FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Evaluation Metrics for Text and Creation of Writing Tool for Sports Journalism

**Luís Correia**


Mestrado Integrado em Engenharia Informática e Computação

July 21, 2020

# Evaluation Metrics for Text and Creation of Writing Tool for Sports Journalism

**Luís Correia**

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: José Magalhães Cruz
External Examiner: Nuno Escudeiro
Supervisor: Sérgio Nunes

July 21, 2020

# Abstract

Automated journalism is set in the Natural Language Generation (NLG) topic of the field of Natural Language Processing (NLP), which itself derives from Artificial Intelligence. Software related to this area automatically produces human-readable journalistic content using information that is provided, without the intervention of human reporters.

ZOS is a Portuguese company that hosts and creates content for an online sports news site, zerozero.pt, mainly focused on football. Recently, ZOS developed an in-house template-based NLG system, the ProseBot, capable of creating simple football match reports, extracting information about the events of a match from ZOS's large database.

In this dissertation, there are two main objectives. One is developing an evaluation system to assess the quality of both human-crafted texts, produced both by ZOS's newsroom reporters, and computer-generated texts by the ProseBot. For the text quality assessment, an API capable of retrieving metrics and information about a text was developed. This system is divided into three modules: a) one that calculates NLG automatic metrics to evaluate computer-generated reports, b) a module that focuses on delivering text attributes and readability formula scorings, and c) a module that uses Part-Of-Speech tagging and Named Entity Recognition techniques to help the other modules in the assessment process. The second objective is creating a tool to help the writing process of football match reports by using pre-generated text. The developed tool permits the user to select events from a football match, rendering text from the ProseBot, in the form of sentences or paragraphs, including information regarding the selected events. At any time users have the ability to post-edit the generated text to their preferences. Additionally, some functionalities of the metrics system are integrated into this tool.

The metrics system was assessed by inquiring the members of the ZOS's newsroom, through questionnaires, about the importance of the application of text attributes and readability indicators on their reports. The answers registered that reporters classified the implemented metrics as important for journalistic work. To assess the utility of the match report creation tool, reporters from ZOS were interviewed while testing the system. Additionally, a questionnaire was provided to evaluate the importance of the tool's features: a) selecting match events to produce match reports, b) ability to post-edit the generated text, and to study the relevance of the various match events available. A further purpose of this questionnaire was to evaluate the usability of the tool's interface. The results acquired from the interviews and questionnaires showed that reporters find the integrated features helpful for the production of football match reports. In general, the metrics system and the developed tool had a positive appraisal by the members of the ZOS's newsroom, after the user assessment phase. The results indicate that the implemented systems have a beneficial impact on journalistic work.

**Keywords**: Natural Language Generation, Automated Journalism, Sports, Information Systems, Human-Computer Interaction

# Resumo

O Jornalismo Computacional enquadra-se no tópico da Geração de Linguagem Natural (GLN), no ramo do Processamento de Linguagem Natural (PLN), derivado da área da Inteligência Artificial. Software relacionado com o jornalismo computacional tem a capacidade de produzir conteúdo jornalístico, sem a intervenção de jornalistas, utilizando dados que são disponibilizados. A ZOS é uma empresa portuguesa que cria conteúdo para um website de notícias desportivas, zerozero.pt, focado no contexto futebolístico. Recentemente, a ZOS desenvolveu um sistema interno de GLN, baseado em *templates*, capaz de criar pequenos resumos sobre partidas de futebol, extraindo a informação dos seus eventos da grande base de dados da ZOS.

Nesta dissertação existem dois objetivos principais. O primeiro objetivo é desenvolver um sistema de avaliação para a estimar a qualidade de textos, quer produzidos pelos jornalistas da redação da ZOS, quer gerados automaticamente pelo ProseBot. Para a avaliação da qualidade de texto foi desenvolvida uma API capaz de fornecer métricas e informações sobre um texto. Este sistema foi dividido em três módulos: a) um módulo que calcula pontuações de métricas usadas em sistemas de GLN para avaliar as notícias geradas por computador, b) um módulo capaz de oferecer informações sobre os atributos textuais e fórmulas de legibilidade e c) outro módulo que utiliza duas técnicas de PLN: *Part-Of-Speech tagging* e Reconhecimento de Entidades Mencionadas, de forma a ajudar os outro módulos no processamento de texto. O segundo objetivo é a criação de uma ferramenta de ajuda à produção de notícias sobre jogos de futebol, recorrendo a texto previamente gerado. Esta ferramenta permite que o utilizador selecione eventos de um jogo de futebol, produzindo texto gerado pelo ProseBot que contém a informação selecionada, em forma de frases. A qualquer altura, os utilizadores podem editar o texto gerado e adaptá-lo às suas preferências.

O sistema de métricas foi avaliado através de inquéritos à redação da ZOS, tendo como objetivo perceber a importância da aplicação de atributos textuais e indicadores de legibilidade nas suas notícias. As respostas recolhidas mostram que os jornalistas da ZOS classificam a aplicação das métricas desenvolvidas como importantes para o trabalho jornalístico. De forma a avaliar a ferramenta de produção de resumos de futebol, foram também feitas entrevistas a elementos da redação da ZOS enquanto estes testavam o sistema. Além disso, foram usados inquéritos para avaliar a importâncias das funcionalidades da ferramenta: a) seleção de eventos do jogo para produzir texto, b) possibilidade de editar o texto gerado. Com este inquérito estudou-se ainda a importância dos eventos de jogo presentes na ferramenta e estimou-se a usabilidade da sua interface. Os resultados das entrevistas e inquéritos mostram que os elementos da redação da ZOS encontraram grande utilidade nas funcionalidades integradas na ferramenta desenvolvida. Em geral, o sistema de métricas e a ferramenta desenvolvida tiveram um impacto positivo no trabalho jornalístico da ZOS.

**Palavras-Chave**: Geração de Linguagem Natural, Jornalismo Computacional, Desporto, Sistemas de Informação

# Acknowledgments

I would like to thank my family for the amazing support, especially in these times of uncertainty. To my friends with whom I shared many hours over Discord, making this work a little easier. And to my girlfriend who, not only during this period, has shown incredible support, even while working on her own dissertation.

Special thanks to my supervisor Sérgio Nunes for his guidance and insights throughout this process. I would also like to thank Marco Sousa and Pedro Dias from ZOS who, since day one, provided me with all the resources to develop the best work possible.

*"It is better to go forward without a goal, than to have a goal and stay in one place, and it is certainly better than to stay in one place without a goal."*

Andrzej Sapkowski

# Contents

# List of Figures

# List of Tables

# Code Listings

# Abbreviations

| | |
|---|---|
| E2E | End-to-end |
| FRE | Flesch Reading Ease |
| NER | Named Entity Recognition |
| NLG | Natural Language Generation |
| NLP | Natural Language Processing |
| POS | Part-Of-Speech |
| REG | Referring Expression Generation |

# Chapter 1

# Introduction

## 1.1 Context

ZOS is a Portuguese company that hosts and produces content for a sports news site, zerozero.pt, mainly focused on football. Most of its audience is Portuguese, but their system is also prepared for a wider audience as articles are also written in Spanish, English and Brazilian Portuguese.

Besides producing hand-written news, the news outlet developed an in-house system that creates automatically generated football match reports, the ProseBot. The automated journalism area has been evolving and producing pieces of software that save journalists time when writing articles by automatically generating written text. This sort of software relies on a great amount of organised information ready to be extracted and turned into natural language.

A missing characteristic of the ProseBot is the ability to automatically evaluate the quality of the computer-generated texts. The evaluation of Natural Language Generation systems has been growing for the past years, however, it is characterised by some variety, and it is difficult to compare systems directly as the solutions are specific to the problem-based. There are multiple forms of evaluating NLG systems, normally human subjects are involved in this process because, even though human evaluation is not very rigorous, they can offer more than just ratings and provide broader insights about a system. Automatic evaluation of NLG systems is frequently achieved by the use of metrics. This methodology provides fast and cheap results by comparing generated texts against reference texts. The evaluation of text quality can be broadened to the articles written by human elements of the ZOS newsroom. There are metrics used to assess the readability and intelligibility of texts to observe how well readers interpret a report.

## 1.2 Objectives and Motivation

One of the main goals of this dissertation is to develop a system to automatically assess text quality, for both computer-generated and human-written match reports. Until now, the evaluation process of the generated content was done by human ratings with questionnaires to expert writers from ZOS.

ZOS covers the majority of football matches, some of them happening at the same time, which generates data and statistics about the match events. The news outlet has a large database to store all the produced information and make it available to be used in the creation of articles and match reports. That said, the other focal point of this work is to develop a tool that saves writers time through computer-generated match report text rendering based on the selection of events from a football match while making it possible for users to post-edit the match report at any time.

## 1.3 Problem Statement

### 1.3.1 Problem

Online news platforms are accessed by a great number of readers everyday. ZOS hosts and produces content for a sports news website, zerozero.pt, focused on a football-themed context, that produces nearly 60 articles per day. Additionally, ZOS developed a system that automatically generates football reports, the ProseBot. There is interest in finding a way to assess the quality of news written by elements of the newsroom, but also computer-generated reports created by the ProseBot, without human involvement.

On the other hand, ZOS maintains a large database with information and statistics about events of the majority of the football matches. However, most of the time reporters are responsible for the writing of the most important matches, meaning that less relevant ones do not have a published article about them, even though there is available information to do so. ZOS would benefit from an automated approach to tackle this problem without consuming more of the writers' time, but still producing decent content for those matches.

### 1.3.2 Proposed Solution

To assess text quality, we proposed the development of a system that analyses each news article and returns metrics used to evaluate the quality of the text. This system relies on different approaches to extract and display valuable text metrics, which can be integrated into a dashboard.

Along with the mentioned system, an automated helping tool for writing football match reports was developed which will retrieve information from ZOS's database for every match. The novelty of this tool is that users can select match events, and the system will generate paragraphs or sentences about the selected information and include it in the report text. The users will be able to post-edit the generated text and adapt it to their preferences.

## 1.4 Document Structure

This report has 7 main chapters. Chapter 2 contains a definition of natural language generation and its tasks. It also includes a study about the evaluation of NLG systems and the application on real systems. Chapter 3 includes a review of existing readability metrics, how they can be adapted to non-English languages, and their applications. Chapter 4 presents an overview of the system

developed by ZOS, the ProseBot, and information about related systems. Chapter 5 explains the development of the metrics system to assess text quality. It also contains the applications and the user assessment of the system. Chapter 6 describes the development and implementation of a post-editing writing tool that renders text, based on football match event selection. The user assessment of this tool is also shown. Chapter 7 describes the experiments conducted with the NLG automatic metrics and readability indicators, as well as their results. Chapter 8 includes the conclusions and a description of possible future enhancements.

# Chapter 2

# Evaluation of NLG Systems

## 2.1 What is Natural Language Generation?

Reiter and Dale [57] characterise Natural Language Generation (NLG) as "*the sub-field of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce understandable text in English or other human language from some underlying non-linguistic representation of information*". It seems like this definition suits the data-to-text generation better, although text-to-text is too an important instance in the field of NLG. The last mentioned approach takes linguistic content as input and produces, consequently, a new text as output. The application of text-to-text methods can be seen on machine translation systems and text summarisation, for example in the biomedical domain [46].

Sometimes there is a need for a system that takes data, non-linguistic content, as input and convert it into text based on a certain context. Data-to-text generation serves this purpose and has been used in information-rich areas, like Journalism [26, 51, 59]. This approach as been used to develop systems that generate short tailored smoking cessation letters, based on responses to a four-page smoking questionnaire [58], others can produce football summaries, as the ProseBot [59] explained in the Section 4.2.

Automatic football report applications are the perfect example of the need for NLG systems. Although it seems like a niche, there are always fans interested in smaller matches that do not catch the eye of the press. Moreover, a human journalist would rather cover the World Cup final than writing a report about a match from the Portuguese third division. Therefore, an automatic generation of a report for smaller matches would reveal to be a benefit for both fans and writers [73].

There is not a list of rules and complex conditions to define whether a system is a NLG system or not. The definition line of NLG is quite blurry, every system that produces text as output, regardless of the input, context or objective, is indeed a NLG system.

## 2.2 NLG Tasks

The Natural Language Generation process can be divided in several modules to simplify the design of NLG systems. Some authors suggest four essential tasks for NLG systems [57], some others suggest a more complex set, with six steps [26]. Here are the mentioned six steps that can be applied to the majority of systems:

1. **Content determination**: Deciding what information should be communicated to the user;
2. **Text planning**: Deciding how the information should be rhetorically structured;
3. **Sentence aggregation**: Deciding how the information will be split among individual sentences and paragraphs, and what cohesion devices (eg, pronouns, discourse markers) should be added to make the text flow smoothly;
4. **Lexicalisation**: Finding the right words and sentences to express information;
5. **Referring expression generation**: Selecting the words and phrases to identify domain objects;
6. **Surface realisation**: Combining all words and phrases into well-formed sentences in a grammatically correct manner.

The mode of operation of NLG tasks can be illustrated by the design of an I-T-O (Input-Throughput-Output) model, in Figure 2.1 based on Latzer et al. [38].

### 2.2.1 Content Determination

In the content determination process, the developer of the NLG system needs to decide which information should be included in the future generated text, and which should not. In general, there is more information present in the data than the one to be expressed through text. The Content determination phase involves choice, about what will be the communicative intention (e.g. whether it is an informative text or even a narrative) and about what information will be conveyed. For example, in a system that generates football reports, even though the data may contain information about every pass, throw-in or foul, typically there is no need for such detailed information.

### 2.2.2 Text Planning

After deciding what information should be communicated, the NLG system needs the structure of the text. For the football report domain, first, it usually starts with a paragraph with general information about the match: the score, the teams involved in the match, attendance, etc. Then the goals should be described (who scored them and at what time) in the temporally correct order. Finally, a paragraph for the conclusion and the outcome of the match regarding the competition. The end of this process should result in a well-defined structure of paragraphs and sentences for the produced text.

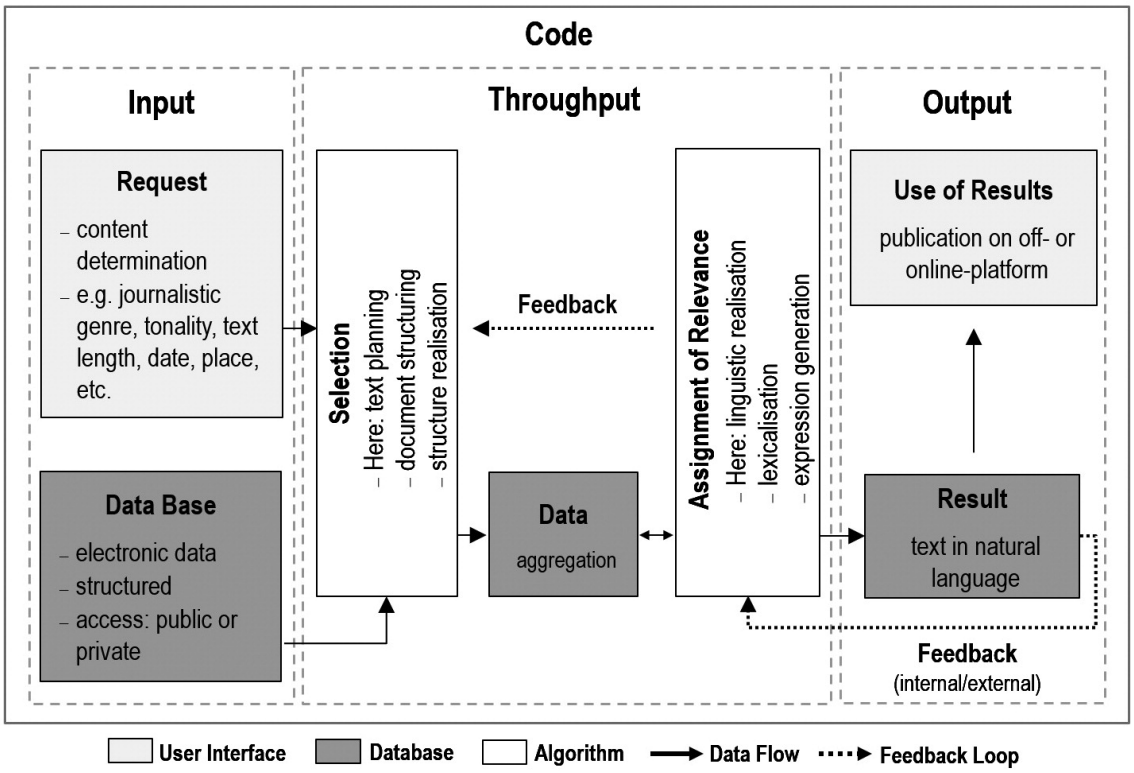



Figure 2.1: I-T-O model for NLG based on Latzer et al. [38].

### 2.2.3 Sentence Aggregation

This is the process by which related messages are grouped together in sentences, so that the text becomes more fluid and readable. Reiter [57] explains the possible operations in Sentence Aggregation:

- Conjunction and other aggregation. For example, converting sentence 1) into 2):

  1. Sami has played for FC Porto. Sami has played for CD Aves. 2. Sami has played for FC Porto and CD Aves.

- Pronominalisation and other reference. For example, converting sentence 3) into 4):

  3. Sami just scored a goal. Sami is phenomenal. 4. Sami just scored a goal. He is phenomenal.

- Introducing discourse markers. For example, converting sentence 5) into 6):

  5. Sami sprained his ankle, he should be substituted. 6. Sami sprained his ankle, *so* he should be substituted.

These operations aim to make the text more human-like and easily readable, without adding or changing information.

### 2.2.4 Lexicalisation

This process reflects an important decision of finding out the word or phrase alternatives to express the messages' building parts. For example, a goal can be expressed as "to put the ball in the back of the net", "to add one into the account", etc. So, this process adds variety to the produced text.

### 2.2.5 Referring Expression Generation

The Referring Expression Generation (REG) is similar to the lexicalisation process, however, this step is when it is determined how entities are referred so that they can be distinguished by the user reading the text.

### 2.2.6 Surface Realisation

Surface realisation is the final step, after all the words and references are agreed upon, there is still a need to combine them and form well-structured sentences. Gatt and Khramer [26] mention the most used approaches for this task:

- templates
- grammar-based systems
- statistical approaches

Templates are an easy way to achieve realisation, however, they can get too restrictive and change between domains. Also, this approach requires a lot of human labour and has a poor scalability in contexts that need considerable linguistic variation. On the other hand, having a template-based system allows full control over the quality of texts produced, avoiding grammatical problems. For example, in a football report system, templates for a substitution can be structured as such:

**$playerout** was subbed out for **$playerin** in the **$minute** minute.

This template has three variables, one for the name of the player that was subbed out, another for the player that entered the pitch and another one regarding the minute of the substitution. When filled with right data, this template can then generate sentences as:

Moussa Marega was subbed out for Fábio Silva in the 67th minute.

Grammar-based systems are an alternative to the template-based approach which is domain-independent. This approach requires hand-crafted grammar rules, that brings some development difficulties in deciding which of the valid outcomes is a better option in a certain context.

Statistical approaches require less manual labour, more adaptable and depend on having accessible historical data. Coverage is typically high, as long as the data needed is available.

## 2.3 Evaluation Methods

As was mentioned in the Section 1.1, evaluation of NLG systems is very subjective and with so many different approaches to different contexts it is difficult to compare systems directly. As Gatt and Khramer [26] note, there are two reasons behind that. The first one is variable input, there is not a default format for input in NLG systems, and normally to compare different systems their input has to be similar. The second reason is the existence of multiple possible outputs. NLG systems deal with a vast amount of output variation, each piece of input can result in a range of possible output results. There are systems whose goal is to produce output variation.

However, there are some common questions to NLG systems: will the objective of the system to be measurable against external criteria objectively, or will it be subjectively evaluated using human judgement?

Hence, the agreement on the methodological distinction of **intrinsic** and **extrinsic** evaluation methods [32]. An intrinsic evaluation measures the performance of a system unrelated to its setup, how the users perceive the system's results. Therefore, attributes like text quality, correctness of output and readability qualify as intrinsic. Extrinsic evaluation analyses the impact of the system on the real world, for example, deploying an NLG system in the real world and measure whether it achieves its desired outcomes, such as changing user behaviour [21].

Although, there are authors [54] that prefer the distinction between:

- **Task-based evaluation** (extrinsic): measurement of real-world impact of a NLG system
- **Human ratings**: human judgement of intrinsic attributes of the output of a system
- **Metrics**: comparing system's output against reference texts

### 2.3.1 Human-based Intrinsic Evaluation

One of the methodologies within intrinsic evaluation relies on human judgements for rating the output of a certain NLG system. The subjects are exposed to the text produced and are asked to rate them following the attributes the system is evaluating, usual criteria are readability, fluency, and adequacy. For example, to evaluate the ProseBot system [59] human ratings were used to judge the quality of the match summaries produced. Professional journalists were invited and during the testing, they were asked to rate (in a 1 to 5 scale) the comprehensibility and the fluency of the texts.

