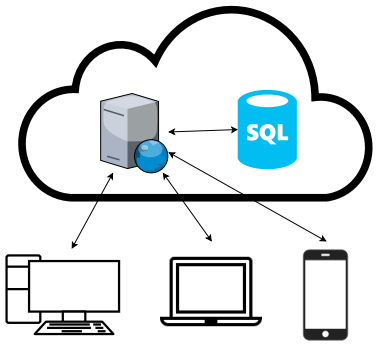


Figura 1. Arquitetura Geral do Sistema

O *framework* Django utiliza o conceito do padrão *Model-Template-View* (MTV), onde a modelagem dos dados fica à cargo da camada *model*, a interface do sistema é feita com *templates* e o *view* é a camada que cuida dos direcionamentos e controle geral da aplicação. Um modelo bem parecido com o muito utilizado *Model-View-Controller* (MVC). O Django utiliza também o mapeamento objeto-relacional, ficando a cargo do programador as classes de modelagem dos dados e o Django se encarrega da comunicação com o banco de dados.

2.1. Avaliação de Usabilidade

Com o objetivo de receber um feedback de usuários que utilizaram o sistema, nós aplicamos o teste de usabilidade *System Usability Scale* (SUS) [Brooke et al. 1996] com um professor e 29 alunos. O teste consiste em 10 perguntas que questionam o usuário sobre a usabilidade do sistema, incluindo sua facilidade de uso ou não, consistência entre outras questões. Ele utiliza a Escala Likert de 5 pontos em suas questões, pedindo ao respondente para responder à uma afirmação com o quanto ele concorda com a mesma, variando de "discordo completamente" até "concordo completamente".

O Auto-Avaliador CIR obteve uma média de 75,1/100 no teste de usabilidade SUS, com as avaliações individuais variando de 42,5 à 97,5. É comumente aceito pela comunidade de usabilidade que a média para esse teste é de 68, portanto uma nota maior que essa média indica que o sistema possui uma boa usabilidade. Como o valor obtido pelo Auto-Avaliador CIR é apenas um pouco maior que a média, ainda há muito o que se melhorar, principalmente em relação ao seu design, como indicado em uma questão de texto livre incluída no questionário.

Além das questões do SUS e a que perguntava sobre o que pode ser melhorado no sistema, fizemos mais uma questão utilizando a Escala Likert, com a seguinte afirmação: *"Eu acho que um sistema de avaliação automática de respostas discursivas é muito útil,*

[1] www.djangoproject.com/

especialmente em um contexto de curso à distância, através de um ambiente virtual de aprendizagem.". Para essa afirmação, apenas um usuário discordou completamente, cinco foram neutros, quatro concordaram e 20 concordaram completamente. Utilizando o valor 1 para discordo completamente até 5 para concordo completamente, o Auto-Avaliador CIR obteve 4,4/5 para essa afirmação, indicando que os usuários majoritariamente concordam que a avaliação automática pode ser muito útil.

3. Apresentação do software

Como já exposto, o objetivo do Auto-Avaliador CIR[2] é fornecer um ambiente virtual para a realização de provas discursivas, a ser utilizado por professores e alunos, de forma totalmente gratuita[3]. O primeiro passo ao se cadastrar no sistema é escolher qual tipo de usuário será a sua conta, como visto na Figura 2. No caso da escolha como professor, é necessário que após o cadastro o usuário envie um e-mail para o administrador comprovando que é um professor, para evitar que qualquer pessoa possa se cadastrar como professor e avaliar respostas de alunos. Para o usuário estudante não há essa necessidade.

Figura 2. Escolha do tipo de usuário

A partir do cadastro, o sistema é dividido em dois, onde professores terão acesso à algumas ações e alunos à outras. Nas Figuras 3 e 6 é possível observar o menu principal de professores e de estudantes, com as ações mais relevantes do sistema.

3.1. Ações do Sistema

A seguir estão descritas as ações mais importantes que podem ser realizadas por usuários do sistema, dividido por usuários professores e estudantes.

Para **usuários professores**:

- **Cadastrar nova prova:** cadastra uma nova prova em branco com seu título e descrição.
- **Criar nova questão:** cadastra uma nova questão à uma prova, junto com possíveis respostas de referência esperadas pelo professor e conceitos envolvidos na questão (como palavras-chave).