This kind of methodology frequently results in a variation of reliability of the human rating. Gatt and Khramer [26] have interesting points about this issue, for example, if subjects are confined to a predefined scale and rate a text with the lowest rating, if then they come across a text that is to be judged worse then the first one, there is no way of indicating that difference. A related concern emerges from this, if whether it is better to give subject users different objects for them to compare, or just letting the subjects rate the texts in their standards.

### 2.3.2 Metric-based Intrinsic Evaluation

In metric evaluation, human judgement is not involved, instead, there is a comparison of the texts generated by the system against a collection of "gold-standard" reference texts, handwritten texts of high quality. This process can be executed using a variety of different metrics and they can be used to evaluate the coverage of data used for the creation of a text. Metrics are also applied in the rating of sentences concerning attributes as adequacy, fluency (syntactic accuracy) and informativeness. There are additional metrics of interest such as range, evaluating the ability to produce valid variants, and readability.

Here are some of the most used metrics when evaluating NLG systems:

- **BLEU** (Bilingual Evaluation Understudy)
- **METEOR** (Metric for Evaluation of Translation with Explicit Ordering)
- **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation)
- **NIST**
- **CIDEr** (Consensus-based Image Description Evaluation)

To understand how these metrics work, the definition of **n-gram** must be clear. N-grams are a set of co-occurring words within a given window and, when computing the n-grams, typically one word is moved forward, although a greater number of words can be moved in more complex scenarios [30]. For example, for the sentence *"Sami scores a great goal for the away team."*. If **N=2**, resulting in bigrams, then the n-grams would be: [Sami scores], [scores a], [a great], [great goal], [goal for], [for the], [the away], [away team]. So, essentially, just one word moves forward to generate the next bigram. If N=3 was used (trigrams) the n-grams would be: [Sami scores a], [scores a great], [a great goal], [great goal for], [goal for the], [for the away], [the away team].

The number of n-grams in a sentence $K$ with $X$ number of words can be calculated in the following expression: $Ngrams_K = X - (N - 1)$.

In a paper studying the relevance of unsupervised metric in a task-oriented setting, there is a good overview of the metrics [66]. There is also a paper by Wolk et. al [76] that compares the majority of these metrics.

#### 2.3.2.1 BLEU

The BLEU (Bilingual Evaluation Understudy) [49] metric compares n-grams between a candidate expression and a reference expression. This metric was conceived to judge the quality of a machine translation. The primary task is to compare n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches do not depend on position, moreover the best candidate text is the one linked to more matches. It can be used both on single sentences or on multi-sentence test sets. BLEU implements a brevity penalty, applied when the output text has a smaller length than the reference text used.

In his study, Papineni et al. [49] conclude that "*BLEU's strength is that it correlates highly with human judgements by averaging out individual sentence judegment errors over a test corpus rather than attempting to divine the exact human judgement for every sentence*".

BLEU has its limitations as it does not consider various types of errors (substitutions, insertions, synonyms, paraphrase, stems); moreover it has undesirable properties for single sentence uses, this metric is designed to be used at corpus-level [64].

#### 2.3.2.2 METEOR

As it is described Banerjee and Lavie [7], METEOR is an automatic metric for machine translation evaluation that is based on a generalised concept of unigram matching between the machine-produced translation and human-produced reference texts. Once the unigram matching process ends, METEOR computes a score designed to directly capture how well-ordered the matched words are concerning the reference text.

This metric is evaluated by measuring the correlation between metric scores and human judgements of translation quality. Unlike BLEU, METEOR was designed to produce good correlation with human judgement at the sentence or segment level.

#### 2.3.2.3 ROUGE

ROUGE [41] stands for Recall-Oriented Understudy for Gisting Evaluation, and it is a package of metrics created for the evaluation of automatic summarisation systems. It includes measures to automatically determine the quality of a summary comparing it to reference summaries created by humans.

The measures count the number of overlapping units such as n-grams, word sequences, and word pairs between the computer-generated summary and the reference ones. ROUGE has 5 different metrics: ROUGE-N (N-gram Co-Occurrence Statistics), ROUGE-L (Longest Common Subsequence), ROUGE-S (Skip-Bigram Co-Occurrence Statistics) and ROUGE-SU (an extension of ROUGE-S). To evaluate the effectiveness of ROUGE measures, the correlation between ROUGE assigned summary scores and human evaluation scores are analysed.

#### 2.3.2.4 NIST

NIST [17] is a metric resulting from an improvement of BLEU. It finds that the use of less frequent n-grams produces a more informative text, giving it more importance. In this sense, NIST rewards the use of rarer words and has a small brevity penalty than BLEU for smaller variations in the text length. In some cases, NIST shows to be more reliable and to produce higher quality evaluation when compared to BLEU [8].

#### 2.3.2.5 CIDEr

CIDEr (Consensus-based Image Description Evaluation) [74] is a metric to automatically generate image captions. It was one of the automatic metrics used in the E2E NLG Challenge in 2019 [20]. To determine the weight of each n-gram a TF-IDF (Term Frequency Inverse Document Frequency) measure is used, then, using the average cosine similarity of the candidate text against the references, a score for the n-grams of a certain length is computed.

#### 2.3.2.6 Discussion

There is an open discussion about the reliability of automatic metrics. In a paper by Novikkova et al. [48] in which a wide range of metrics are investigated, the authors conclude that state-of-the-art automatic evaluation metrics do not express the effectiveness of manual evaluation. Furthermore, this study found that the disparity between the scale between human and automatic metric evaluation can be at fault for the weak distinguishing of output quality. Also, the action of metrics are very dependent on the system and reference data used. However, automatic metrics can be useful on the system-level as they show where the system is failing (e.g. showing an abrupt variation of the score in some situation).

Ehud Reiter [54, 55, 8, 56] has been very critique of the automatic metric methodologies. He suggests that to use metrics and achieve a good correlation with human ratings, the reference texts need to be very high quality. When evaluating the BLEU metric [55], he came to the conclusion that the correlation "is very dependent on contextual factors". One of the downsides of the use of metrics is that many are (not the word overlapping ones) restricted to the English language.

## 2.4 Application on NLG Systems

After a literature review, there is an overview of recent NLG systems based on different approaches for the text generation process, included in distinct domains and contexts and applying various evaluation methodologies. Table 2.1 exhibits the various systems with the respective purposes, generation approaches, and evaluation methods.

The system by Aoki et al. [5] is a model for the generation of informative stock market comments, using an encoder-decoder architecture. The evaluation of that model was conducted by automatic metrics (BLEU) and human evaluation through the consultation of an expert in finance. The text generated by the model was compared against quality reference texts of real market comments and output of the base model [47]. For human evaluation, the finance expert was asked about the attributes of fluency and informativeness. After the evaluation process, the BLEU score showed a greater value for the original model than for the baseline one (23.6% against 21.88%) [5]. It is common for encoder-decoder systems to use metrics as BLEU to compare the output with other same-architecture systems, as it would very laborious to manually evaluate it.

PASS is a data-to-text template-based system, developed at Tilburg University, that generates football reports from match information for the Dutch league [73]. One of the particular elements

| Name | Purpose | Domain | Year | Generation approach | Evaluation method |
|---|---|---|---|---|---|
| SUMTIME [71] | Weather forecast reports | Weather Forecast | 2003 | Knowledge-based | Human + metrics |
| PASS [73] | football reports | Sports | 2017 | Templates | Human |
| Dong et al. [18] | Generating product reviews | E-Commerce | 2017 | Encoder-decoder | BLEU score |
| Aoki et al. [5] | Comments on stock markets | Finance | 2018 | Encoder-decoder | Human + BLEU score |
| Taniguchi et al. [72] | Generating football commentary | Sports | 2019 | Encoder-decoder | Human+ BLEU score |

Table 2.1: List of recent systems, their purpose, domain, generation approach and evaluation method.

of PASS is that the system creates texts tailored towards fans of one club or the other, observed in the change of tone of the reports. The system is open-source and uses a modular design, any user can add and use extensions. A human-based approach was used to evaluate clarity and fluency and if the tailoring feature is accurately recognised. For the evaluation process, 20 students were invited and shown 20 reports generated by the system. First, they were asked for the fans of which team the report was intended, to assess the tailoring functionality. After, they were asked about the attributes of clarity and fluency. The tailoring functionality was correctly identified in 91% of the cases. Moreover, clarity and fluency showed positive ratings.

SUMTIME is an NLG system that generates weather forecasts texts from numerical weather data. This system is knowledge-based, while it has an informed dataset, the rules are human-crafted [71]. In a recent study [8], this system was subjected to an experiment to determine how well a range of automatic metrics would correlate with human evaluation. For this experiment, 21 forecast dates and reference texts written by specialists were used. For automatic evaluation, the chosen metrics were BLEU (BLEU-4), NIST (NIST-5) and ROUGE (ROUGE-4) to assess systems and texts. String Edit (SE) distance [60] was used as a baseline. As for human evaluation, 9 experts and 21 non-experts were recruited. They were asked to rate from 0 to 5 the texts generated by the system and the corpus, based on the attributes of readability, clarity and general appropriateness. The results for every system of SUMTIME are present in Table 2.2:

The SUMTIME system was designed to deviate from corpus because, some times, it produces better content by itself, considering human evaluation. The authors concluded that "deviating from the corpus in such a way decreases the system's score under corpus-similarity metric". This conclusion relates to metrics designed for machine-translation, as BLEU, in which rarely there is a translation automatically produced that has the level of quality of a reference translation. In conclusion, the authors recognised the NIST metric as the one with the highest correlation with

| System | Experts | Non-experts | NIST-5 | BLEU-4 | ROUGE-4 |
|---|---|---|---|---|---|
| SUMTIME-Hybrid | 0.762 | 0.77 | 5.985 | 0.192 | 0.582 |
| pCRU-greedy | 0.716 | 0.68 | 6.549 | 0.315 | 0.673 |
| SUMTIME-Corpus | 0.644 | 0.736 | 8.626 | 0.569 | 0.835 |
| pCRU-roulette | 0.622 | 0.714 | 5.833 | 0.156 | 0.571 |
| pCRU-2gram | 0.536 | 0.65 | 5.592 | 0.223 | 0.626 |
| pCRU-random | 0.484 | 0.496 | 0.296 | 0.075 | 0.464 |

Table 2.2: Evaluation scores for different SUMTIME systems adapted from Belz et al. [8].

expert opinion.

The work by Dong et al. [18] is an encoder-decoder system that creates models to generate product reviews aiming different attributes (rating, user, etc). This system uses an attribute encoder and a sequence decoder, alongside an attention mechanism. The dataset used was based on the Amazon book reviews and respective metadata. Just like in the football match commentary generation system, the BLUE metric (BLEU-1 and BLEU-4) is used to assess the different models. The results are shown in Table 2.3.

| Method | BLEU-4 (%) | BLEU-1 (%) |
|---|---|---|
| Rand | 0.86 | 20.36 |
| MELM | 1.28 | 21.59 |
| NN-pr | 1.53 | 22.44 |
| NN-ur | 3.61 | 26.37 |
| Att2Seq | 4.51 | 30.24 |
| Att2Seq+A | **5.03*** | **30.48*** |

Table 2.3: Evaluation results for the different methods used adapted from Dong et al. [18], the values marked with an * indicate the highest scoring ones.

In this case, the BLEU metric was used to evaluate the performance of different baseline methods. The authors conclude that the model with the attention mechanism (Att2Seq+A) is the best performing one, based on the BLEU-1 and BLEU-4 highest score. It should be noticed, that even though the BLEU metric was originally created for purposes of machine-translation, this metric is widely used to assess system internal performance, mostly on ML-based systems.

The work developed by Taniguchi et al. [72] is a data-to-text system, developed in the Tokyo Institute of Technology, capable of generating commentary for English Premier League games. For text generation, the system uses an encoder-decoder approach with an attention mechanism and placeholder reconstruction. In the evaluation process of the system, both human evaluation and automatic evaluation were conducted. BLEU was the metric chosen for automatic evaluation, used against reference texts not restricted to commentaries, which can be a problem in this process. Therefore, human evaluation was used to compensate for the possible accuracy problems of BLEU. Ten subjects were asked to rate the text on a scale between 1 and 3 for grammaticality

and informativeness. BLEU scores were divided based on the length of the evaluating text (equal or shorter than 10, 15 and 20). After the experiment, the information shows that the BLEU score decreases with the increase of the length of the text.

# Chapter 3

# Readability Metrics

## 3.1 Flesch Reading Ease

The Flesch Reading Ease Score (FRE) is a metric that determines the intelligibility and readability of texts, introduced by Rudolph Flesch in the early 1940's. The final score ranges between 0 and 100, higher values indicate better legibility, intelligibility and readability. For the English language, the Flesch Reading Score is calculated as follows [61]:

$$206.835 - 1.015\frac{totalwords}{totalsentences} - 84.6\frac{totalsyllables}{totalwords}$$

Since the metric scores text readability between 0 and 100, readability levels were created and mapped to the corresponding Flesch readability scores. In the following Table 3.1, there is a representation of how scores and readability levels are mapped:

| Flesch Reading Ease Score | Reading Difficulty | Example of Style |
|---|---|---|
| 91-100 | Very easy | *Readers's Digest* |
| 81-90 | Easy | *Time* |
| 71-80 | Fairly easy | *US News* |
| 61-70 | Standard | *New York Times* |
| 51-60 | Fairly difficult | "The Ambassadors", by Henry James |
| 31-50 | Difficult | Corporate annual report |
| 0-30 | Very difficult | Legal contract |

Table 3.1: Mapping of the Flesch Reading Ease scores adapted from Finn's article [23].

Initially, this metric was aimed towards assessing legibility in educational texts, as in school books; nowadays it is used in more diverse contexts. We can find applications of the Flesch Reading Ease to assess textual content in websites and apps and also in text editing software. For example, *Microsoft Word* provides a built-in tool to display Flesch Reading scores.

## 3.2 Flesch-Kincaid Grade Level

The Flesch-Kincaid Grade Level originated from a recalculation of the Flesch Reading Ease formula conducted by the US Navy in 1975. The aim was to change the resulting scores value and convert it to values that had immediate impact in real life situations.

Readability metrics are frequently used in the education area, so the "Flesch-Kincaid Grade Level Formula" was introduced, presenting a score in the form of a U.S. education system grade level. For the English language, the grade level is computed with the following formula [36]:

$$0.39\frac{totalwords}{totalsentences} + 11.8\frac{totalsyllables}{totalwords} - 15.59$$

Rather than having a table representing the mapping between readability levels and FRE scores, the results are equivalent the grade level of education that the reader would require to be able to understand the text. For example, if a text scores a Flesch-Kincaid level of 8 then people with at least the eighth grade of education, around 13 to 14 years old, should comprehend it.

## 3.3 Gunning Fog Index

The Gunning Fog Index was introduced in 1952, by Robert Gunning. The metric first appeared in his book *The Technique of Clear Writing* [29]. Newspapers and popular magazines were the main environments for the testing and creation of the Gunning Fog Index, the author claimed that newspapers were full of "fog" and were unnecessarily complex. The Gunning Fog index gives a score in a range from 0 to 20, typically. The value of the score corresponds to the education grade that the reader should have to understand the text on the first reading, much like Flesch-Kincaid Grade Level, mentioned in section 3.2. This readability metric uses the following formula:

$$0.4(\frac{totalwords}{totalsentences} + \frac{complexwords}{totalwords})$$

An important aspect of the Gunning Fog Index is identifying the complex words of a text. The procedure, for the English language, is to count the number of words of three or more syllables that are not:

- proper nouns
- hyphenated easy words
- two-syllable verbs with -ed, -es or -ing suffix

However, the assumption that all multi-syllabic words are difficult to read, even using the filtering procedure mentioned, is one of the major flaws of this metric. Another aspect is that the formula is just suited for passages with more than 100 words.

## 3.4 Automated Readability Index

The Automated Readability Index, commonly known as ARI, is a readability metric used to assess the level of understandability of a text, introduced by Smith and Senter in 1967 [63]. This formula, as some mentioned earlier, outputs a value corresponding to the education grade level. The ARI metric measures word length and sentence length, the difference being that the word length is calculated based on the number of characters rather than the number of syllables. Here is the formula to calculate the Automated readability Index:

$$4.71\frac{totalcharacters}{totalwords}+0.5\frac{totalwords}{totalsentences}-21.43$$

## 3.5 Coleman-Liau Index

The Coleman-Liau Readability Formula was designed by linguists Meri Coleman and T.L. Liau and introduced in 1975. Like other popular readability metrics, the Coleman-Liau Index approximates the minimum U.S. education grade level to comprehend a certain text. Like the Automated Readability Index (section 3.4), this grade-level predictor relies on the number of characters to calculate the word length, as it would be easier for characters to be counted by a physical optical scanning device used in the 1970's. Moreover, the author adds that "word length in letters is a better predictor of readability than word length in syllables" [14]. The original formula for the Coleman-Liau Index is:

$$0.0588L-0.296S-15.8$$

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

## 3.6 Dale-Chall and New Dale-Chall

Inspired by Rudolph Flesch and his metric Flesch Reading Ease, Edgar Dale and Jeanne Chall created the Dale-Chall Readability Formula. However, unlike other metrics this formula assess word difficulty using the number of "hard" words. These words are all that do not appear in a designed list of common words for the English language defined in the paper *"A formula for predicting readability"* written by Dale and Chall in 1948 [15]. This list resulted from a survey involving fourth-grade students in which they identified words that were familiar to them. The original list had 763 words that 80% of the 4th-graders determined as familiar.

Later, in 1995, the Dale-Chall readability formula gets a rework, becoming the New Dale-Chall readability formula and expanding the list from 760 to 3000 familiar and easy to read words [13]. To compute this readability metric, the following formula is used:

$$RawScore = 0.1579*\%difficultwords+0.0496\frac{totalwords}{totalsentences}$$

If the percentage of difficult words is greater than 5% then the score should be adjusted by:

$$AdjustedScore = RawScore + 3.6365$$

If not, the raw score will be the final adjusted score. Finally, Table 3.2 is used to assess the adjusted grade level.

| Adjusted Score | Grade Level Readability Ease |
|---|---|
| 4.9 or lower | 4th-grade student or lower |
| 5.0 to 5.9 | 5th or 6th-grade student |
| 6.0 to 6.9 | 7th or 8th-grade student |
| 7.0 to 7.9 | 9th or 10th-grade student |
| 8.0 to 8.9 | 11th or 12th-grade student |
| 9.0 to 9.9 | 13th or 15th-grade college student |
| 10 or above | 16th-grade college graduate student or above |

Table 3.2: Dale-Chall adjusted grade level conversion.

The most useful aspect is that the New Dale-Chall formula takes word familiarity into account which allows to target word difficulty in different contexts where it can be used by adding specific words to the list.

## 3.7 SMOG

The SMOG (Simple Measure of Gobbledygook) metric appeared in 1969 through the article *"SMOG Grading - a New Readability Formula"* by G. Harry McLaughlin. The author claimed that this method would assess the readability of texts quicker than other previous formulas. Much like the Gunning Fog index (Section 3.3), SMOG also takes the number of polysyllabic words into account. The SMOG formula is designed for a sample text of 30 words and there are some few steps to apply this method [44]:

1. Count 10 consecutive sentences near the beginning of the text to be assessed, 10 in the middle and 10 near the end.

2. In the selected 30 sentences find and count every polysyllabic word (3 or more syllables), even if it appears more than once.

3. Estimate the square root of the number of polysyllabic words counted.

4. Add 3 to the last square root estimated value, giving the reading grade level.

Resulting in the following formula:

$$SMOGgrade = 3 + \sqrt{polysyllablecount}$$

Also, McLaughlin introduced some additional premises to his method:

- Count as a sentence any string of words ending with a period, question mark or exclamation point, semi-colon does not count as sentence-ending punctuation.
- Hyphenated words are considered a single word.
- Numbers that are written should be counted if polysyllabic if presented in a numeric form they should be count based on pronunciation.
- Abbreviations should be pronounced as unabbreviated to determine if they are polysyllabic.
- Proper nouns should be considered

If the assessed text has less than 30 sentences the steps to follow are different:

1. Count the number of sentences.
2. Count the number of polysyllables.
3. Divide the number of sentences in the text into 30 (e.g. a text that has 20 sentences: 30/20 = 1.5).
4. Multiply the number obtained in step 3 by the numbers of polysyllabic words (step 2).
5. Lookup the grade level in the SMOG conversion table 3.3.

| Polysyllabic word count | Approximate Grade Level (+/- 1.5) |
|---|---|
| 0 - 2 | 4 |
| 1 - 6 | 5 |
| 7 - 12 | 6 |
| 13 - 20 | 7 |
| 21 - 30 | 8 |
| 31 - 42 | 9 |
| 43 - 56 | 10 |
| 57 - 72 | 11 |
| 73 - 90 | 12 |
| 91 - 110 | 13 |
| 111 - 132 | 14 |
| 133 - 156 | 15 |
| 157 - 182 | 16 |

Table 3.3: SMOG Conversion Table.

## 3.8 Fry Graph

The Fry Graph Readability Formula is a popular metric developed by Edward Fry and it first appeared on the article *"A Readability Formula That Saves Time"* in 1968 [25]. Fry assumes that texts containing shorter sentences and words with less syllables become more readable. The metric estimates the required grade level of a reader as follows:

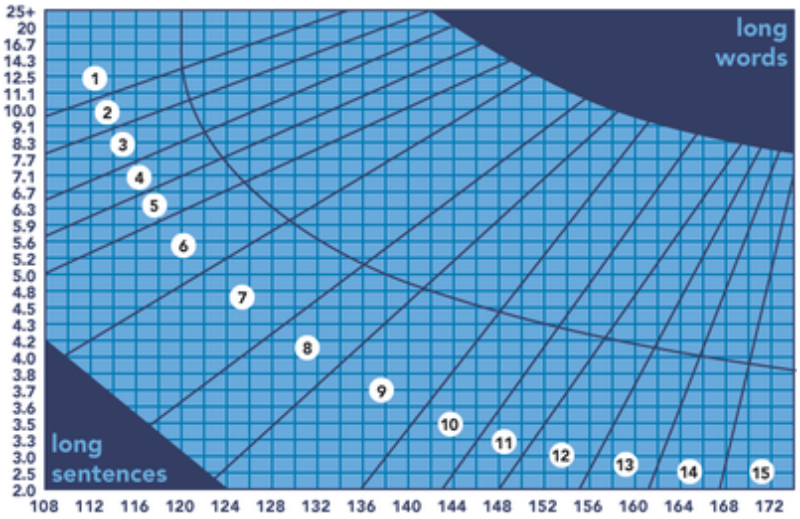



Figure 3.1: Vector graphic based on Fry Graph readability formula adapted from Fry's work [25].

1. Select a sample of 100 words that properly represents the assessed text.
2. Count the number of sentences and estimate the fraction of the last sentence to the 1/10 if it does not divide accurately into a perfect number of sentences.
3. Count the number of syllables of the sample.
4. Inspect the graph (Figure 3.1) and plot dot where the lines intersect: x-axis being the number of sentences and the y-axis the number of syllables in a 100-worded sample. This method shows the section corresponding to the readability grade level of the whole text.

Note that to reach a more accurate score a higher number of samples should be processed. The recommendation is to use three randomly chosen 100-worded samples.

## 3.9 Raygor Estimate Graph

The Raygor Estimate Graph is a readability tool introduced by Alton L. Raygor, in 1977 [52]. This metric measures the average number of sentences and letters in a 100-worded sample. Its usage is similar to the Fry formula (Section 3.8): select a sample of 100 from the text; count the number of sentences and estimate a value if there is not a perfect number of sentences within 100 words

(i.e. half sentence = 0.5); count the number of words with six or more characters. Finally, plot the obtained results on the graph (Figure 3.2), in which the y-axis is the average number of sentences and x-axis the average number of word with more than 6 letters, the intersection will give a grade level ranging between 3 and 14.

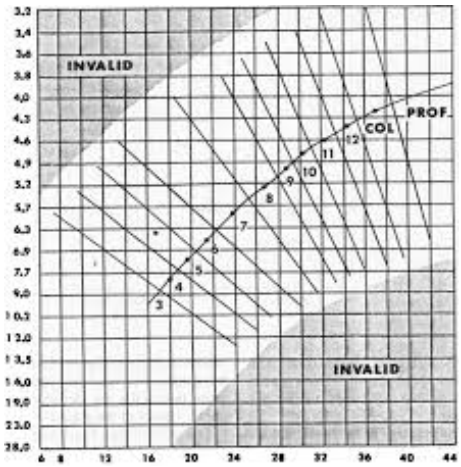



Figure 3.2: Raygor Estimate Graph from Baldwin et al. [6]

## 3.10 FORCAST

The FORCAST formula was developed in 1973 by John S. Cyalor, Thomas G. Sticht, and J. Patrick Ford and it first was published in the article *"Literacy Discussion"* [12].

Unlike other metric, FORCAST is typically used for multiple-choice quizzes and forms, rather than running text. This indicator outputs a readability grade level using the following formula:

$$GradeLevel = 20 - (N \div 10)$$

where N is the number of monosyllabic words in the sample text. The corresponding age to read can be obtained by:

$$AgetoRead = 25 - (N \div 10) years$$

## 3.11 SPACHE

G. Spache created the Spache Readability Formula, through the article *"A New Readabiility Formula for Primary-Grade Reading Materials"*, published in 1953. This metric aims to assess readability for third-grade level texts or below and like the Dale-Chall Readability Formula, it calculates the grade level based on sentence length and number of unfamiliar words. If the sample

text is targeted to a more advanced audience (over 4th-grade level), it is recommended to use the Dale-Chall [69].

The SPACHE formula is the following for a sample text of 100 to 150 words extracted from the original text:

$$0.141 \times \frac{totalwords}{sentences} + 0.086 \times \%unfamiliarwords + 0.839$$

The unfamiliar word are all of those not appearing in the Revised Spache Word List [70].

## 3.12 Adaptation to Non-English Languages

Readability is an important trait that makes text clearer and more accessible to its readers. However, even though accessibility is an important aspect of readability, every predictor mentioned was originally designed for the English language. Naturally, metrics were introduced for non-English languages with different attributes and complexities as the Arabic language for which AK Al Tamimi et al. developed an automatic readability index, in 2014 after analysing 1196 Arabic texts [2]:

$$AWL \times 4.414 - 13.468$$

where AWL is the average number of characters per word.

One of the most popular non-English readability formulas is LIX (Lasbarhetsindex) which was created in 1968 by the Swedish Carl-Hugo Björnsson. Like the Flesch formulas, this metric measures the average number of sentences and number of long words (more than 6 characters) and can be computed by the following formula [9]:

$$LIX = \frac{words}{periods} + \frac{(longwords \times 100)}{words}$$

This metric is more commonly applied to Swedish or Danish. Table 3.4 helps to interpret the LIX scores:

| Text difficulty | LIX Score |
|---|---|
| Very easy | 20-25 |
| Easy | 30-35 |
| Medium | 40-45 |
| Difficult | 50-55 |
| Very difficult | 60 |

Table 3.4: Interpretation of LIX scores (originally for Swedish texts), adapted from [53].

Moreover, there was also an adaptation of the existing readability formulas to non-English languages. In 1980, B. Gilliam et al. adapted the Fry Graph readability formula for the Spanish

language to measure textbooks at a primary level [27]. The Fernandez Huerta Index is a popular readability indicator for the Spanish language, introduced in 1959, which resulted from an adaptation of the Flesch Reading Ease formula. This metric is still widely used for Spanish texts, following the formula [22]:

$$206.84 - (0.60 \times syllables) - (1.02 \times SW)$$

where SW is the numbers of sentences in a sample of 100 words.

The FRE metric was also adapted to the French language, creating the Kandel & Moles Index introduced in 1959. This index uses the formula below [34]:

$$207 - 1.015Lp - 0.736Lm$$

where Lp and Lm are respectively the average number of words per sentence, and syllables per word.

Considering that ZOS and their website, www.zerozero.pt, hosts articles that mainly target a Portuguese audience, let us take a look into the existing readability methods for the Portuguese language. In 2019, Hélder Antunes and Carla Teixeira Lopes published an article that evaluates the adequacy, for the Portuguese language, of readability indicators originally created for the English language. In this study, five different English readability metrics are considered: SMOG, Flesch-Kincaid, ARI, Coleman-Liau and Gunning Fog; all of them output a grade level [4].

Firstly to evaluate the difference of the application of these metrics for the two languages (English and Portuguese) the authors used ten parallel corpora from movies subtitles to PHP language documentation. For this phase, using the collected parallel corpora, the original readability metrics were applied to both languages. After this process, the authors found that readability predictors that measure either the number of syllables in a word or the number of complex words per sentence gave higher grade-level scores for the Portuguese language, meaning a lower readability value. Yet, readability indicators that calculate word length using letter count, as ARI and Coleman-Liau, output similar grades.

For a second phase of the study, the authors gathered 65 school books used in the Portuguese education system and analysed them for particularities that would affect the output of traditional readability metrics. It was concluded that the Portuguese language tends to have longer words, with more syllables. So, for this language, the concept of a difficult word, used in the Gunning Fog Index (section 3.3), would be a word with 4 or more syllables, rather than 3 or more syllables as it is considered for English samples.

Finally, by performing a multiple linear regression for the dataset containing the Portuguese school books, an adaptation to the traditional indicators was proposed, that would have a better correlation to texts in the Portuguese language. The adapted readability metrics are presented in Table 3.5.

| Metric | Formula |
|---|---|
| SMOG | $16.830 \times \sqrt{CW \times 30 \div SE} - 23.809$ |
| Flesch-Kincaid | $0.883 \times WO \div SE + 17.347 \times SY \div WO - 41.239$ |
| ARI | $6.286 \times CH \div WO + 0.927 \times WO \div SE - 36.551$ |
| Coleman-Liau | $5.730 \times CH \div WO - 171.365 \times SE \div WO - 6.662$ |
| Gunning Fog | $0.760 \times WO \div SE + 58.600 \times CW \div WO - 12.166$ |
| CH - characters, CW - complex words, SY - syllabes, WO - words, SE - sentences | |

Table 3.5: Adjusted Portuguese readability metrics. Table adapted from by Hélder Antunes and Carla Teixeira Lopes, 2019 [4].

## 3.13 Real applications

Last sections explained how readability methods were introduced and how the formulas should be enforced. The majority of those metrics were created for assessing the readability of texts for the students in the U.S. education system, however, the potential of their application goes beyond that context. Different areas have special attention to the accessibility of the texts accessed by readers whom should comprehend them fairly easily. Readability evaluation can be found in studies as: classification of spam emails [65], assessment of survey question difficulty [40], U.S. Supreme Court brief analysation [42], readability of special education safeguard documents [43], readability of 4th and 5th-grade textbooks [31], adequacy of readability for security policies [3], difficulty of reading academic texts [68], and readability of tweet, SMS and chats [16].

In healthcare, there is a special focus to make patients understand educational materials about conditions that may affect them. Therefore, a lot of medical studies test the readability of informative texts to the general public, specially for online articles. In 2017, Akhil Kher et al. released an article about the readability assessement of online educational material about congestive heart failure. In this study, the authors adopted six different readability metrics: Flesch-Kincaid, Gunning Fog Index, Coleman-Liau, SMOG and Flesch Reading Ease. A Google search query ("congestive heart failure") was used to filter 70 out of 100 resulting websites and use them as a dataset. Only five out of the all collected websites were in compliance to the acceptable readability grade (6th-grade level). The mean scores were: Flesch-Kincaid (9.79), Gunning Fog Index (11.95), Coleman-Liau Formula (15.17), SMOG Index (11.39) and Flesch Reading Ease (48.87). The article concludes that most websites are not within the recommended readability levels, indicating that better efforts should be made to make online medical information more accessible to the general population [35].

A similar study was conducted by P. Mira et al., in 2012, assessing the readability of medical material available in the American Academy of Facial Plastic and Reconstructive Surgery website. The metrics used were Flesch Reading Ease, Flesch-Kincaid Grade Level, SMOG Grading, Coleman-Liau Index, New Fog Count, New Dale-Chall Formula, FORCAST formula, Raygor Readability Estimate, and the Fry Graph. The recommended readability score for medical re-

| Purpose | Domain | Year | Used Readability Metrics |
|---|---|---|---|
| Classification of spam emails | Web | 2013 | FRE, FKGL, FORCAST, GFI, SMOG |
| Survey question difficulty assessment | Social Sciences | 2013 | DC, FRE, FKGL, GFI |
| U.S. Supreme Court brief analysation | Law | 2011 | FRE, FKGL |
| Special education safeguard documents | Education | 2012 | SMOG |
| 4th and 5th-grade textbooks | Education | 2012 | FKGL, GFI |
| Difficulty of reading academic texts | Education | 2017 | FRE, FKGL |
| Security policies | Computer Security | 2017 | FRE, SRM |
| Readability of tweets, SMS and chats | Social Media | 2014 | FRE |
| Online material regarding facial reconstructive surgery | Health | 2012 | FRE, FKGL, SMOG, CL, NDC, FORCAST, |
| Trial recruitment description | Health | 2016 | FKGL, GFI, NDC, SMOG |
| Online material regarding congestive heart failure | Health | 2017 | CL, FRE, FKGL, GFI, RG, SMOG |
| Relation between education level and readability in newspapers | Journalism | 2004 | FRE, GFI, FKGL |
| Assessment of news topics | Journalism | 2013 | FRE |
| Assessment of popular newspapers | Journalism | 2018 | FKGL, GFI, SMOG |

CL - Coleman-Liau Index, (N)DC - (New) Dale-Chall, FRE - Flesch Reading Ease, FKGL - Flesch-Kicaid Grade Level, GFI - Gunning Fog Index, SRM - Strathclyde Readability Measure [75]

Table 3.6: Overview of application of readability metrics for different purposes and domains.

sources is around the 6th-grade level, although the average grade at the end of the assessment was around 12th-grade level [62].One other example is a article, written in 2015 by Danny TY Wu et al., which conducts an assessement of the readability of trial descriptions to recruit participants. The metrics used are the New Dale-Chall, Flesch-Kincaid Grade Level, SMOG and Gunning Fog Index. After analysing the results, the authors discovered that the readability predictors, on average, indicated that a person needed at least 18 years of education to easily understand the material [77].

One focal point of this dissertation is the journalistic area where readability is an important factor when it comes to make news accessible for the target audience. Readability formulas pioneers, like Robert Gunning, creator of the Gunning Fog Index (section 3.3), was highly experienced in the world of newspapers and publishing. Sets of experiments are conducted to assess readability in journalism, as is the case of a 2013 study by I. Flaounas et al., where the Flesch Reading Ease Formula was used to compare the readability of different news topics. When ranking the topics from highest readability to lowest, the authors found that "Sports" and "Arts" were the highest scoring ones whereas topics like "Politics" and "Environment" were less readable. Also, the same metric (FRE) was used to rank readability levels of different news outlets [24].

In 2003, T. McLellan et al. studied the readability of Australian newspapers, comparing it to the current standards of education. The authors examine and characterise three different regional newspapers and how the behave through the years; firstly inspecting the complexity of news writing for both editorials and articles, using basic measures as number of paragraphs, sentence length, article length, and syllables per word. The same process is used for examining the levels of readability output by the three metrics used: Gunning Fog, Flesch-Kincaid Grade Level, and the Flesch Reading Ease. Finally the metric statistics are compared against each other ordered by year and then by newspaper [45].

Some other studies assess the ease to read of popular newspapers, as is the case of the 2018 work by Erik Jonsson which measures readability for the broadsheet *The Daily Telegraph* and the tabloid *Daily Mail*. Conducting this experiment, the author used five methods to analyse the articles: readability formulas, active and passive voice usage, type-token ratio, clauses per sentence, and linking words. The readability formulas used were the Flesch-Kincaid Grade Level, SMOG and Gunning Fog Index. The average results for each predictor are presented in table 3.7.

| Newspapers | Flesch-Kincaid | SMOG | Gunning Fog |
|---|---|---|---|
| *The Daily Telegraph* | 14.6 | 12.3 | 16.5 |
| *Daily Mail* | 12.7 | 11.8 | 14.6 |

Table 3.7: Results using readability formulas for 10 articles from *The Daily Telegraph* and 10 articles from the *Daily Mail*, adapted from the study by Jonsson [33].

Although readability predictors as SMOG indicate that the *Daily Mail* articles are more understandable (lesser grade level values), the author found that two of the methods used, type-token ratio and the linking words, showed the opposite.

# Chapter 4

# ProseBot

## 4.1 Automated Journalism

Automated journalism is an emerging topic that is increasing in popularity. Graefe [28] believes it will play a major role in the process of news creation in future. Automated journalism processes rely upon the analysis of structured data where the interesting events must be identified to produce narratives. The steps present in most automated journalism software are illustrated in Figure 4.1, they are similar to the ones described in Section 2.2, considering that automated journalism is also an NLG problem.

In the first step, the system has to gather the available data regarding the narrative to be written. This works best in data-heavy domains such as sports. For example, to write a football match report the system needs various information about the events, for instance, the final score, scored goals, the players who scored those goals, or red cards. The second phase is the identification of the most important events for the narrative, resulting from pre-defined criteria, different for each context. The selected events are then ranked by their newsworthiness, following rules set by domain experts. In the football reporting context, the information about the goals is more significant than the number of fouls committed during a match, for example. Next, the system generates a narrative using NLG approaches, turning collected and filtered data into fluent written text. The final step is the publishing of the narrative, usually preceded by an expert review. The post-editing of generated texts is common for NLG systems.

### 4.1.1 Existing Solutions

There is already some existing software in use to help the production of automated news. In this section, we will explore three distinct tools for this area: Wordsmith[1], Arria Studio[2] and Data2Text Studio. Wordsmith is a solution by Automated Insights, this data-driven tool transforms available and organized information into written words. The Arria NLG project also claims the ability to transform structured into natural language, providing high scalability. Both of these

[1] https://automatedinsights.com/wordsmith/
[2] https://www.arria.com/

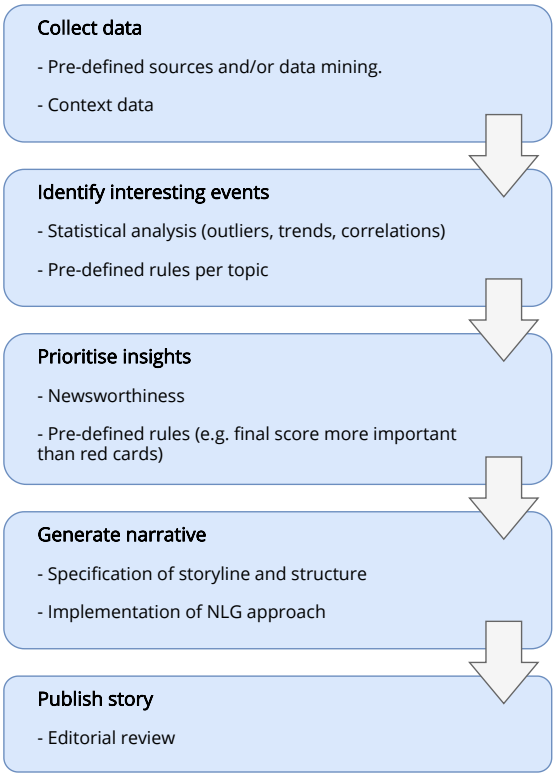



Figure 4.1: Steps of automated journalism systems, adapted from Graefe's work. [28]

tools are closed-source and used for commercial purposes and there is no information about their particularities. Data2Text Studio [19] is one of the solutions for automated journalism that creates templates and rules through machine learning. Besides, means are provided for developers to edit templates of pre-trained models. This tool has applications to different sectors, from sports news to weather reports. The downside of the mentioned tools is that all of them just focus on the English language while ZOS's target readers are multi-language. In 2019, a Finnish broadcasting company implemented the Voitto-robot, based on templates and decision trees. Later the template files were made accessible to journalists so they could modify them [51].

### 4.1.2 Benefits

The automated journalism process offers multiple advantages on the supply of the news for its readers. One of the main benefits is the speed of the process from the collection of the data to the writing of an article. Automation makes the production of articles with available data almost real-time. Automated journalism also, not being limited by human resources, increases the number of news produced in a time frame, when compared to human journalistic work.

One other advantage of automated journalism systems is the accuracy of the narratives. If the intrinsic data available is precise, then produced news articles show a lower error rate. Algorithms "*do not make simple mistakes like misspellings, calculation errors, or overlooking facts*" [28]. In

addition, algorithms objectively look at facts, following predefined rules, resulting in unbiased narratives. Automated journalism also provides personalisation of the produced content, tailoring it to its readers. The production of news in different languages, functionality of the ProseBot (Section 4.2), using the same underlying information is a major benefit of automated journalism.