[2]www.autoavaliadorcir.com
[3]youtu.be/nacBmterQsk

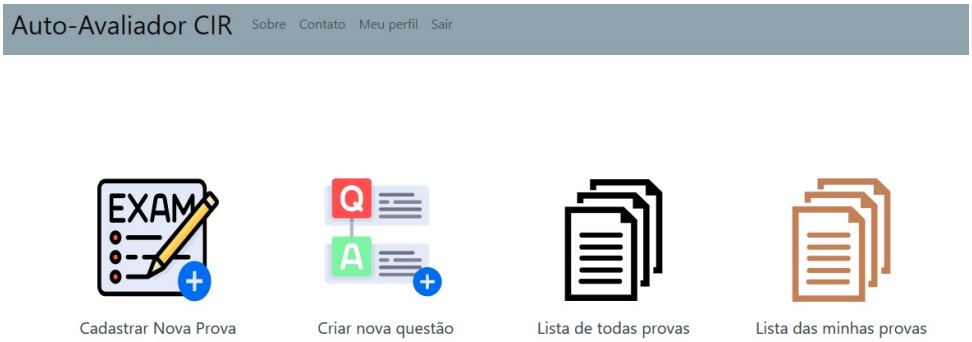


Figura 3. Menu principal do professor

- **Listar todas provas:** lista todas as provas cadastradas para visualização.
- **Listar minhas provas:** lista apenas as provas criadas pelo usuário.
- **Ver questões de uma prova:** mostra todas as questões de uma prova específica, como visto na Figura 4.

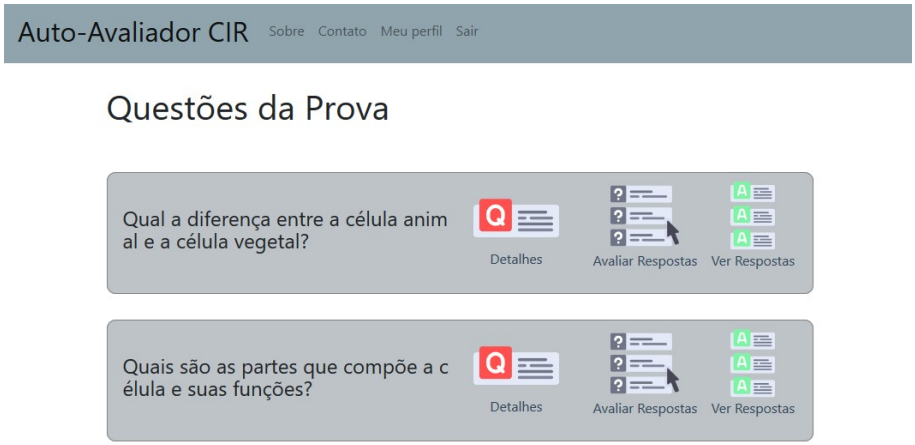


Figura 4. Lista de questões de uma prova

- **Detalhes de uma questão:** mostra todos os detalhes de uma questão, como seu enunciado, respostas de referência e conceitos cadastrados.
- **Ver respostas de uma questão:** mostra todas as respostas dos alunos para uma questão.
- **Avaliar respostas de uma questão:** apresenta uma resposta por vez de uma questão específica para o professor atribuir uma nota. Ele tem a opção de avaliar e terminar ou avaliar e continuar (avaliando a próxima resposta), como visto na Figura 5.

Figura 5. Avaliação de respostas pelo professor

- **Adicionar resposta de referência:** adiciona uma resposta de referência à uma questão a escolha do professor. Diferentes professores podem colaborar com as provas cadastrando suas próprias respostas.
- **Adicionar conceito:** como a ação acima, realiza a adição de mais conceitos envolvidos no contexto da questão.

Para **usuários estudantes**, com menu principal representado na Figura 6:

Figura 6. Menu principal do aluno

- **Ver todas questões ainda não respondidas por mim:** mostra ao aluno todas as questões cadastradas no sistema, para que ele possa escolher qual responder, das que ele ainda não respondeu.
- **Lista de provas:** mostra todas as provas disponíveis.
- **Lista das minhas respostas:** mostra uma lista com todas as respostas já feitas pelo usuário, com o enunciado e sua resposta, junto com a nota dada pelo professor que criou a questão, média da nota de outros professores e a nota dada pelo avaliador automático.
- **Busca por provas ou questões:** mostra um campo para buscar por provas de um professor específico, utilizando seu nome ou seu ID (o ID fica disponível ao professor em seu perfil).
- **Responder questão:** responde apenas uma questão isolada, da lista de todas questões.