### 4.1.3 Limitations

Automated journalism requires well-organised data so that computer software can read and process it. For that matter, automation suits data-driven contexts such as football reporting. Although there are domains where information is simply not available. In other cases, information exists, but the quality is poor for generating narratives from it. Algorithms in automated journalism can not correlate insights taken from data analysis. For example, Leonel Messi scores a hat-trick against Real Madrid and, for that, he is considered the man of the match. An algorithm would not correlate the two pieces of information without the existence of a specific rule for that condition.

However, the major limitation of automated journalism and the major reason why this topic is currently in such an experimental state is the writing quality of computer-generated narratives. By only considering underlying information, algorithms make produced news appear too technical, without the human nuances.

## 4.2 ProseBot

Currently, there are lots of information being generated per match and it would be very resource-consuming to have journalists analyse each game so that they could write an article about it. Therefore, ZOS helped to develop a system that could process the information stored on their databases and produce a small summary of a match and its main events. The ProseBot is a template-based NLG system with a data-to-text approach built on the shoulders of a text generation system mentioned in Section 4.3.2, GameRecapper [1].

In 2019, Vasco Ribeiro devoted his master thesis to enhance the ProseBot in partnership with ZOS, making his goal to increase the variety and quality of the content produced [59]. After the enhancements, the ProseBot could generate content in four different languages: English, Spanish, Portuguese and Brazilian Portuguese, create a full article with defined sections: title, subtitle, summary, and body, include a higher number of match events and produce phrases carrying more content.

Regarding the system's architecture, ZOS developed an API that groups the data of a certain match in a JSON structure, this structure is used by the ProseBot system as input data.

There is a Generation module, the content generation basis algorithm, that receives as input a JSON structure containing the match data and the language in which the content should be generated. This module also accesses a template collection. Each template has gaps to be filled out depending on the variable information that can outcome from different events and sections of the article. Additionally, grammatical and linguistic functions are applied to the generation

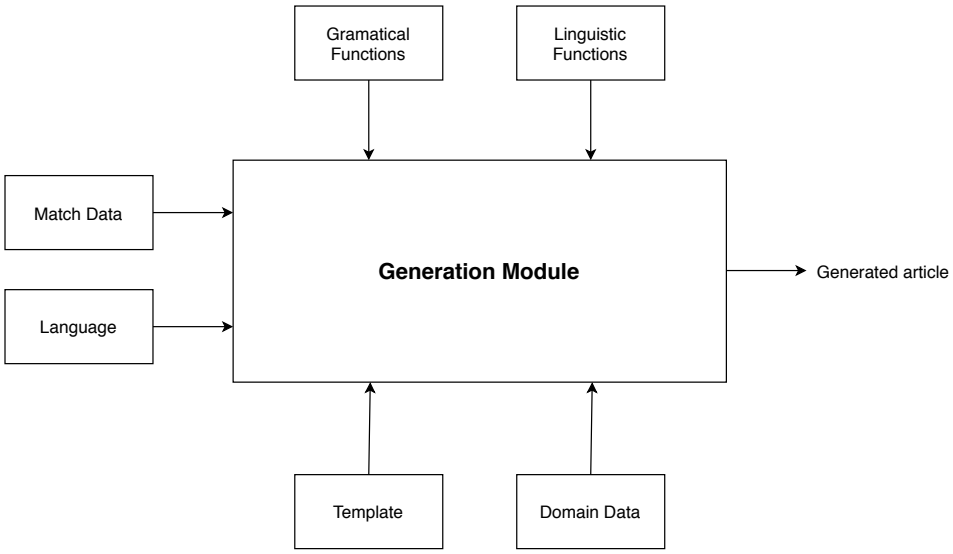



Figure 4.2: ProseBot system's architecture.

algorithm to assure text correctness grammatically and to turn numerical information into plain text.

The content generated by the ProseBot ended up being evaluated based on the methodology presented by Bouayad-Agh et al. [10] to rate the text quality. In total, 36 reports for football matches of the Portuguese league were generated and then distributed in three different questionnaires, written in Portuguese. Fifteen experts from ZOS were involved in this evaluation process. Three additional questionnaires were conceived, written in the other three languages that the ProseBot can generate (Spanish, English, Brazilian Portuguese). Each questionnaire was evaluated by three journalism experts with knowledge of each language. The subjects were asked to rate in a scale from 1 to 5 the understandability and the fluency of the shown texts.

Furthermore, to explore how general users perceive the reports generated by the ProseBot in comparison to real human-crafted articles, another questionnaire was created. This questionnaire contained 10 news reports, 5 generated by the ProseBot and the other 5 real reports published on ZOS's site, *www.zerozero.pt*. The involved users were asked if each report was ready to be published online, without specifying if they were either automatically generated or written by reporters.

For the questionnaires concerning understandability and fluency, 25 out of the 36 reports were attributed the maximum rating. Regarding the aspect of understandability, the average rating was 4.61 (92,22%), as in terms of the aspect of fluency the average rating was 4.16 (83,22%). Table 4.1 contains some additional information related to the obtained results.

| | Understandability | Fluency |
|---|---|---|
| Nº Reports with 100% rating | 6 | 0 |
| Nº Reports with under 90% rating | 9 | 28 |
| Nº Reports with under 80% rating | 1 | 10 |
| Maximum Rating (%) | 100 | 96 |
| Minimum Rating (%) | 76 | 72 |
| Average Rating (%) | 92.22 | 83.22 |

Table 4.1: Results information about understandability and fluency in Portuguese.

During this part of evaluation process, a relation between the number of goals of a reported match and the results. Matches without any goals show a lower rating for understandability, related to the existence of few events reported in the article generated. On the other hand, the increase of number of goals per match brings a slight increase of understandability and decrease of fluency. The cause for this fluency decrease is the repetition of information in longer texts by the ProseBot, making the text feel less natural.

Table 4.2 shows the results when evaluating the aspects of understandability and fluency in other languages. The results show that there is no relevant variation of text quality for generated texts in other languages, since the system is equally designed for each language with the same number of templates with the same complexity.

| | Portuguese | Brazilian Portuguese | English | Spanish |
|---|---|---|---|---|
| Average Understandability | 4.62 | 4.47 | 4.55 | 4.55 |
| Average Understandability (%) | 92.33 | 89.38 | 91.05 | 91.03 |
| Average Fluency | 4.18 | 4.19 | 4.22 | 4.19 |
| Average Fluency (%) | 83.67 | 83.80 | 84.38 | 83.80 |

Table 4.2: Results information about understandability and fluency in other languages.

An interesting aspect is the evaluation of the developed self-evaluation tool for the ProseBot and how it compares to ratings given by real reporters.

In conclusion, the articles produced by the system were believed to be correct and ready to be published.

## 4.3 Previous Work

### 4.3.1 GoalGetter

GoalGetter is a system capable of converting match data to speech (data-to-speech approach), developed at the Eindhoven University of Technology [37].

The text generation process is based on:

- the use of templates,
- a prior Knowledge State that records which data have been conveyed, and which have not yet been conveyed,
- a Context State that formulates conditions concerning the use of referential and quantificational expressions.

After this process, there is a Prosody module that enriches the text assigns accents and metrical structure boundaries. Finally, the enriched text becomes an input for the voice generation module, which converts it into speech signal.

### 4.3.2 GameRecapper

GameRecapper is a data-to-text NLG system developed by João Aires in partnership with ZOS [1]. This system is based on GoalGetter and generates summaries for Portuguese football matches, from data provided by an API developed by ZOS. A template-based approach was used for this system, forcing the developer to manually create templates for the different events and characteristics of a football match. The articles created by the system are structured in three paragraphs: an introduction where the result and teams are presented, a description of the goals scored and a conclusion where changes of standings on the table are analysed. Most of the time, the system can produce fluid and understandable text. However, in some cases, the fluidity of the content generated decreases due to a high number of similar phrases, that use the same template. This ended up turning the articles more repetitive and to look more artificial.

### 4.3.3 Statistical Language Modeling

In 2017, João Soares developed a data-to-text NLG system in partnership with ZOS [67]. Unlike GameRecapper (Section 4.3.2), this system uses a statistical approach for the generation of football matches reports based on the extraction of information of a corpus. The corpus was built with a great number of previous match reports, so models could be conceived and learn from them. The corpus is human-crafted and like GameRecapper the report is divided into three parts: introduction, goal description, and conclusion. After the corpus is built, the system trains corpus-based models that generate non-lexical phrases that later go through a process of lexicalisation. Although this statistical system can produce more varied content than a template-based system, sometimes texts generated lack information and length.

# Chapter 5

# Metrics System for Sports Journalism

## 5.1 Introduction

One of the main focus of this dissertation is the evaluation of journalistic text in the football domain and the output of the generated texts by the ProseBot, examined in Chapter 4. We focused on two sets of metrics: NLG automatic metrics, like BLEU, NIST, and METEOR and readability indicators. Additionally, Part-Of-Speech (POS) tagging and Named Entity Recognition (NER) application in the journalistic context was explored. All the metrics and NLP techniques were incorporated in a RESTful API created with the Python framework Flask [1] and divided into three modules. Initially, the evaluation methods were projected to just be used for computer-generated texts, but later on, it seemed interesting to apply readability measures to real articles hosted in the website, as well as the POS tagging and named entity recognition modules. Just the automatic metrics, normally used for NLG evaluation, were exclusively applied to the ProseBot output. Figure 5.1 visually describes the system architecture.

Modules are independent of each other and have distinct workflows, although similar. Each module uses different support Python libraries and has its own storage space. The NLG metrics module bears two functionalities. The first one is retrieving the values of the NLG metrics of text after the comparisons against reference texts. For the action of receiving the NLG metric scores for a text, first, the user emits an HTTP GET request to the /automatic_metrics_scores endpoint, alongside a JSON object containing the text, the text's match id code and the language of the text, represented by indication 1.1 in Figure 5.1. After processing the information, the module responds, sending back a JSON object holding the scores of the NLG metrics (indication 1.2 in Figure 5.1). The other feature of this module is adding a text to the reference text folder system. To achieve this, the client-side send an HTTP GET request to the /add_reference endpoint alongside a JSON object containing the reference text, the text's match id code and the language of the text, this step is illustrated by indication 1.3 in Figure 5.1. The module responds with an indication of whether the action was successful or not, in the form of a JSON object (indication 1.4 in Figure 5.1).

[1] https://palletsprojects.com/p/flask/

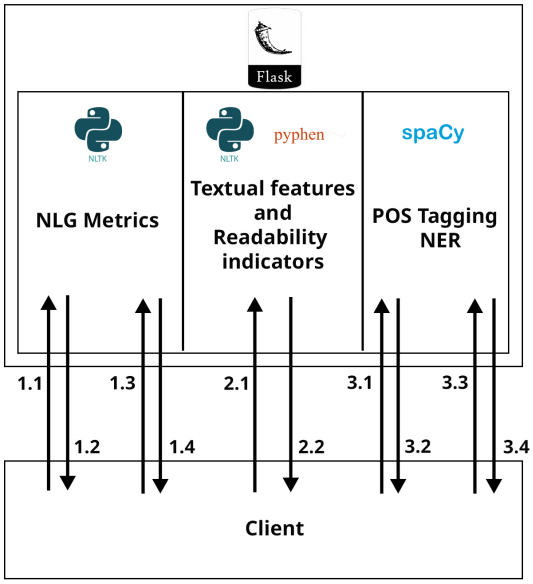



Figure 5.1: Metrics API Architecture.

The readability metrics and textual features module has only one endpoint that delivers the readability metrics and text characteristics of a text. To use the only functionality of the module, the client-side emits an HTTP GET request to the /readability endpoint and sends a JSON object containing the text and the language of the text, pictured in indication 2.1 in Figure 5.1. After the calculations, the module retrieves a list of readability metrics and a list of textual features in the form of a JSON object (indication 2.2 in Figure 5.1).

The last module includes two different natural language processing techniques: Part-Of-Speech (POS) Tagging and Named Entity Recognition (NER). Both of them are explained and exemplified in Section 5.4. For POS tagging feature, an HTTP GET request is sent by the client-side to the module endpoint /pos_tag alongside a JSON object containing the text and the language of the text, as illustrated in indication 3.1 in Figure 5.1. As a response, the system delivers an organised list of part-of-speech tokens as a JSON object (indication 3.2 in Figure 5.1). On the other hand, to get the named entities, an HTTP GET request is sent by the user to the endpoint /ents with a JSON object containing the text and the language of the text, as demonstrated in indication 3.3 in Figure 5.1. After processing the information brought by the request, the module retrieves a list of named entities identified in the text through a JSON object (indication 3.4 in Figure 5.1).

To make information accessible to users, ZOS has a database containing various data about matches, teams, and players provided by Opta that collects live football match statistics. The database access to get information is moderated by the zerozero.pt API. The overall structure was important for the development of the metrics system considering that some services integrated into the Flask API bits of information as entry parameters, especially the match identification code (ID). Additionally, during the development of the different modules, other advantages were

found for the used modules. A more in-depth explanation of all the modules and endpoints used will be presented in the next sections.

## 5.2 NLG Automatic Metrics Module

This module was developed to assess the texts produced by ProseBot, comparing with text references and finding correlations with the collected human ratings. The automatic metrics used in the system are BLUE, NIST-2 and METEOR, created for evaluating machine-translation systems but commonly used for NLG evaluation, as discussed in Chapter 2.

For the integration of these metrics, the Natural Language Toolkit (NLTK) Python library[2] was used, one of the most popular libraries for Natural Language Processing that provides an implementation for them in the translation metrics package[3]. For the BLEU metric, a method based on the work of Papineni et al. [49], for calculating the metric for a single corpus-level, was used for all the hypotheses and their respective references, rather than a sentence-level computation. For METEOR, we used the NLTK method, nltk.translate.meteor_score, able to calculate its score for the hypothesis with multiple references, based on the ideas described in a paper by Alon Lavie et al. [39]. Finally, also for NIST a method to calculate a single corpus-level score for all the hypotheses and their respective references was used. Single corpus-level was prefered over sentence-level scoring so that instead of averaging the sentence-level scores, there is micro-average precision as in the original BLEU metric and every method uses a list of tokens as the hypothesis input. Tokens, much like n-grams, are the parts of text that are divided into components as words and punctuation, in a process known as tokenization, illustrated in Listing 5.1. The word_tokenize method is also used for parsing the texts in this module.

```
>>> from nltk.tokenize import word_tokenize
>>> s = '''Good muffins cost $3.88 in New York.  Please buy me
... two of them. Thanks.'''
>>> word_tokenize(s)
['Good', 'muffins', 'cost', '$', '3.88', 'in', 'New', 'York', '.', '
    Please', 'buy', 'me', 'two', 'of', 'them', '.', 'Thanks', '.']
```

Code Listing 5.1: Python example of tokenization, using word_tokenize method from the NLTK.tokenize library.

### 5.2.1 Endpoints

This API module has two endpoints: */automatic_metrics_score* and */add_reference*; and receives JSON objects as parameters. The first endpoint returns the BLEU, METEOR and NIST scores for

[2] https://www.nltk.org/

[3] For documentation, see https://www.nltk.org/_modules/nltk/translate/.

| **Endpoints** | */automatic_metrics_score* | */add_reference* |
|---|---|---|
| **Usage** | Getting the scores for BLEU, METEOR and NIST of a text | Adding a reference text to the system, where it will be organized by language and match ID code |
| **Required request parameters** | JSON object with three attributes:<br>- Language (Type: string, Key: "lang")<br>- Match ID (Type: integer, Key:"id")<br>- Report text (Type: string, Key:"text") | JSON object with three attributes:<br>- Language (Type: string, Key: "lang")<br>- Match ID (Type: integer, Key:"id")<br>- Reference text (Type: string, Key:"text") |
| **Parameter Example** | Listing 5.2 | Listing 5.2 |
| **Response** | JSON object with the BLEU, METEOR and NIST scores as attributes | JSON object with a 'success' attribute indicating whether the action was successful or not |
| **Response Example** | Listing 5.3 | Listing 5.4 |

Table 5.1: Description of the NLG metrics module endpoints' usage, required request parameters, and response.

a text compared against its reference texts. The latter is used for adding texts to the reference folder system. Both of the service endpoints require a JSON object as a parameter with three attributes: language, match ID, and the generated text string: the report text for retrieving NLG metrics and the reference text for the */add_reference* action. An explanation of this module's endpoints is provided in Table 5.1.

After receiving a request, the */automatic_metrics_score* endpoint responds with a JSON object containing the BLEU, NIST and METEOR scores after comparison of the text against its reference texts, illustrated in Listing 5.3. The */add_reference* return a JSON object indicating if the reference text was successfully added to the file system in the Flask API server, exemplified in Listing 5.4.

```
1  {
2      "lang" : "en",
3      "id" : 6942190,
4      "text" :  "Borussia Dortmund crushed Schalke 04, 4-0, on Saturday,
           on matchday 26. In this competition, Dortmund's team came from
           four wins, and Gelsenkirchen's team came from a draw. Raphael
           Guerreiro was on fire. After 28 minutes, Erling Haaland opened
           the scoring for Borussia Dortmund, laid on by Thorgan Hazard.
           Shortly before the interval, Raphael Guerreiro fired home
           Dortmund's team's second goal, with a left-foot shot, laid on by
            Julian Brandt. After the break, Thorgan Hazard struck for eam,
           with a right-foot shot from outside the box, laid on by Julian
```

```
            Brandt. After 63 minutes, Raphael Guerreiro netted the final
            goal of the game, laid on by Erling Haaland. After the result
            Borussia Dortmund are 2nd in the table, 54 points, while Schalke
             04 occupy 8th place, 37 points. In relation to forthcoming
            league matches, Dortmund's team visit Wolfsburg. Meanwhile,
            Gelsenkirchen's team will host FC Augsburg."
    }
```

Code Listing 5.2: JSON object required by the NLG metrics module endpoints.

```
{
    "bleu" : bleu_score,
    "meteor" : meteor_score,
    "nist" : nist_score
}
```

Code Listing 5.3: JSON object response from endpoint */automatic_metrics_score*.

```
{
    "success" : true
}
```

Code Listing 5.4: JSON object response from endpoint */add_reference*.

## 5.3 Readability Metrics Module

One form of assessing the quality of a text and how readers perceive it is the measurement of readability features. Text attributes as average word length can appraise the adequacy of content for a certain audience.

This module of the API, the Readability Metrics module, analyses and processes texts, and consequently, deliver information about its readability features. Originally, this evaluation element was idealised to only be applied to automatically generated news. However, it was expanded to be used for real match reports hosted in the zerozero.pt website to create a dashboard and for posterior comparison of results.

### 5.3.1 Textual attributes

The first step for the development of this module was the extraction of information from the input text, to select information about textual characteristics. Similarly to the reference-based metrics module, the process of tokenisation was used to isolate words, punctuation and sentences using methods from the NLTK library [4]. Moreover, a hyphenation process was adopted to identify and extract syllables from the plain text using Pyphen [5], a Python module to hyphenate words using internal or external dictionaries. One problem with hyphenation is in how it works differently depending on the language used, but nonetheless, Pyphen provides support for a great variety of idioms, including Portuguese. After the implementation of these mechanisms, the module could extract and use the following readability attributes from a text:

- Sentence count
- Word count
- Letter/character count
- Average words per sentence
- Average syllables per word
- Average letters per word
- Polysyllables count
- Longest sentence length

### 5.3.2 Readability formulas

As discussed in Chapter 3, the application of readability formulas is a popular approach to identify the peculiarities of a text's target audience (i.e. minimum education level to understand a text). Most of the traditional readability predictors use textual attributes to calculate a score, so using the extraction of attributes from Subsection 5.3.1 as a foundation, the integration of the formulas was fairly simple. The readability formulas used were the **Flesch Reading Ease Scoring**, **Flesch-Kincaid Grade Level**, the **Automated Readability Index (ARI)**, the **Coleman-Liau Index**, and the **Gunning-Fog Index**. The incentive to use the enumerated metrics out of a varied scope of readability formulas is the clear adaptation to the Portuguese language, the main idiom of the reports hosted on the zerozero.pt website, made by Hélder Antunes and Carla Teixeira Lopes [4], mentioned in Section 3.12. Thus, the selected readability indicators can be confidently used for both English and Portuguese Language.