- **Responder questões:** responde uma prova completa, com todas as questões que a mesma possui, como visto na Figura 7.

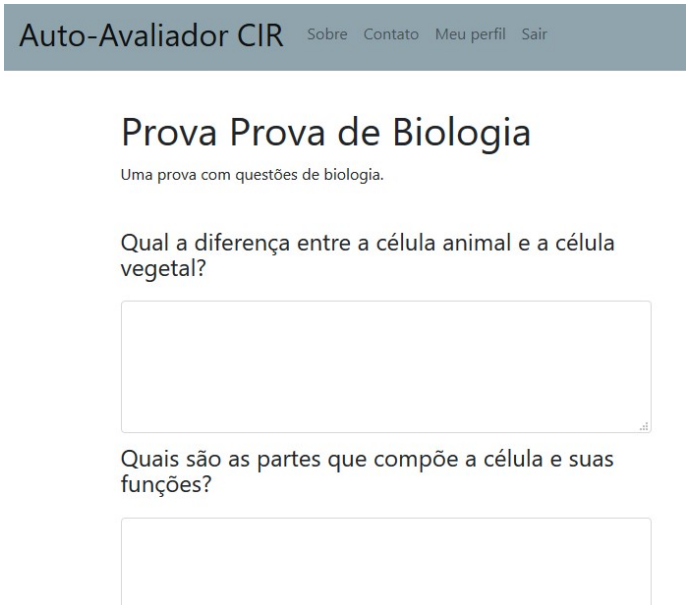


Figura 7. Parte de uma prova de biologia com algumas questões

4. Considerações Finais

Este trabalhou apresentou o Auto-Avaliador CIR, uma ferramenta colaborativa e inteligente para correção automática de respostas discursivas. Ele foi criado com o objetivo de auxiliar professores no processo de avaliação de seus alunos, pois ao realizar o processo automaticamente libera o professor para se dedicar a suas outras atividades, como a preparação de aulas.

Em um primeiro momento, seu uso poderia se dar no contexto de avaliações em que um rápido feedback é necessário, tanto para alunos quanto para os professores. Sua aplicação poderia ocorrer através de cursos online, que já utilizam a estrutura da web para realizar suas atividades. Já acostumados ao contexto virtual, alunos poderiam receber feedback através da resolução de questões discursivas e descobrir o que já está suficiente e o que é necessário estudar mais, de maneira automatizada.

O sistema já foi utilizado por cerca de 245 estudantes do ensino fundamental e atualmente conta com a correção automática apenas para 15 questões de biologia. Porém, seu uso por mais instituições, professores e alunos poderia rapidamente transformá-lo em um produto com avaliações automáticas para diversas questões de várias áreas do conhecimento.

O Auto-Avaliador CIR ainda está em fase inicial de seu processo como produto, mas poderia ser futuramente trabalhado como um *web service*, para permitir fácil integração com outros sistemas de ambientes virtuais de aprendizagem. Porém, já atualmente, um professor poderia começar com sua utilização e quando com um número razoável de respostas por questão fosse atingido, poderia aproveitar de sua avaliação automática, onde o uso contínuo do sistema melhoraria constantemente seu desempenho.

Referências

ABED (2016). *Censo EAD Brasil 2016 - Relatório Analítico de Aprendizagem a Distância no Brasil.*

Bird, S. and Loper, E. (2004). Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.

Burrows, S., Gurevych, I., and Stein, B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, pages 60–117.

Jacomini, M. A. and Penna, M. G. d. O. (2016). Carreira docente e valorização do magistério: condições de trabalho e desenvolvimento profissional. *Pro.posições*, 27(2):177–202.

Nascimento, M. d. G. C. d. A. and Santos, J. V. (2015). Sessão Especial 05 - Políticas educacionais e currículo: interfaces na educação infantil e ensino fundamental 37ª Reunião Nacional da ANPEd – 04 a 08 de outubro de 2015, UFSC – Florianópolis.