### 5.3.3 Endpoints

The main goal of this module is to receive a text as input and output a set of readability metrics related to its attributes. There is only one endpoint for this module - */readability* - which receives a JSON object as a parameter, containing two attributes: the language, either Portuguese or English, and the text string; as it is exemplified in Listing 5.5. After processing the text and calculating the text metrics and readability formulas scores, the module returns an JSON object with the scores

[4] word_tokenize and sent_tokenize http://www.nltk.org/api/nltk.tokenize.html#module-nltk.tokenize
[5] https://pyphen.org/

| **Endpoints** | */readability* |
|---|---|
| **Usage** | Getting the text attributes and readability formulas of a text. |
| **Required request parameters** | JSON object with two attributes:<br>- Language (Type: string, Key: "lang")<br>- Report text (Type: string, Key: "text") |
| **Parameter Example** | Listing 5.5 |
| **Response** | JSON object containing the scores of the readability formulas used and a list of the text attributes. |
| **Response Example** | Listing 5.6 |

Table 5.2: Description of the text attributes and readability formulas module endpoints' usage, required request parameters, and response.

of the readability formulas used, listed in Section 5.3.2, and a list containing the textual features enumerated in Section 5.3.1. Table 5.2 details the */readability* endpoint.

```
{
    "lang" : "en",
    "text" :  "Liverpool triumphed over Wolverhampton, 1-2, on Thursday
        , in the match for the 24th round. In this competition,
        Wolverhampton's team came from a win, and Liverpool's team came
        from five wins. Jordan Henderson was on fire. After 8 minutes,
        Jordan Henderson opened the scoring for Liverpool, from a corner
        , laid on by Alexander-Arnold. In the 51st minute, Raul Jimenez
        struck for Wolverhampton, through a header, laid on by Adama
        Traore. After 84 minutes, Roberto Firmino fired home Liverpool's
         team's second goal, with a left-foot shot, laid on by Jordan
        Henderson. After the result Liverpool are 1st in the table, 70
        points, while Wolverhampton occupy 7th place, 34 points. In
        their next fixture, Liverpool's team visit West Ham. Meanwhile,
        Wolverhampton's team visit Manchester United."
}
```

Code Listing 5.5: JSON object required by the readability module endpoint.

```
{
```

```
        "ari": 7.17,
        "coleman_liau": 3.62,
        "flesch": 43.34,
        "flesch_kincaid": 1.94,
        "gunning_fog": 2.03,
        "text_attr": {
            "complexword_count": 7,
            "letter_count": 630,
            "longest_sent": 23,
            "sent_count": 9,
            "sentlen_average": 14.555555555555555,
            "syll_count": 229,
            "word_count": 131,
            "wordlen_average": 1.748091603053435,
            "wordletter_average": 4.809160305343512,
        }
    }
```


Code Listing 5.6: Output JSON object by the readability module endpoint.

### 5.3.4 Integration

After its development, this module of the API was incorporated in ZOS's production process and installed in the zerozero.pt back office. For every news page an readability evaluation section was added where metrics and formula scorings are displayed, as it is exemplified in Figure 5.2. This addition was shared in all the elements of the newsroom for experimentation purposes.

Additionally, readability metrics and textual features were used for an overall view of statistics of ZOS newsroom. With the collaboration of ZOS, we created a dashboard ranking match reports by the all the implemented metrics, exemplified in Figure 5.3. Individual statistics about the elements of the newsroom were also added to the dashboard.

## 5.4 POS Tagging and Named Entity Recognition Module

This is a module with two main mechanisms of natural language processing: Part-Of-Speech (POS) Tagging and Named Entity Recognition (NER). These text processing techniques are not a focal point of the developed work, however they were found useful in some aspects of the evaluation of match reports, especially for human-written ones. In the next paragraphs POS Tagging and NER will be briefly explained.

### 5.4.1 Part-Of-Speech Tagging

POS tagging is a widely used process in the area of computer linguistics that classifies a token based on its definition and context in a sentence or paragraph, labelling it appropriately as a noun,


Figure 5.2: Article page with an added side section containing metrics and readability formula scorings.

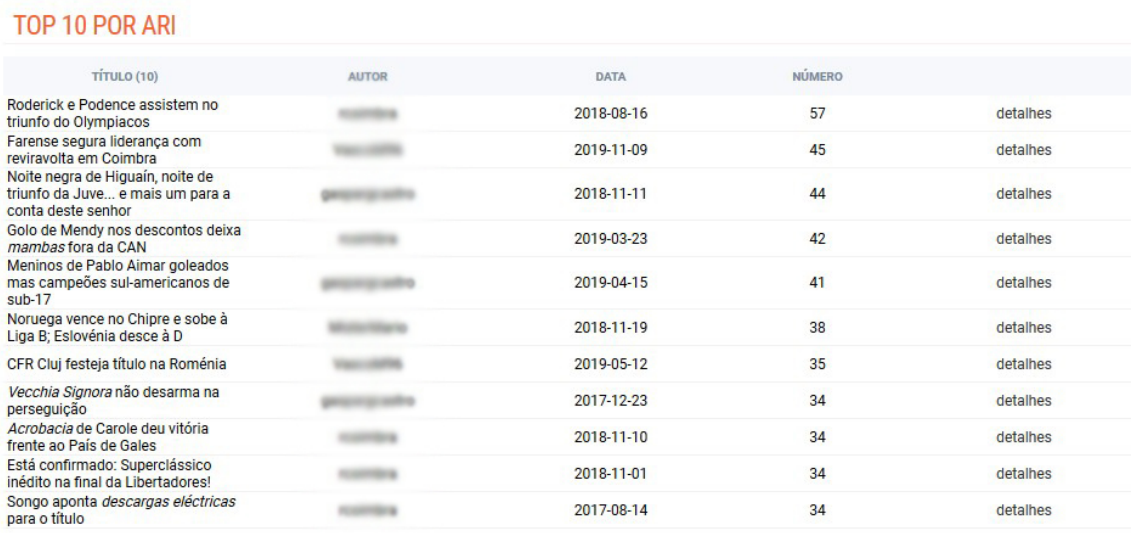
verb, adverb, adjective, pronoun, conjunction, and interjection. In listing 5.7 is an example of POS tagging using a NLTK library for that purpose.

```
>>> sentence = "My name is Jocelyn"
>>> token = nltk.word_tokenize(sentence)
>>> token
['My', 'name', 'is', 'Jocelyn']
>>> nltk.pos_tag(token)
[('My', 'PRP$'), ('name', 'NN'), ('is', 'VBZ'), ('Jocelyn', 'NNP')]
```

Code Listing 5.7: POS Tagging Example using NLTK.

### 5.4.2 Named Entity Recognition

On the other hand, Named Entity Recognition (NER) is a process towards information extraction with the goal of locating and classifying named entities in text into pre-defined categories such as the names of persons, organisations, locations, expressions of times, quantities, monetary values, etc. NER suits the sports journalism context where it can extract names of clubs, organisations, people, players, managers, and cities.



TOP 10 POR ARI

| TÍTULO (10) | AUTOR | DATA | NÚMERO | |
|---|---|---|---|---|
| Roderick e Podence assistem no triunfo do Olympiacos | [redacted] | 2018-08-16 | 57 | detalhes |
| Farense segura liderança com reviravolta em Coimbra | [redacted] | 2019-11-09 | 45 | detalhes |
| Noite negra de Higuaín, noite de triunfo da Juve... e mais um para a conta deste senhor | [redacted] | 2018-11-11 | 44 | detalhes |
| Golo de Mendy nos descontos deixa *mambas* fora da CAN | [redacted] | 2019-03-23 | 42 | detalhes |
| Meninos de Pablo Aimar goleados mas campeões sul-americanos de sub-17 | [redacted] | 2019-04-15 | 41 | detalhes |
| Noruega vence no Chipre e sobe à Liga B; Eslovénia desce à D | [redacted] | 2018-11-19 | 38 | detalhes |
| CFR Cluj festeja título na Roménia | [redacted] | 2019-05-12 | 35 | detalhes |
| *Vecchia Signora* não desarma na perseguição | [redacted] | 2017-12-23 | 34 | detalhes |
| *Acrobacia* de Carole deu vitória frente ao País de Gales | [redacted] | 2018-11-10 | 34 | detalhes |
| Está confirmado: Superclássico inédito na final da Libertadores! | [redacted] | 2018-11-01 | 34 | detalhes |
| Songo aponta *descargas eléctricas* para o título | [redacted] | 2017-08-14 | 34 | detalhes |

Figure 5.3: Example of ranked information in the created dashboard.

### 5.4.3 SpaCy Library

For this module of the API, the spaCy [6] Python library was used, also focused on text processing. One of the most important features of this NLP library, and what made us embrace it, was the provided pre-trained models designed for numerous purposes and available for an impressive range of languages, Portuguese included.

Both selected models for English and Portuguese included the POS Tagging and NER features with decent accuracy values. Coincidentally, the Portuguese model used was trained on Wikipedia, mostly focused on news and media. For NER, it can classify locations, organisations, people, and other miscellaneous keywords. In Listing 5.8 there is an example of the Portuguese trained model applied to an excerpt of a match report, written in Portuguese as well.

```
1
2 #Model loading and setup
3 >>> nlp = spacy.load("pt_core_news_sm")
4 >>> text = "O FC Porto triunfou sobre o Benfica numa tangencial vitoria, no
      Sabado, 3-2, na jornada 20."
5 >>> doc = nlp(text)
6
7 #POS Tagging Output
8 >>> [(token.lemma_, token.pos_) for token in doc]
9 [('O', 'DET'), ('FC', 'PROPN'), ('Porto', 'PROPN'), ('triunfar', 'PROPN'),
     ('sobrar', 'ADP'), ('o', 'DET'), ('Benfica', 'PROPN'), ('numa', 'VERB'),
      ('tangencial', 'NOUN'), ('vitoria', 'PROPN'), (',', 'PUNCT'), ('o', '
```

[6] https://spacy.io

```
    ADP'), ('Sabado', 'PROPN'), (',', 'PUNCT'), ('3-2', 'NUM'), (',', 'PUNCT
    '), ('o', 'PROPN'), ('jornada', 'PROPN'), ('20', 'PROPN'), ('.', 'PUNCT'
    )]

#Named Entity Recognition Output
>>> [(ent.text, ent.label_) for ent in doc.ents]
[('FC Porto', 'ORG'), ('Benfica', 'ORG'), ('Sabado', 'LOC')]
```

Code Listing 5.8: spaCy Portuguese model application example.

POS Tagging wise, the model finds and classifies tokens as nouns or proper nouns (PROPN), verbs (VERB), determiners (DET), punctuation (PUNCT), prepositions (ADP), and numerals (NUM). The results are not always perfect, as the model recognises the token 'triunfar' (in English, to triumph) as a preposition. As in the case of NER, the model identifies three entities: 'FC Porto' and 'Benfica' as organisations (ORG), both of them football clubs; and 'Sábado' (in English, Saturday) as a location (LOC), which is not quite right, reminding the existing error rate. However, the model demonstrates positive results when recognising football clubs and player names, useful for labelling the report with relevant keywords for future readers.

### 5.4.4 Endpoints

This module has two distinct endpoint: */pos_tag* to output POS tagging, and */ents* to output NER labelling. In this module, for both endpoints, the argument of the received request requires a JSON object containing the text and language attributes, as exemplified in Listing 5.5. The */pos_tag* endpoint a JSON object listing the tokens for every different part-of-speech identified during text processing, as illustrated in Listing 5.9. The */ents* endpoint returns the list of recognised entities organised in locations, miscellaneous, organisations, and people. The response JSON object is illustrated in Listing 5.10. Table 5.3 describes of the endpoints of this module.

```
    "pos": {
        "nouns": [
            "ball",
            "score",
            "player"
        ],
        "preps": [
            "after",
            "the"
        ],
        "verbs": [
            "shoot",
            "dribble"
        ],
```

```
        ...
    }
```

Code Listing 5.9: JSON object response for the */pos_tag* endpoint.

```
    "ents": {
        "locations": [
            "London",
            "England",
            "North London",
            "Tottenham"
        ],
        "organisations": [
            "Spurs",
            "Tottenham",
            "Wolverhampton"
        ],
        "misc": [
            "NES",
            "Wolves",
            "Doherty",
        ],
        "people": [
            "Joao Moutinho",
            "Jimenez",
            "Bergwijn",
            "Mourinho",
            "Aurier",
            "Rui Patricio"
        ]
    }
```

Code Listing 5.10: JSON object response for the */pos_tag* endpoint.

| **Endpoints** | */pos_tag* | */ents* |
|---|---|---|
| **Usage** | Getting labelled tokens after the part-of-speech tagging of a text | Getting the extracted named entities of a text |
| **Required request parameters** | JSON object with two attributes:<br>- Language (Type: string, Key: "lang")<br>- Report text (Type: string, Key: "text") | JSON object with two attributes:<br>- Language (Type: string, Key: "lang")<br>- Report text (Type: string, Key: "text") |
| **Parameter Example** | Listing 5.5 | Listing 5.5 |
| **Response** | JSON object with an array of tokens organised by the respective part-of-speech | JSON object with a list of entities organised by people, organisations, locations and miscellaneous |
| **Response Example** | Listing 5.9 | Listing 5.10 |

Table 5.3: Description of the POS tag and NER module endpoints' usage, required request parameters, and response.

## 5.5 User Assessment

To assess how important the metrics provided by this module for six elements of ZOS's newsroom, we prepared a questionnaire, illustrated in Appendix C. The questionnaire was sent through ZOS's communication channel on the 20th of June 2020 and the results were collected up until the 25th of June 2020. In this questionnaire, the respondents are asked about the importance of consulting the textual attributes (Subsection 5.3.1) and readability indicators (Subsection 5.3.2) of match reports, and how they benefit the journalistic work. The questions were answered on a five-point scale where 1 is "Not important" and 5 is "Very important". The results of the questionnaire are presented in Figure 5.4.

Three out of the six respondents (50%) found the ability to consult the number of words of a report to be "Important", while the other half (50%) were "Neutral" about it. Five of the respondents (83%) indicated that consulting the number of sentences was "Important" for them, while just one (17%) thought the word count information was "Very important". Regarding the ability to consult the number of characters of a report, three out of the six respondents (50%) found it to be "Important", two (33%) were "Neutral" about it, and one (17%) considered it "Not very important". When asked about the ability to inspect the average number of words per sentence, two of the respondents (33%) considered it "Very important", three (50%) found it "Important" and one (17%) respondent remained "Neutral". About the ability to consult the number of polysyllables, two (33%) considered it "Important", two (33%) were "Neutral", and the other two (33%) classified it as "Not very important".

Next, reporters read a description of readability formulas used and they were asked about the importance of displaying their scores for a report. Two (33%) of the respondents considered it to be "Very important", while the other four (67%) classified it as being "Important". Finally, we

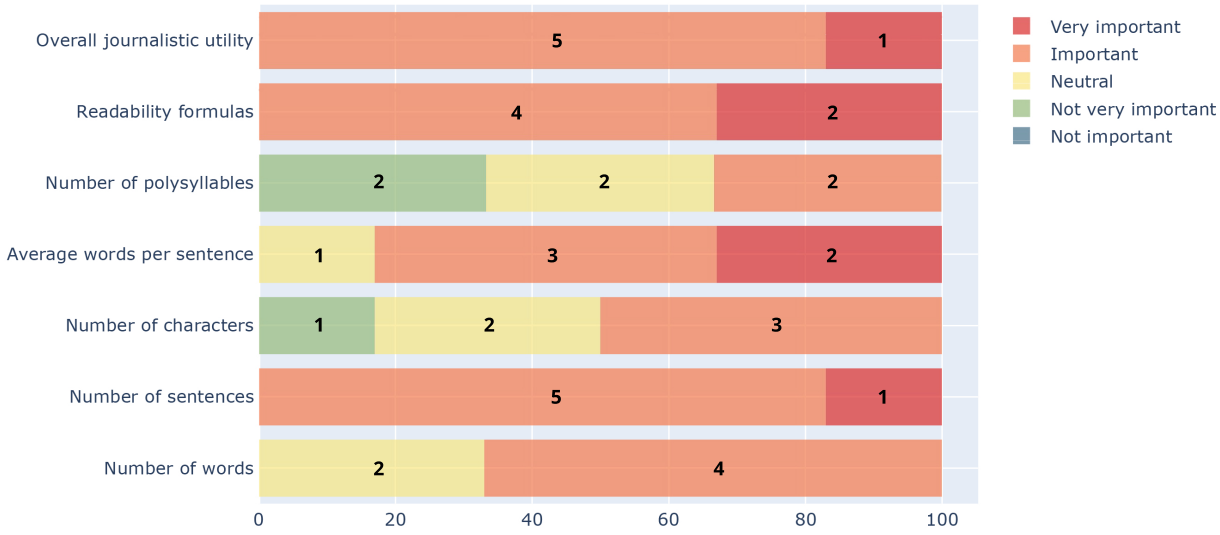



Figure 5.4: User assessment regarding the importance of different metrics applied to match reports for journalistic work.

questioned the utility of this set of metrics for journalistic work. Five out of the six respondents (83%) classified it as "Useful", and one respondent (17%) rated it as "Very useful".

# Chapter 6

# Match Event Selection UI for Automatic Report Generation

## 6.1 Introduction

A football match report consists in a narration of all the events occurred during the 90 minutes. In those articles, newsrooms captivate readers by describing the major happenings in a match, what was the score, how the goals were scored, who played, who got sent off, which player had a remarkable exhibition, and how did the score influence the classification of the competition. Well-described sets of information are an important factor of the production of match reports. However, most of the times writers do not produce reports for less relevant matches, as it would be too time-consuming, considering the analysis of the game information plus the writing process, even though that the match event information is available and accessible.

The user interface developed during this work tries to reduce the time spent on writing a football match report. It provides the ability to create a simple report for any match that has its data available. For a certain match, it allows users to select relevant match events, where the system renders pre-generated paragraphs or sentences related to that information, extracted from the ProseBot system.

Sometimes, NLG systems can generate a draft text with decent content but need some human expert editing before being ready to be released. This process is called post-editing, common in the NLG area. In this tool the writer can post-edit the report and add or remove text freely upon event selection, the system will only serve as a helping tool so that the control is always on the human side. The interface also integrates an area that displays text metrics of the report text and readability formulas scores received from the readability module of the developed API (Section 5.3.2).

## 6.2 Report Structure and Events

For determining the events that would be available for selection on the interface, we had to find which ones were normally featured in real reports. After analysing a set of reports, the main events and information were isolated. Let us take as an example the following report of an Arsenal win over Manchester United, 2-0, on the first day of 2020:

> # Arteta claims first Gunners win
> **Pepe and Sokratis on target against Man Utd**
> Rookie Arsenal boss Mikel Arteta claimed his first win after the Gunners overpowered an out-of-sorts Manchester United at the Emirates on New Year's Day.
> Short of the injured-again Paul Pogba, the visiting Red Devils started the contest on the front foot, but found themselves behind on eight minutes when Arsenal winger Nicolas Pepe slotted home his fifth goal of the season after a fortuitous bounce of the ball.
> From that juncture, Arsenal imposed themselves on the contest, Pepe hitting the post before Greek defender Sokratis doubled their advantage when he smashed home from a corner.
> Ole Gunnar Solskjaer's United show glimpses of promise after the break, but - in truth - never looked like finding their way back into the game.
> The defeat leaves the Red Devils in fifth spot, five points shy of Chelsea who drew 1-1 with Brighton earlier in the day.
> Meanwhile. Arsenal move up to tenth in the table, four points behind United."

Figure 6.1: Match report example of an Arsenal win over Manchester United, 2-0, on the 1 of 2020, taken from ZOS's English website, playermakerstats.com.

In relation to the report structure, the first well-defined aspect is the division between the title, subtitle and text body. The title section is usually a small and infatuating phrase emphasising some important part of the match. The subtitle, usually more discreet, adds some more information about the game and server as a leverage to the report body. The focus of the interface is on the report body, the section were all the events and happenings in a match are narrated. The first paragraph of this section tends to be an introduction and brief summary of the match. As for events, clearly the most important are the goals scored, which players scored them and how the goals were scored. Another important information spotted in the text is the positions modification in the classification table of the competition, in this case the Premier League. Player information is also meaningful for the report narrative, readers are interested about the performance of the players, as players that had a positive influence on the match scoring goals or assisting. Moreover, particularly for the zerozero.pt website, there is an additional relevance about the performance of Portuguese players playing in foreign leagues. One more important feature for match narrations are events that have an high impact on the course of the game as red cards and penalties. In some cases, the form, latest results of a team, is an important factor as well as the classification of the teams before a match.

Finally, and usually as the conclusion paragraph there is information about the next matches for each team.

So, after the analysis process, the events to be selected are divided in seven distinct sections:

- **Teams form**, indicating the five last results for both home and away team.
- **Classification before the match**, indicating the position on the classification table and number of points for both home and away team.
- **Goals scored**, pointing out the players who scored, the minute of the match of the goal, and if it was scored from a penalty kick.
- **Red cards**, indicating which players got a red card and the minute of the red card.
- **Key players**, registering which players performed well, scored or assisted more than one goal with special emphasis on Portuguese players in foreign leagues.
- **Classification after the match**, indicating the position on the classification table and number of points for both home and away team.
- **Next fixture**, pointing the next opponent and if the fixture is either home or away for both home and away team.

## 6.3 Views and Interaction

This interface is designed to produce football match reports in a fast and simple way by selecting information items, add pre-generated paragraphs or sentences related to the selected events. Even though the generated text is the main feature, the user has the power of editing the text at any given time. In this section, we will illustrate the steps of the user interaction to achieve all the functionalities of this system.

Firstly the interface appears to the user with the totality of its sections completely blank: both event selection field and report text areas; indicating that they will be filled out by some user action. The only highlighted area of the interface is the input field where the user is indicated to insert the match ID corresponding to the match to be reported. The initial view is represented in Figure 6.2.

After the insertion of an ID of a match, the interface will load all the data while displaying a brief loading animation. Subsequently, all the information in the match event section is filled out and ready to be selected; and on the other hand, the report text section presents a generated suggested title, subtitle and body introduction.

Afterwards, the user can select the intended event information, which will be added in the form of a sentence or a paragraph to the report body section. The structure of the event information is based on a tree-like view, in which multiple nested options can be displayed and selected. In this case, each event section enumerated in section 6.2 is an option that contains children options

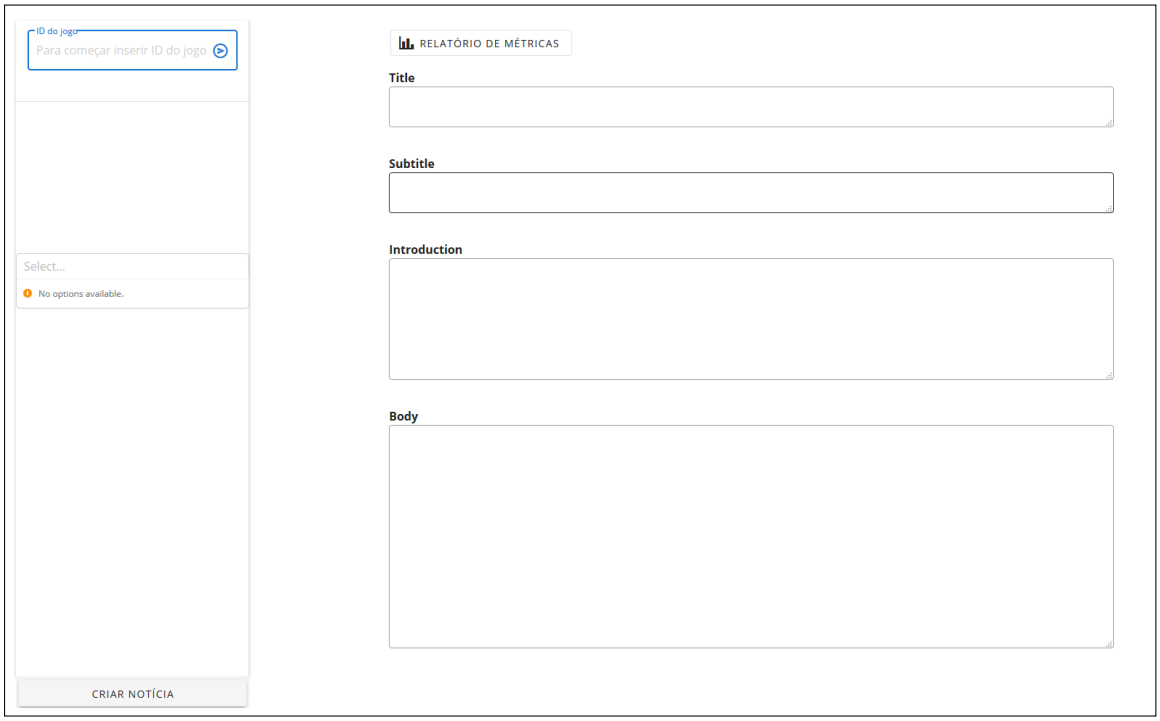



Figure 6.2: Initial view of the writing tool interface.


Figure 6.3: View after inserting the match code and loading the event information.

enclosing partial information of that section. Figure 6.4 illustrates the different selection options within a section and how the sentences are structured based on the chosen information in an effort to increase fluency in the text generation.

An additional feature is the possibility to consult the text metrics and readability assessment

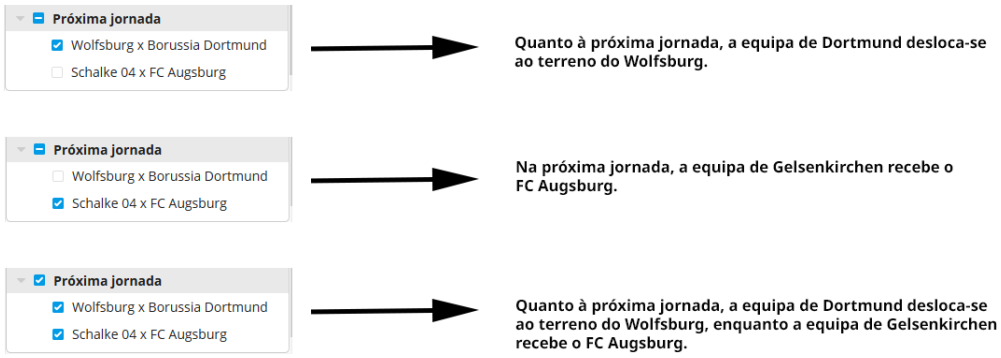



Figure 6.4: Sentence construction based on selected information.

applied to the report content, at any time. To navigate to that view the user should press the button on the top of the report text sections. This triggers the appearance of a modal menu displaying the metrics of the produced text divided in two sections - textual metrics and readability scorings. Next to every exhibited metric there is the corresponding average value for zerozero.pt reports for comparison purposes, as illustrated in Figure 6.5. Plus, the general values are updated every time the report text suffers a change.

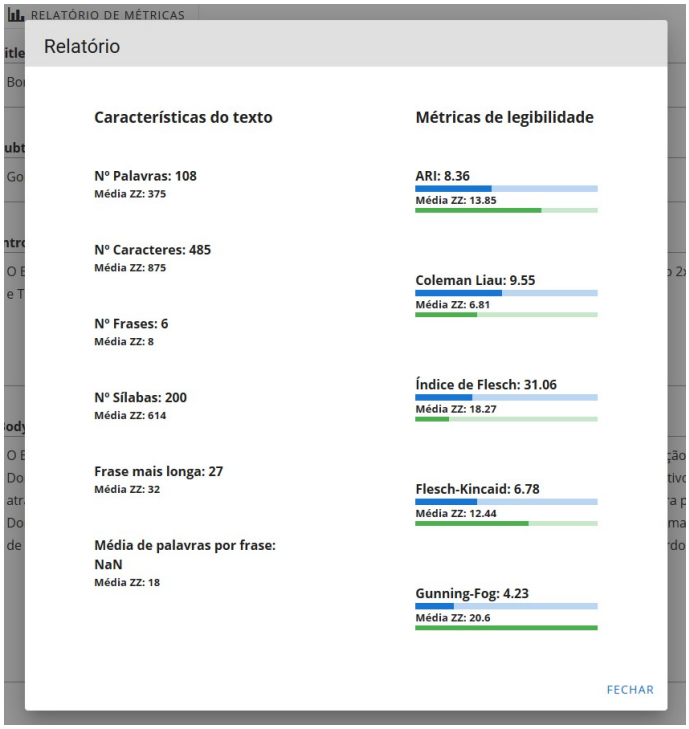



Figure 6.5: Interface area showing text metrics, alongside the average values of the zerozero.pt reports for each metric.

## 6.4 System Architecture

The aim of the interface is to be easily used by a writer and make the user-flow optimally simple. To achieve that user experience, the interface was developed with the help of a progressive JavaScript front-end framework, Vue.js [1]. This writing tool was designed to retrieve information from real match data from ZOS's database possible with a connection to their in-house API. However, this main API provides an immense amount of data for a single match, a great portion is not included in the set of information to be displayed by the interface. Plus the data structure is not prepared to be used by the interface without some previous processing. For that purpose, a REST API was created in Node.js [2] using the Express.js [3] library, creating a connection between the interface and the zerozero.pt API to parse the important data from the vast information available and properly structure it to be used by the front end interface and transferred into the event selection sections. Additionally, there is a metrics system which gets its data from the readability module of the developed Python Flask API. Every metric value is compared to the average values of all the reports hosted in zerozero.pt. Below there a diagram illustrating the architecture of this journalistic writing tool.

The step that initiates all the communication between servers and the interface is the introduction of a valid match ID in the input area of the interface by the user. Then, the interface makes a GET HTTP request to the Node.js server using the Axios [4], a Node.js library for web applications needing to consume and display data from an external API, and sending the match ID as the request parameter. The Node.js server receives the request, gets the match ID and formulates its own request to the ZOS main API, using the receiving ID. The main API retrieves a JSON object with the all the match information, the Node.js server parses that JSON response, grouping the relevant data and correctly structuring the response to be consumed and displayed by the interface. In Appendix A is an example of the final JSON object received by the interface, containing all the events to be shown to the user, plus some basic information about the game: result, league, and date; and about the teams: names and crest images.

The generated report texts are first created as a whole text and then divided in sections corresponding to each event and stored in the Node.js middle server as JSON objects. One downside of this approach is the fact that generated reports must be divided in sections and introduced in the system manually as there is not a automated form of dividing the report in an event-aware fashion.

In terms of the integration with the metrics system, every time the user triggers the display of the metrics section, Axios sends a request with the current text as a parameter to both endpoints of the readability module of the metric API, mentioned in section 5.3. In return, the interface gets and display the information about the textual features and the readability formulas scorings.

[1] https://vuejs.org/
[2] https://nodejs.org
[3] http://expressjs.com/
[4] https://github.com/axios/axios

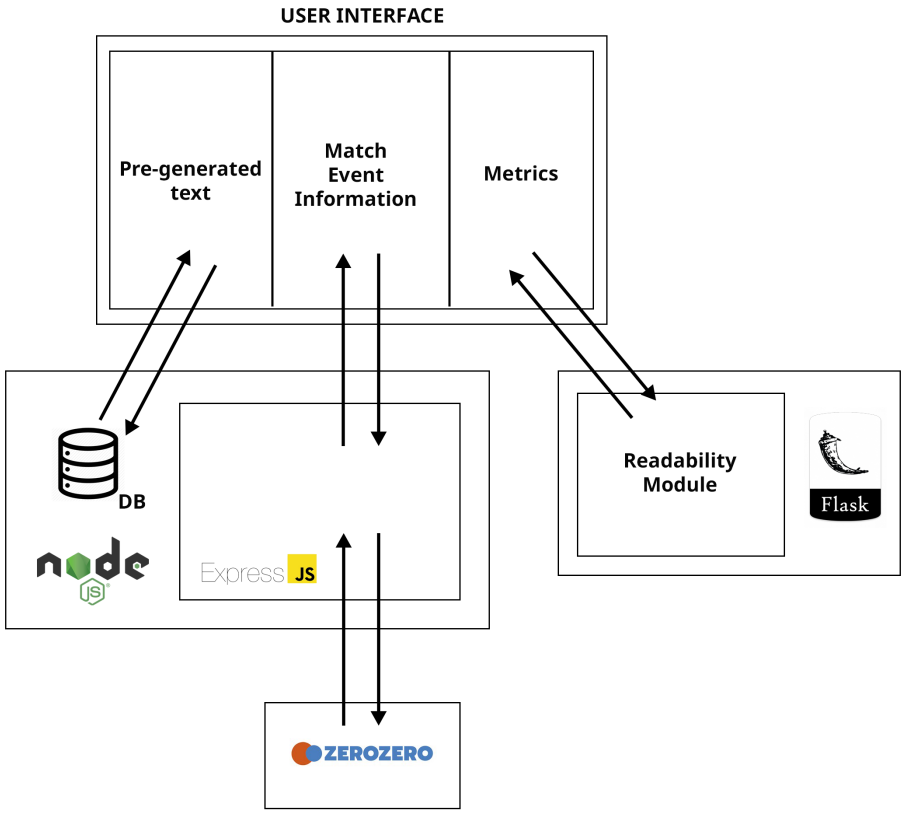



Figure 6.6: Writing tool system architecture.

## 6.5 User Assessment

To assess the importance and usability of the helping tool to produce football matches reports, we reached out to the system's target users: real journalists. For this evaluation phase, seven elements from the zerozero.pt writing department were invited for a direct interview. During the interview, the main concept was explained and they were invited to use the system. The task was to create reports for 4 different matches, as well as consult the text metrics, without any helping hints.

Most of the subjected journalists easily surpassed the tasks, although there were a few difficulties and some user feedback. The first slight complication in the user interaction was the fact that half of the users took more time than expected to find the area to input the ID of the match. A major topic of user feedback was that in reports information about different events is usually separated in distinct paragraphs rather than in a single block as it was implemented in the interface.

After the interface interaction stage, the reporters filled out a questionnaire, in Appendix B evaluating the importance of the eligible events and assessing the general usability of the system. The latter was measured with the System Usability Scale (SUS) [11], a reliable tool for estimating usability of a system with a 10-item form with a Likert Scale-like five response options; from

"Strongly agree" to "Strongly disagree". The responses are then calculated and normalised into a final usability score on a scale from 0-100. Figures 6.7 and 6.8 illustrate the results obtained from the questionnaire.

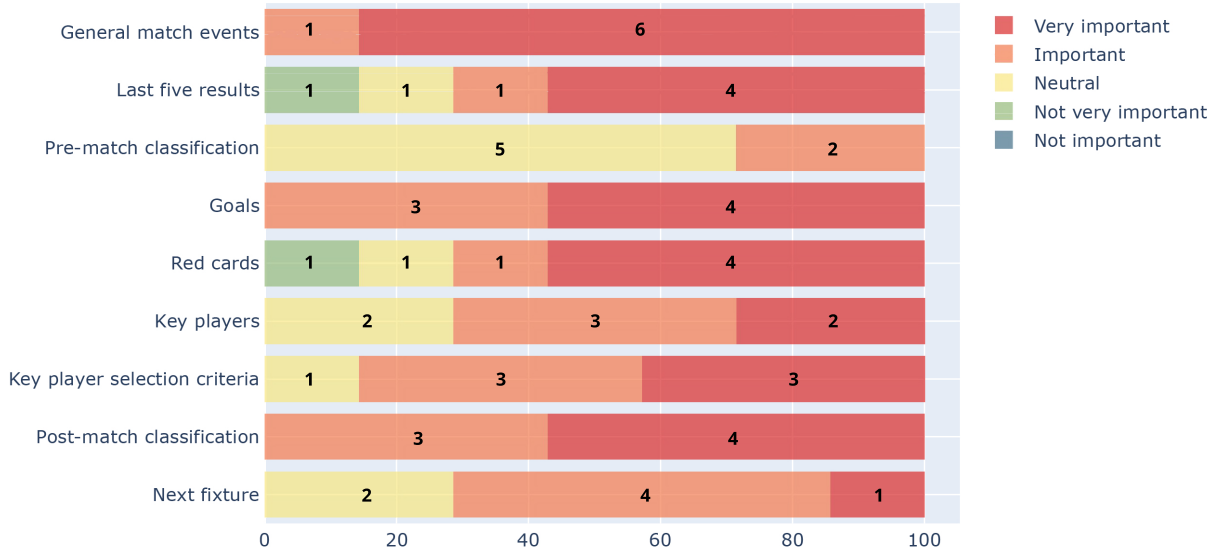



Figure 6.7: Writing tool's interface user assessment regarding the importance of selecting different match events.

Firstly, we inquiry the respondents for the importance of controlling and selecting information about the events of a match, using a five-point scale in which 1 is "Not important" and 5 is "Very important". It was considered "Very important" by six out of the seven and "Important" by the rest of the elements. When asked about the ability to control information about the five latest results of both home and away team, four journalists considered it to be "Very important" (57.1%), while the others evaluated it as four (14.3%), three (14.3%) and two (14.3%) scale points respectively. Five out of all the respondents (71.4%) showed to be neutral towards the ability to manage the information about the classification before the match, while one journalist (28.6%) found it to be "Important. Regarding the control of information about the goals scored, 57.1% considered it to be "Very important" and 42.9% to be "Important". In respect to red card information, 57,1% classified it as "Very important" as other answers were equally split between "Important" (14.3%), "Neutral" (14.3%), and "Not very important" (14.3%).

When asked about the idea of controlling the information about the key players of a match, two respondents found it "Very important" (28.6%), three considered it to be "Important" (42.9%) and other two classified it as "Neutral" (28.6%). For this section, the criteria established was that players with more than a goal, players with more than one goal assist, and Portuguese players playing in foreign competitions would be considered key players for the match report. Regarding this topic, three subjected newsroom elements found that the criteria are very well applied (42.9%),

three classified it only as well applied, the other two were neutral. Four journalists (57.1%) answered "Very important" when asked about the control to manage the post-match classification information for both teams; the other three described that option as "Important" (42.9%). Only one journalist (14.3%) considered "Very important" the ability to control information about the next fixture for both home and away team, while four of them responded as "Important" (57.1%) and two as "Neutral" (28,6%).

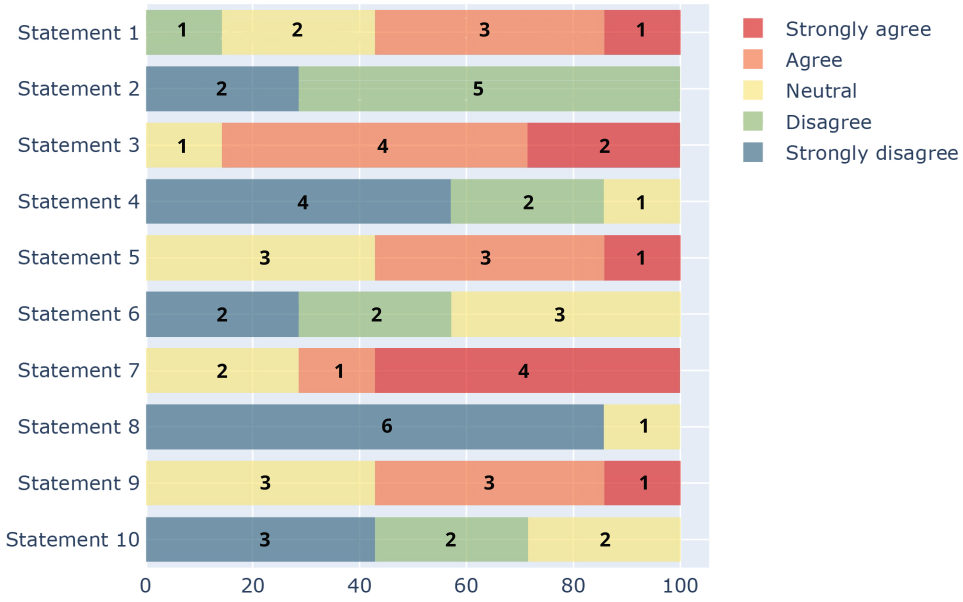



Figure 6.8: SUS results from the questionnaire. The SUS statements are:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Additionally, we added some open-ended questions so users could make suggestions regarding

the different match events mentioned in the questionnaire. Regarding the last five results, users suggested the ability to access a broader history of results between the two teams rather than just five latest results. For events related to the classification of the teams, respondents proposed an indication of possible variation in classification positions before and after the match. Concerning the teams' next fixtures information, users suggested showing a brief overview of the opponents' current performance. Finally, one of the most requested features was the addition of information about seasonal stats of goals scorers, players who assisted for the match's goals, red-carded players, and key players in general.

Usability-wise, the system achieved a SUS score of 76.5, indicating above-average usability. The interviews developed great user feedback, causing the interface to be adjusted in some aspects. The text related to each event in the report body section was divided into paragraphs, the match ID input area was highlighted and some additional indications were added to lead the insertion of the match code by the user. In retrospective, there was an overall positive towards the system by the interviewed reporters.

# Chapter 7

# Experiments

## 7.1 Application of NLG metrics to Computer-Generated Reports

The objective of this experiment is to assess which parameters influence score variations in different NLG evaluation metrics and how well these metrics correlate with human evaluation. In this experiment, NIST, BLEU and METEOR were used as metrics. For the collection of human ratings, we used the data from Vasco Ribeiro's work [59], where five ZOS journalists assessed a set of 30 matches from the 2018/2019 season of the Portuguese League generated by the ProseBot for both fluency and understandability on a five-point scale.