Passero, G., Haendchen Filho, A., and Dazzi, R. (2016). Avaliação do uso de métodos baseados em lsa e wordnet para correção de questões discursivas. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 1136.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Santos, J. C. A. d. et al. (2016). Avaliação automática de questões discursivas usando lsa. *Universidade Federal do Pará*.

Silva, G. L. F. and Rosso, A. J. (2008). As Condições Do Trabalho Docente Dos Professores Das Escolas Públicas De Ponta Grossa – Pr. *VII Congresso Nacional de Educação da PUCPR - EDUCERE e no III Congresso Ibero-Americano sobre Violências nas Escolas - CIAVE*, pages 495–536.

Annex

ANNEX A – DATASET QUESTIONS

1. Qual a diferença entre a célula animal e a célula vegetal?
2. Quais são as partes que compõe a célula e suas funções?
3. O corpo humano possui vários tipos de células que se organizam, de acordo com suas especializações e funções, formando os tecidos. Quais são as características do tecido epitelial?
4. Correr, estudar e dançar são atividades que necessitam de muita energia e que podemos realizar graças às nossas células que trabalham sem parar. Como chamamos o conjunto de reações químicas que ocorrem ao nível das células e como ele acontece?
5. Qual a diferença entre fenótipo e genótipo?
6. O que significa transmissão de caracteres hereditários?
7. "O que define o sexo na espécie humana são as características sexuais primárias, ou seja, os órgãos sexuais. Tanto os homens quanto as mulheres têm órgãos sexuais externos e internos. Quais são e quais as funções dos órgãos sexuais internos do homem?
8. Explique o mecanismo de inspiração e de expiração do ar no corpo humano:
9. Em condições normais e estando acordada uma pessoa pode suspender a respiração temporariamente ou acelerar o ritmo respiratório na hora em que desejar fazê-lo. Mas uma pessoa não consegue mesmo que queira, provocar a falta total de gás oxigênio no organismo simplesmente parando de respirar. Por quê?
10. Os cromossomos humanos podem ser estudados em células extraídas do sangue. Em qual das células sanguíneas deve ser feito este estudo. Por quê?
11. Quais são as diferenças entre veias e artérias?
12. Qual é a função do fígado no processo digestivo?
13. Apesar do avanço que a medicina vem apresentando no início do século XXI, ainda nos deparamos com grandes desafios na área da medicina preventiva. Devemos sempre ficar atentos ao calendário de vacinas e às campanhas de vacinação. Quais são as funções das vacinas no corpo humano?
14. Por que é necessário comer vários tipos diferentes de alimentos?
15. O que a hemodiálise faz no corpo humano?

PUBLICATIONS

Works published by the author during the Master's Degree:

Main Publications

1. Lucas B. Galhardi, Cinthyan R. S. C. Barbosa, Rodrigo C. Thom de Souza, Jacques D. Brancher, **Portuguese Automatic Short Answer Grading**, Proceedings of the XXIX Brazilian Symposium on Computers in Education (SBIE), 10/2018, SBC, p. 1373-1382, ISSN 2316-6533, (Qualis CC, B1)

2. Lucas B. Galhardi, Jacques D. Brancher, **Machine learning approach for automatic short answer grading: a systematic review**, Advances in Artificial Intelligence - IBERAMIA 2018. IBERAMIA 2018. Lecture Notes in Computer Science, vol 11238. Springer, Cham, 11/2018, p. 380-391, Online ISBN 978-3-030-03928-8, (Qualis CC, B2)

3. Lucas B. Galhardi, Helen C. M. Senefonte, Rodrigo C. Thom de Souza, Jacques D. Brancher, **Exploring Distinct Features for Automatic Short Answer Grading**, Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), 10/2018, SBC, p. 1-12, (Qualis CC, B4)

4. Lucas B. Galhardi, Jacques D. Brancher, **Auto-Avaliador Colaborativo e Inteligente de Respostas**, Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação (WCBIE 2018), 10/2018, SBC, p. 142-149, ISSN 2316-8889

Complementary Publications

1. Lucas B. Galhardi, Cinthyan R. S. C. Barbosa, João C. Neto, Jacques D. Brancher, **Analisador Léxico-Morfológico de Redações de Estudantes no Estilo do ENEM**, Anais da Conferência Internacional sobre Informática na Educação (TISE) - Nuevas Ideas en Informática Educativa, 11/2018, p. 509-513, ISBN 978-956-19-1111-6, (Qualis CC, B5)