For this experiment, out of all the assessed match reports, ten matches reports of a match week of the Portuguese league produced by the ProseBot were used. For each computer-generated match report, the respective human-crafted match report was collected from zerozero.pt, used as a text reference. Table 7.1 displays the obtained results of the BLEU, METEOR, and NIST scores after the computer-generated match reports were compared against the reference texts (human match reports from zerozero.pt), versus the average human evaluations for both fluency and understandability. BLEU and METEOR scores range from 0 to 1, depending on the occurrence of n-grams of the hypothesis in the reference corpus. On the other hand, NIST weights the n-grams differently, by giving more importance to less frequent ones. NIST scores usually range from 0 to 10, as indicated by Zhang et al. [78] in a study where BLEU and NIST metrics scores are also compared to human judgements to assess correlation.

Afterwards, we calculated the Pearson's correlation [50] between the NLG metrics results obtained and the human ratings for understandability and fluency, similarly to the comparison made by Belz and Reiter [8], exhibited in Table 7.2. The results show an overall low correlation between the metrics scores and the human judgements for both fluency and understandability.

In general, the results do not portrait a correlation between automatic and human evaluation. We point out two factors that cause this type of discrepancy: the number of human ratings gathered, the size of the dataset, and the level of adequacy of the NLG metrics applied. When collecting human-written match reports that would fit as a text reference, some problems were faced, especially finding reports that would not have some preconceived facts as backstories to managers or

| Match report | BLEU-2 | METEOR | NIST-2 | Human rating - Fluency | Human rating - Undestandability |
|---|---|---|---|---|---|
| 1 | 0.15482 | 0.20233 | 2.03385 | 19 | 22 |
| 2 | 0.18446 | 0.17516 | 2.48975 | 20 | 21 |
| 3 | 0.15259 | 0.12774 | 1.98253 | 21 | 23 |
| 4 | 0.15482 | 0.20233 | 2.03385 | 19 | 19 |
| 5 | 0.22241 | 0.27621 | 1.67491 | 21 | 23 |
| 6 | 0.31132 | 0.25044 | 3.01781 | 25 | 22 |
| 7 | 0.06682 | 0.13454 | 0.08129 | 21 | 23 |
| 8 | 0.19437 | 0.2003 | 2.49427 | 20 | 24 |
| 9 | 0.18334 | 0.21146 | 2.14951 | 24 | 24 |
| 10 | 0.15238 | 0.19194 | 1.75408 | 20 | 20 |

Table 7.1: Results of the application of BLEU, NIST and METEOR to the gathered reports, versus the human evaluation results.

| | Fluency | Undestandability |
|---|---|---|
| BLEU | 0.57591 | 0.07069 |
| METEOR | 0.31696 | -0.00186 |
| NIST | 0.261897 | -0.09019 |

Table 7.2: Calculated Pearson's correlation between BLEU, METEOR and NIST, and human judgements of fluency and understandability.

players. For example, in a report about the Wolverhampton away win over Tottenham, the writer mentions Wolves manager, the Portuguese Nuno Espírito Santo as "*another Mourinho protégé*" because of their backstory as manager and player. To reduce the impact of this noise present on some match reports, we used the POS tagging and NER techniques from the module of the developed API to pre-process the text, removing entities and using POS tagging to remove tokens not related to the match events. Reiterating Belz and Reiter's work [8], to obtain trustworthy results the NLG community needs to institute processes to collect adequate references in the context they are being utilised.

## 7.2 Report Characterisation and Readability Assessment

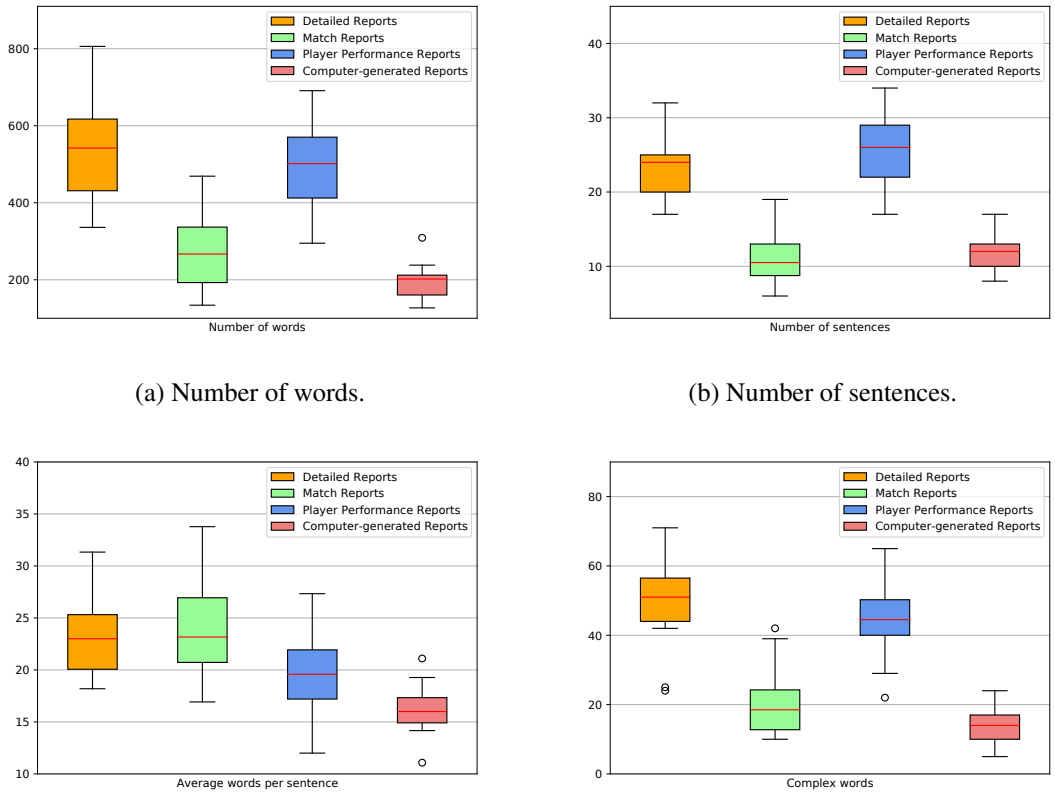
In this section we explore how journalistic sports content can be evaluated without human involvement. Readability metrics were used to evaluate both human-written match reports taken from zerozero.pt and computer-generated match reports. Three different categories of human-written match reports were used for this experimental section:

- **Post-match reports**, this type of articles give a brief narrative of the important events and indicate key information about the match. An example can be found in Section 6.2, the

structure and content of these reports are similar to the ones the developed writing tool (Chapter 6) aims to produce.

- **Detailed reports**, these reports are related to more relevant matches that need to be described more thoroughly, usually longer.
- **Player performance reports**, these articles are focused on how players performed, giving an overview of the best and worst player performances.

For the assessment, 20 reports of each category were collected plus 20 computer-generated match reports, randomly chosen from the ProseBot. Firstly, the characteristics of each set of reports were collected based on the following text features: number of words, number of sentences, number of words per sentence, word length, and number of words with more than three syllables (complex words). Figure 7.1 illustrates the obtained results.




(a) Number of words.

(b) Number of sentences.

(c) Average number of words per sentence.

(d) Number of complex words.

Figure 7.1: Results of the characterisation of four different collections of match reports by number of words, number of sentences, average number of words per sentence, average word length, and number of complex words.

After exploring every feature, the information was gathered to characterise each set of reports. In the first place, detailed reports recorded the second-highest average number of sentences (Figure 7.1b) when compared with other sets and have a fairly high variability of results. The results

show that this category has the highest number of words (Figure 7.1a) in the reports and the highest variance of the number of words out of all collections. Regarding average words per sentence (Figure 7.1c), this collection has the second-highest score. Moreover, results show that this category records the highest values for the number of polysyllabic words (Figure 7.1d) amongst all the other collections and registers a high variability for that text peculiarity.

Post-match reports have the lowest average number of sentences of all the collections (Figure 7.1b), and the average number of words (Figure 7.1a) in this sort of articles is the lowest of the human-written categories. This category has the highest set of results for the number of words per sentence (Figure 7.1c), probably resulting from the low number of sentences and the high number of words, indicating more complex sentences. Besides, this collection records one of the highest variability levels for that feature. The number of polysyllables (Figure 7.1d) appearing in post-match reports is the lowest for the manually written compilations.

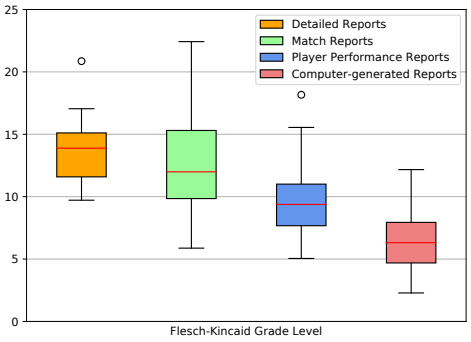
Player performance reports are the ones containing the highest average number of sentences (Figure 7.1b) and the highest fluctuation of that feature of all the collections. Compared to other collections, this one has an average level of words per report (Figure 7.1a) and registers the lowest word to sentence ratio (Figure 7.1c) of the human-written collections, showing high variability. Results also show that player performance reports have the second-highest number of polysyllables (Figure 7.1d), but have the lowest fluctuation of all the human-created categories.

The analysis of the results for computer-generated reports indicates the median number of sentences is 12 (Figure 7.1b), even though this category shows less variability and the lowest number of words (Figure 7.1a) compared against the other collections. Results determined that this collection has the lowest number of words per sentence out of all the collections (Figure 7.1c), and, unlike every other human-based report collection, has a low variability in this area. Computer-generated reports tend to adopt simpler words, having the lowest number of polysyllabic words out of all the collections (Figure 7.1d), and the lowest variability as well, a result visible across all of the features measured for this collection.

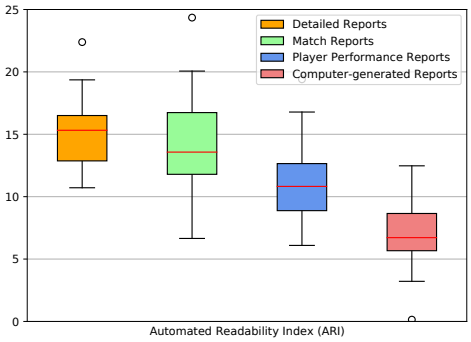
In concluding notes, generally, all human-written collections show higher variability and sentence complexity in the results when compared against the reports generated by the ProseBot.

All the metrics from the readability module of the API developed and described in Chapter 5 were applied to the collected reports: Flesch-Kincaid Grade Level, Automated Readability Index (ARI), Coleman-Liau Index and the Gunning Fog Index. Figure 7.2 illustrates the results obtained.

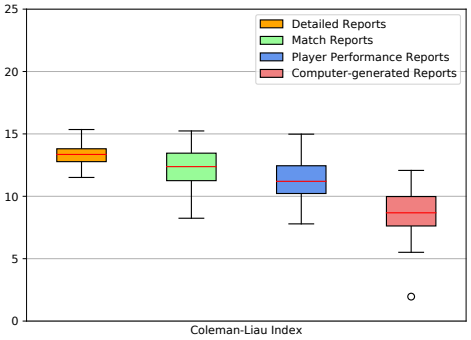
Detailed reports record the highest values of readability grade-level of all collections for every readability formula. Additionally, this collection shows the lowest fluctuation out of the human-written reports. Match reports show the highest variability of readability levels out of all collections. Moreover, most of the grade-level results are high compared against the other categories. Like match reports, player performance articles present a high variability of readability scorings, yet they record the lowest average grade-level of the categories produced by the ZOS's newsroom. Computer-generated reports are the ones showing the lowest readability grade-levels with no results over the 14th-grade level for any readability formula. All the readability formulas adopted



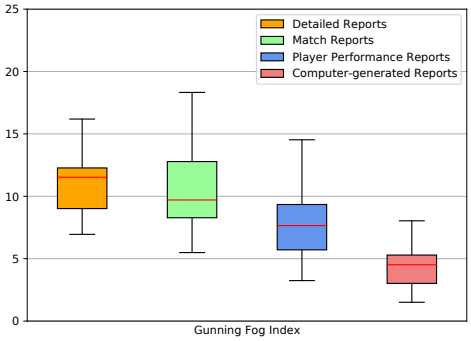

(a) Flesch-Kincaid Grade Level.




(b) Automated Readability Index (ARI).




(c) Coleman-Liau Index.




(d) Gunning Fog Index.

Figure 7.2: Results of the application of the Flesh-Kincaid Grade Level, Automated Readability Index, Coleman-Liau Index, and Gunning-Fog Index to different reports categories.

had a similar behaviour when applied to the different categories. The Gunning-Fog Index (Figure 7.2d) registered an overall higher readability ease indication concerning the other readability formulas scorings. The Coleman-Liau (Figure 7.2c) is the readability indicator showing less variation of scorings, this formula is the only one measuring word length by number of characters.

# Chapter 8

# Conclusions and Future Work

## 8.1 Conclusions

During this dissertation, we focused on two major points: exploring a form of evaluating the quality of football match reports of the website zerozero.pt, and developing a tool to make the process of writing football match reports simpler for reporters. ZOS has the ability to produce both human-written and computer-generated football match reports, the latter created by an in-house NLG template-based system, the ProseBot.

In this sense, during this work, we explore the NLG process and its different tasks and approaches. Moreover, we concentrate on how NLG systems output is evaluated, finding that it can be done either by human judgements or the use of automatic metrics, which compare NLG texts against reference texts. On the other hand, for assessing the quality of both human-written and computer-generated match reports, we went through different readability indicators. We studied how these readability indicators were adjusted to other non-English languages, with special attention to the Portuguese language, the main language of the readers of the zerozero.pt website.

A metrics system was created for assessing the quality and delivering metric information about football match reports. It was implemented as a Flask API using different NLP support libraries. Plus, the metrics system was divided into three different modules: one for providing the scores of automatic metrics for computer-generated reports, another one presenting readability indicators scorings and textual features, and the last one for applying POS tagging and named entity recognition techniques to the match report texts. The development of this system originated the creation of dashboards displaying metric information provided by all the modules of the API for every match report. Other dashboards were created to have an overall view of the metrics for different authors and reports. Finally, we assessed the importance of the metrics system for the context of Sports Journalism by requesting elements of the ZOS newsroom to answer a questionnaire, reporters classified both textual attributes and readability indicators as important for the journalistic work developed at ZOS.

The other main focus of this work was the creation of a tool that would give the writer the ability to produce football match reports in an automated fashion. The developed interface allows the

user to select events from a football match, rendering text from the ProseBot including information about the selected events, in the form of sentences or paragraphs, which can be post-edited at any time. The interface was developed as a single webpage using the Vue.js framework. Additionally, we developed a server using the Express.js framework from Node.js to parse the information received from the main API used by ZOS from where the event information is extracted. Interviews were conducted with experienced ZOS reporters, alongside questionnaires, to assess the importance of selecting events to generate texts, the value of post-editing the generated texts, and the overall usability and UX. The results acquired from the interviews and questionnaires showed that reporters find the integrated features helpful for the production of football match reports, moreover the system's interface showed a good usability. The overall response from the reporters regarding the developed tool was positive, although some adjustments were recommended and applied in the final review.

## 8.2 Future Work

The use of NLG automatic metrics and readability indicators do not represent the full scope of the factors for text quality. A wider range of metrics can be applied to the developed metrics system, which provides that sort of scalability.

In the case of the tool for automated match report generation, the post-editing ability can be used as a gateway for the system to save the edits made by human reporters to learn from them. Also, the developed tool would be useful for editing templates and adding new ones to the ProseBot, adding more variety to the texts produced.

The reports text, taken from the ProseBot, and rendered by the writing tool is manually divided into event sections and manually inserted into the system. The human involvement in this process is due to ProseBot not labelling the event or information that each sentence or paragraph of a match report is associated with. Future labelling of the match reports would make the process of fetching the portion of text related to an event automatic, for all the football matches in ZOS's database.

# References


[1] João Aires. Automatic generation of sports news. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2016.

[2] Abdel Karim Al Tamimi, Manar Jaradat, Nuha Al-Jarrah, and Sahar Ghanem. Aari: automatic arabic readability index. *Int. Arab J. Inf. Technol.*, 11(4):370–378, 2014.

[3] Yazeed Alkhurayyif and George RS Weir. Readability as a basis for information security policy assessment. In *2017 Seventh International Conference on Emerging Security Technologies (EST)*, pages 114–121. IEEE, 2017.

[4] Hélder Antunes and Carla Teixeira Lopes. Analyzing the adequacy of readability indicators to a non-english language. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 149–155. Springer, 2019.

[5] Tatsuya Aoki, Akira Miyazawa, Tatsuya Ishigaki, Keiichi Goshima, Kasumi Aoki, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. Generating market comments referring to external resources. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 135–139, 2018.

[6] R. Scott Baldwin and Rhonda K. Kaufman. A concurrent validity study of the raygor readability estimate. *Journal of Reading*, 23(2):148–153, 1979.

[7] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[8] Anja Belz and Ehud Reiter. Comparing automatic and human evaluation of nlg systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[9] Carl Hugo Björnsson. *Läsbarhet*. Liber, 1968.

[10] Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, and Leo Wanner. Perspective-oriented generation of football match summaries: Old tasks, new challenges. *ACM Trans. Speech Lang. Process.*, 9(2), August 2012.

[11] John Brooke. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, page 189, CRC press, 1996.

[12] John Caylor, Thoman Stict, and J. Patrick Ford. The forcast readability formula. *Literacy Discussion. International Institute for Adult Literacy: UNESCO*, 1973.

[13] Jeanne Sternlicht Chall and Edgar Dale. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books, 1995.

[14] Meri Coleman and Ta Lin Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283, 1975.

[15] Edgar Dale and Jeanne S Chall. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54, 1948.

[16] James RA Davenport and Robert DeLine. The readability of tweets and their geographic correlation with education. *University of Washington*, 2014.

[17] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, 2002.

[18] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632, 2017.

[19] Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin. Data2text studio: Automated text generation from structured data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 13–18, 2018.

[20] Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156, 2020.

[21] Ehud Reiter. How to do an NLG Evaluation: Task-Based (Extrinsic) Performance in Real-World Context. https://ehudreiter.com/2017/04/27/task-based-real-world-nlg-eval/, 2017. Online; accessed 13 January 2020.

[22] José Fernández Huerta. Medidas sencillas de lecturabilidad. *Consigna*, 214:29–32, 1959.

[23] Seth Finn. Unpredictability as correlate of reader enjoyment of news articles. *Journalism Quarterly*, 62(2):334–345, 1985.

[24] Ilias Flaounas, Omar Ali, Thomas Lansdall-Welfare, Tijl De Bie, Nick Mosdell, Justin Lewis, and Nello Cristianini. Research methods in the age of digital journalism: Massive-scale automated analysis of news-content—topics, style and gender. *Digital journalism*, 1(1):102–116, 2013.

[25] Edward Fry. A readability formula that saves time. *Journal of reading*, 11(7):513–578, 1968.

[26] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

[27] Bettye Gilliam, Sylvia C Peña, and Lee Mountain. The fry graph applied to spanish readability. *The Reading Teacher*, 33(4):426–430, 1980.

[28] Andreas Graefe. Guide to automated journalism. Tow Center for Digital Journalism, Columbia University, 2016.

[29] Robert Gunning. Technique of clear writing. McGraw-Hill, 1952.

[30] Grant Ingersoll, Thomas Morton, and Andrew Farris. *Taming text*. Manning, 2013.

[31] Umit Izgi and Burcu Sezginsoy Seker. Comparing different readability formulas on the examples of science-technology and social science textbooks. *Procedia-Social and Behavioral Sciences*, 46:178–182, 2012.

[32] Karen Sparck Jones and Julia R Galliers. *Evaluating natural language processing systems: An analysis and review*, volume 1083. Springer Science & Business Media, 1995.

[33] Erik Jonsson. Read-a-paper-bility: can you read this paper for me?: A readability study of the daily telegraph and the daily mail, Mälardalen University, School of Education, Culture and Communication, 2018.

[34] Liliane Kandel and Abraham Moles. Application de l'indice de flesch à la langue française. *Cahiers Etudes de Radio-Télévision*, 19(1958):253–274, 1958.

[35] Akhil Kher, Sandra Johnson, and Robert Griffith. Readability assessment of online patient education material on congestive heart failure. *Advances in preventive medicine*, 2017, 2017.

[36] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Institute for Simulation and Training, University of Central Florida, 1975.

[37] EAM Klabbers, JEJM Odijk, JR De Pijper, and M Theune. Goalgetter: Football results, from teletext to speech. *IPO Annual Progress Report*, 31:66–75, 1996.

[38] Michael Latzer, Katharina Hollnbuchner, Natascha Just, and Florian Saurwein. The economics of algorithmic selection on the internet. In *Handbook on the Economics of the Internet*. Edward Elgar Publishing, 2016.

[39] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231, 2007.

[40] Timo Lenzner. Are readability formulas valid tools for assessing survey question difficulty? *Sociological Methods & Research*, 43(4):677–698, 2014.

[41] Chin-Yew Lin and FJ Och. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir Workshop*, 2004.

[42] Lance N Long and William F Christensen. Does the readability of your brief affect your chance of winning an appeal. *J. App. Prac. & Process*, 12:145, 2011.

[43] Carmen Gomez Mandic, Rima Rudd, Thomas Hehir, and Dolores Acevedo-Garcia. Readability of special education procedural safeguards. *The Journal of Special Education*, 45(4):195–203, 2012.

[44] G Harry Mc Laughlin. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646, 1969.

[45] Trina McLellan, Grant Dobinson, et al. Tertiary education of journalists and the readability of australian newspapers. *Australian Journalism Review*, 26(1):155, 2004.

[46] Rashmi Mishra, Jiantao Bian, Marcelo Fiszman, Charlene R Weir, Siddhartha Jonnalagadda, Javed Mostafa, and Guilherme Del Fiol. Text summarization in the biomedical domain: a systematic review of recent research. *Journal of biomedical informatics*, 52:457–467, 2014.

[47] Soichiro Murakami, Akihiko Watanabe, Akira Miyazawa, Keiichi Goshima, Toshihiko Yanase, Hiroya Takamura, and Yusuke Miyao. Learning to generate market comments from stock prices. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1384, 2017.

[48] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, 2017.

[49] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[50] Karl Pearson. Mathematical contributions to the theory of evolution.—on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proceedings of the royal society of london*, 60(359-367):489–498, 1897.

[51] Kyösti Puro. Tool for journalists to edit the text generation logic of an automated journalist. Master of Science Thesis, Department of Future Technologies, University of Turku, 2019.

[52] Alton L Raygor. The raygor readability estimate: A quick and easy way to determine difficulty. *Reading: Theory, research, and practice*, pages 259–263, 1977.

[53] readabilityformulas.com. Lix readability formula : The lasbarhetsindex swedish readability formula. https://readabilityformulas.com/the-LIX-readability-formula.php, 2020.

[54] Ehud Reiter. Types of Evaluation: Which is Right for Me? https://ehudreiter.com/2017/01/19/types-of-nlg-evaluation/, 2017. Online; accessed 2 March 2020.

[55] Ehud Reiter. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401, 2018.

[56] Ehud Reiter and Anja Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558, 2009.

[57] Ehud Reiter and Robert Dale. *Building natural language generation systems*. Cambridge university press, 2000.

[58] Ehud Reiter, Roma Robertson, and Liesl M Osman. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1-2):41–58, 2003.

[59] Vasco Ferreira Ribeiro. Jornalista-robot: produção automática de conteúdos de texto como apoio ao jornalismo desportivo. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2019.

[60] Eric Sven Ristad and Peter N Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[61] Rudolf Flesch. How to Write Plain English. https://web.archive.org/web/20160712094308/http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml, 2016. Online; accessed 6 February 2020.

[62] Saurin Sanghvi, Deepa V Cherla, Pratik A Shukla, and Jean Anderson Eloy. Readability assessment of internet-based patient education materials related to facial fractures. *The Laryngoscope*, 122(9):1943–1948, 2012.

[63] RJ Senter and Edgar A Smith. Automated readability index. Technical report, Cincinnati University, Ohio, 1967.

[64] Gwenaelle Cunha Sergio. gcunhase/NLPMetrics: The Natural Language Processing Metrics Python Repository, October 2019.

[65] Rushdi Shams and Robert E Mercer. Classifying spam emails using text and readability features. In *2013 IEEE 13th international conference on data mining*, pages 657–666. IEEE, 2013.

[66] Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*, 2017.

[67] João Soares. Statistical language models applied to news generation. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2017.

[68] Valery Solovyev, Vladimir Ivanov, and Marina Solnyshkina. Assessment of reading difficulty levels in russian academic texts: Approaches and metrics. *Journal of Intelligent & Fuzzy Systems*, 34(5):3049–3058, 2018.

[69] George Spache. A new readability formula for primary-grade reading materials. *The Elementary School Journal*, 53(7):410–413, 1953.

[70] George D Spache and Evelyn B Spache. *Reading in the elementary school*. ERIC, 1973.

[71] Somayajulu Sripada, Ehud Reiter, and Ian Davy. Sumtime-mousam: Configurable marine weather forecast generator. *Expert Update*, 6(3):4–10, 2003.

[72] Yasufumi Taniguchi, Yukun Feng, Hiroya Takamura, and Manabu Okumura. Generating live soccer-match commentary from play data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7096–7103, 2019.

[73] Chris van der Lee, Emiel Krahmer, and Sander Wubben. Pass: A dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 95–104, 2017.

[74] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[75] George RS Weir and Calum Ritchie. Estimating readability with the strathclyde readability measure. In *ICT in the Analysis, Teaching and Learning of Languages, Preprints of the ICTATLL Workshop 2006*, pages 25–32, 2006.

[76] Krzysztof Wołk and Danijel Koržinek. Comparison and adaptation of automatic evaluation metrics for quality assessment of re-speaking. *Computer Science*, 18(2):129, 2017.

[77] Danny TY Wu, David A Hanauer, Qiaozhu Mei, Patricia M Clark, Lawrence C An, Joshua Proulx, Qing T Zeng, VG Vinod Vydiswaran, Kevyn Collins-Thompson, and Kai Zheng. Assessing the readability of clinicaltrials. gov. *Journal of the American Medical Informatics Association*, 23(2):269–275, 2016.

[78] Ying Zhang, Stephan Vogel, and Alex Waibel. Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *LREC*, 2004.

# Appendix A

# Interface Information JSON Object

In this appendix is presented the final JSON object received by the interface of the developed writing tool, mentioned in Section 6.4, containing all the events to be shown to the user, some basic information about the game: result, league, and date; and about the teams: names and crest images.


```
    {
    "match": "Borussia Dortmund vs Schalke 04",
    "result": "4-0",
    "league": "1. Bundesliga 19/20",
    "date": "2020-05-16 14:30:00",
    "home_image": "http://static-img.zz.pt/logos/equipas/107_imgbank.png",
    "away_image": "http://static-img.zz.pt/logos/equipas/105_imgbank.png",
    "options": [
        {
            "id": "form",
            "label": "Ultimos 5 resultados",
            "children": [
                {
                    "id": "form_home",
                    "label": "Borussia Dortmund (DVVVV)"
                },
                {
                    "id": "form_away",
                    "label": "Schalke 04 (EDDDE)"
                }
            ]
        },
        {
            "id": "class_pre_jogo",
            "label": "Classificação pré-jogo",
            "children": [
                {
```

```
                    "id": "class_pre_jogo_home",
                    "label": "Borussia Dortmund - 2º (54 pontos)"
                },
                {
                    "id": "class_pre_jogo_away",
                    "label": "Schalke 04 - 6º (37 pontos)"
                }
            ]
        },
        {
            "id": "goals",
            "label": "Golos",
            "children": [
                {
                    "id": "goal_1",
                    "label": "Erling Haaland (28')"
                },
                {
                    "id": "goal_2",
                    "label": "Raphael Guerreiro (45')"
                },
                {
                    "id": "goal_3",
                    "label": "Thorgan Hazard (48')"
                },
                {
                    "id": "goal_4",
                    "label": "Raphael Guerreiro (63')"
                }
            ]
        },
        {
            "id": "red_cards",
            "label": "Cartões Vermelhos",
            "children": []
        },
        {
            "id": "key_players",
            "label": "Jogadores Chave",
            "children": [
                {
                    "id": "Raphael Guerreiro",
                    "label": "Raphael Guerreiro",
                    "children": [
                        {
                            "id": "Raphael Guerreiro_minutes_played",
                            "label": "87.00 minutos jogados"
                        },
                        {
```

```
                            "id": "Raphael Guerreiro_goals_scored",
                            "label": "2 golos marcados"
                        }
                    ]
                },
                {
                    "id": "Julian Brandt",
                    "label": "Julian Brandt",
                    "children": [
                        {
                            "id": "Julian Brandt_minutes_played",
                            "label": "90.00 minutos jogados"
                        },
                        {
                            "id": "Julian Brandt_assists",
                            "label": "2 assistência(s)"
                        }
                    ]
                }
            ]
        },
        {
            "id": "class_pos_jogo",
            "label": "Classificação pós-jogo",
            "children": [
                {
                    "id": "class_pos_jogo_home",
                    "label": "Borussia Dortmund - 2º"
                },
                {
                    "id": "class_pos_jogo_away",
                    "label": "Schalke 04 - 8º"
                }
            ]
        },
        {
            "id": "next_match",
            "label": "Próxima jornada",
            "children": [
                {
                    "id": "next_match_home",
                    "label": "Wolfsburg x Borussia Dortmund"
                },
                {
                    "id": "next_match_away",
                    "label": "Schalke 04 x FC Augsburg"
                }
            ]
        }
```

```
126         ]
127 }
```

Code Listing A.1: JSON object received by the interface upon request, it contains all the information for the event selection sections.

# Appendix B

# Writing Tool Assessment Questionnaire

This appendix contains the questionnaire, referenced in Section 6.5, used to assess the importance of different aspects of the developed writing tool and its usability. In the first part of this questionnaire, the users had to classify the importance of selecting different football match events for an automated match report in a five-point scale from "Very important" to "Not important at all". For the second part, users respond to the ten statements from the SUS method on a five-point scale from "Strongly agree" to "Strongly disagree".

# Ferramenta de ajuda à produção de notícias sobre jogos de futebol

*Required

## Parâmetros para a criação da notícia

Nesta secção estuda-se a importância da aplicação de diferentes parâmetros para a criação das notícias. Alguns deles estão associados às características do texto gerado (por exemplo, facilidade de leitura) e outros relacionam-se com porções textuais sobre o jogo (golos, cartões vermelhos, resultado, classificação ao final do jogo).

Avalie a importância de poder escolher a informação sobre eventos do jogo. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

Qual a importância de permitir controlar a informação sobre os Últimos Resultados (últimos 5 jogos)? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

O que acrescentaria nos parâmetros relativos aos Últimos Resultados?

Your answer

Qual a importância de permitir controlar a informação sobre a Classificação Pré-jogo? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

O que acrescentaria nos parâmetros relativos à Classificação Pré-jogo?

Your answer

Qual a importância de permitir controlar a informação sobre os Golos? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

O que acrescentaria nos parâmetros relativos aos Golos?

Your answer

Qual a importância de permitir controlar a informação sobre os Cartões Vermelhos? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

O que acrescentaria nos parâmetros relativos aos Cartões Vermelhos?

Your answer

Qual a importância de permitir controlar a informação sobre os Jogadores Chave? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

Como classifica os critérios para escolher Jogadores Chave? (jogadores com 2+ assistências, jogadores com 2+ golos, jogadores portugueses em equipas estrangeiras) *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Mal aplicado | ○ | ○ | ○ | ○ | ○ | Bem aplicado |

Dentro dos Jogadores Chave, qual a importância da informação sobre Minutos Jogados? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

Dentro dos Jogadores Chave, qual a importância da informação sobre Golos Marcados? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

Dentro dos Jogadores Chave, qual a importância da informação sobre Assistências? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

O que acrescentaria nos parâmetros relativos aos Jogadores Chave?

Your answer

Qual a importância de permitir controlar a informação sobre a Classificação Pós-jogo? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

O que acrescentaria nos parâmetros relativos à Classificação Pós-jogo?

Your answer

Qual a importância de permitir controlar a informação sobre a Próxima Jornada? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

O que acrescentaria nos parâmetros relativos à Próxima Jornada?

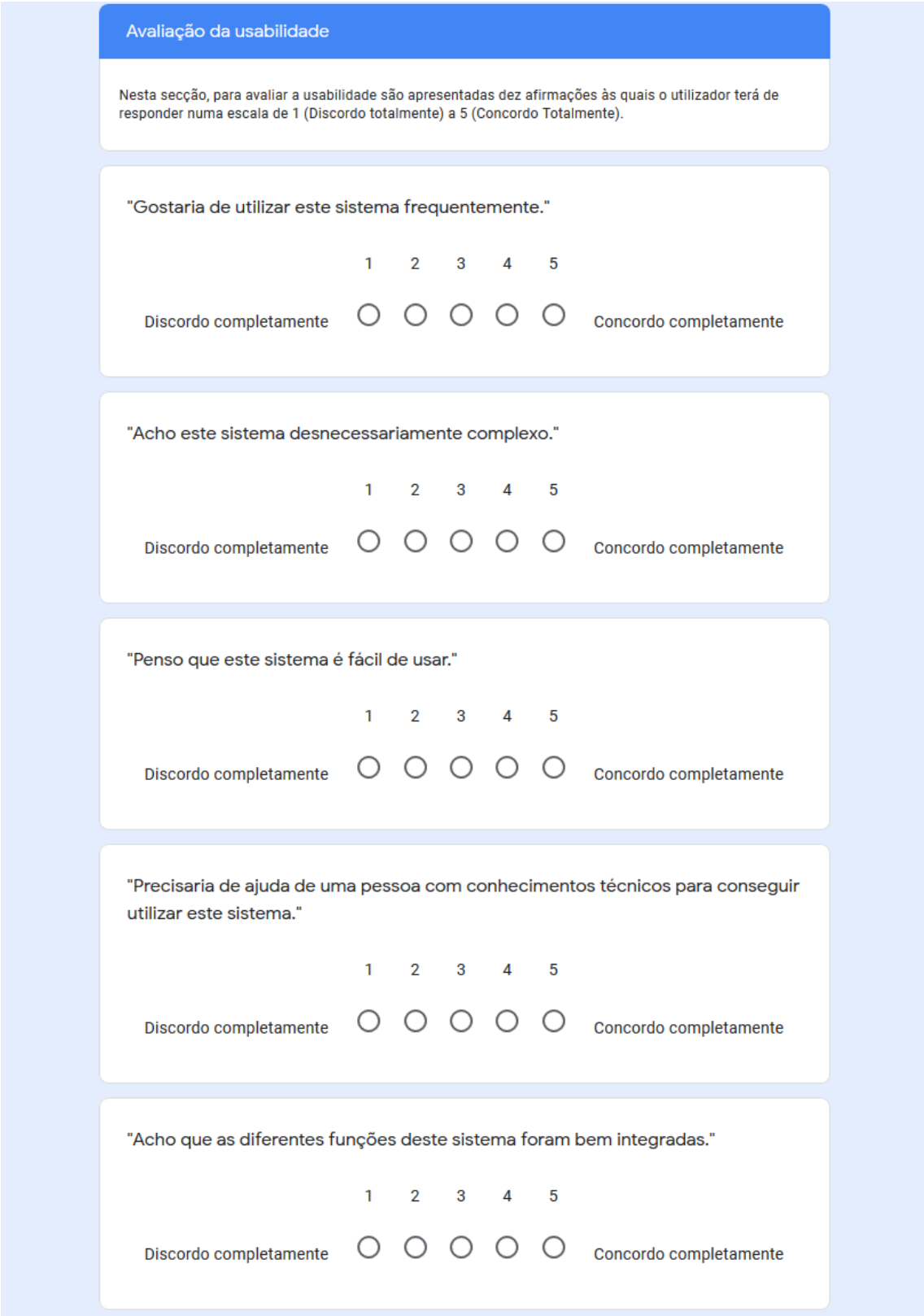
Your answer

Ordene os parâmetros informativos existentes do mais importante para o menos importante. *

| | Últimos resultados | Classificação pré-jogo | Golos | Cartões Vermelhos | Jogadores Chave | Classificação pós-jogo | Encontr da próxim jornad |
|---|---|---|---|---|---|---|---|
| 1ª opção | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 2ª opção | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 3ª opção | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 4ª opção | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 5ª opção | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

Qual a importância de poder ver as métricas sobre o corpo da notícia? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |



Avaliação da usabilidade

Nesta secção, para avaliar a usabilidade são apresentadas dez afirmações às quais o utilizador terá de responder numa escala de 1 (Discordo totalmente) a 5 (Concordo Totalmente).

"Gostaria de utilizar este sistema frequentemente."

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Discordo completamente | ○ | ○ | ○ | ○ | ○ | Concordo completamente |

"Acho este sistema desnecessariamente complexo."

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Discordo completamente | ○ | ○ | ○ | ○ | ○ | Concordo completamente |

"Penso que este sistema é fácil de usar."

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Discordo completamente | ○ | ○ | ○ | ○ | ○ | Concordo completamente |

"Precisaria de ajuda de uma pessoa com conhecimentos técnicos para conseguir utilizar este sistema."

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Discordo completamente | ○ | ○ | ○ | ○ | ○ | Concordo completamente |

"Acho que as diferentes funções deste sistema foram bem integradas."

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Discordo completamente | ○ | ○ | ○ | ○ | ○ | Concordo completamente |


Figure B.1: Writing Tool Assessment Questionnaire

# Appendix C

# Metrics Assessment Questionnaire

This appendix contains the questionnaire, referenced in Subsection 5.5, used to assess the importance of metrics, as textual attributes and readability formulas, for the journalistic work developed at ZOS.

# Avaliação de sistema de métricas textuais para jornalismo desportivo

Sou o Luís Correia, estou a terminar a minha dissertação para o Mestrado Integrado em Engenharia Informática e de Computação em parceria com a ZOS. Durante a tese colaborei no desenvolvimento de um sistema que permite consultar as métricas sobre qualquer artigo ou notícia no site www.zerozero.pt.

Na imagem abaixo está um exemplo da aplicação do sistema de métricas numa notícia. As métricas que podem ser consultadas são:
- número de palavras,
- número de palavras complexas,
- número de letras/caracteres,
- número de frases,
- média de palavras por frase,
- número de palavras da frase mais longa,
- indicadores de níveis de legibilidade.

*Required

Exemplo:

**Características do texto**

Classifique a importância de conseguir consultar as métricas mencionadas para uma notícia. *

| | Muito pouco importante | Pouco importante | Neutro | Importante | Muito importante |
|---|---|---|---|---|---|
| Número de palavras | ○ | ○ | ○ | ○ | ○ |
| Número de frases | ○ | ○ | ○ | ○ | ○ |
| Número de caracteres | ○ | ○ | ○ | ○ | ○ |
| Média de palavras por frase | ○ | ○ | ○ | ○ | ○ |
| Número de palavras complexas | ○ | ○ | ○ | ○ | ○ |

**Indicadores de legibilidade**

Os indicadores utilizados neste sistema são:

- Índice de legibilidade Flesch: avalia de 0 a 100 a facilidade com que lê o texto, correspondendo 0 a menor facilidade legibilidade e 100 a uma maior facilidade de legibilidade.
- Flesch-Kincaid: avalia a legibilidade numa escala de 1 a 14 em forma de anos de escolaridade adequados à leitura, correspondendo 1 a maior facilidade de legibilidade e >14 a menor facilidade de leitura.
- Gunning-Fog: escala igual a Flesch-Kincaid, mas utiliza uma abordagem de avaliação diferente.
- Coleman-Liau: escala igual à anterior e utiliza uma abordagem de avaliação diferente.
- ARI: escala igual às anteriores e utiliza uma abordagem de avaliação diferente.

Classifique a importância de conseguir consultar estes indicadores de legibilidade. *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Pouco importante | ○ | ○ | ○ | ○ | ○ | Muito importante |

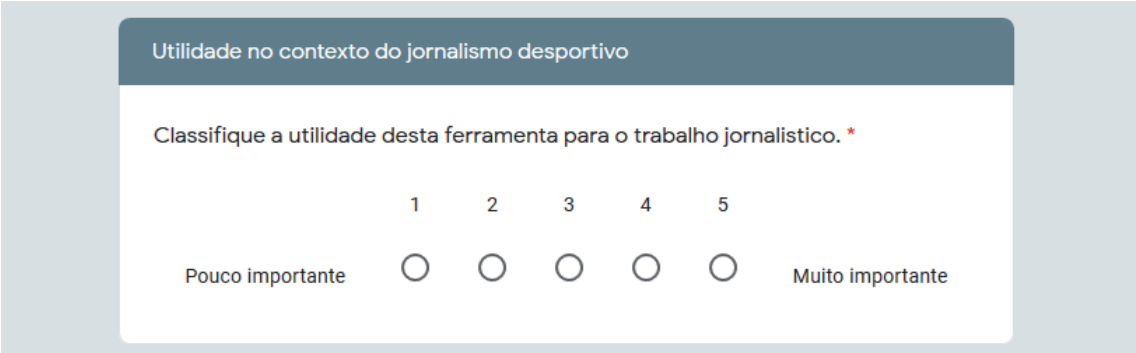



Figure C.1: Metrics Assessment Questionnaire